

# ADEP Data Services Complete Quick Start

## Install and configure JAVA

- 1) <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- 2) Install Java jdk1.6.0\_26 or higher (jdk 1.7 is not supported)
- 3) Select Start > Control Panel > System
- 4) Click the Advanced tab and click Environment Variables
- 5) In the New System Variable box type JAVA\_HOME
- 6) In the value field set the location where you downloaded. For example C:\Program Files\Java\jdk1.6.0\_27
- 7) Click OK.
- 8) In the System Variables area, select Path and click Edit.
- 9) Append %JAVA\_HOME%\bin; at the beginning of the value.
- 10) Click OK.
- 11) Open a Command Prompt
- 12) Verify your java version with: `java -version`

## Install and configure Maven 3

- 1) Download Maven 3 from <http://maven.apache.org/download.html>
- 2) Select Start > Control Panel > System
- 3) Click the Advanced tab and click Environment Variables
- 4) In the New System Variable box type M3\_HOME
- 5) In the value field set the location where you downloaded. For example C:\maven\apache-maven-3.0.3
- 6) Click OK.
- 7) In the System Variables area, select Path and click Edit.
- 8) Append %M3\_HOME%\bin; at the beginning of the value.
- 9) Click OK.
- 10) Open a command prompt
- 11) Test the Maven install with **`mvn -version`**
- 12) This should return

```
Apache Maven 3.0.3 (r1075438; 2011-02-28 12:31:09-0500)
Maven home: C:\maven\apache-maven-3.0.3
Java version: 1.6.0_29-ea, vendor: Sun Microsystems Inc.
Java home: C:\Program Files\Java\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
```

## Install ADEP

- 1) Download ADEP <http://www.adobe.com/devnet/enterprise-platform/trial-downloads.html>
- 2) Select the Digital Enterprise Platform Experience Services
- 3) Rename the quickstart adept-ria-quickstart-10-0-all-all.jar adding adept-ria-quickstart-10-0-all-all-**4502**.jar so you know what port this particular jar is running and configured for.
- 4) Place this jar file in a folder structure named **c:\demo\ADEP**
- 5) Double Click the jar file to install it. This will extract the contents into that folder on your system. You can also run the command below to launch the application.
  - a. Java -jar adept-ria-quickstart-10-0-all-all-4502.jar
- 6) When ADEP starts it will extract the contents into crx-quickstart. You can double click on the jar anytime you want to start the ADEP and it will read the files from the crx-quickstart folder.
- 7) Once started login with username: admin password: admin
- 8) The configuration of ADEP is complete next you'll need to configure Flash Builder with ADEP.

## Install Flash Builder 4.5.0 and Plugins

- 1) Install Flash Builder 4.5.0 [http://www.adobe.com/go/adept\\_qstooling\\_fb450](http://www.adobe.com/go/adept_qstooling_fb450)
- 2) Follow steps to install Flash Builder 4.5.0
- 3) Start Flash Builder ( Right click and "Run as administrator" )
- 4) To install the ADEP plugins Click Help > Install New Software.
  - a. From within **Work with** drop-down, select "Adobe Enterprise Suite extensions for Flash Builder" - <http://aesplugins.adobe.com/aes/tooling/plugin/10.0>.
  - b. Click Select All and select all available plugins, click next.
  - c. Continue with prompts to finish install.

## Enable RDS within ADEP for Flash Builder

- 1) Go to <http://localhost:4502/system/console> and login with admin/admin
- 2) click Configuration
- 3) Find "Adobe Data Services" and click to open it.
- 4) Select "Enable RDS"
- 5) Click Save.

## Download & Install Experience Services SDK and samples into ADEP

- 1) Go to <http://localhost:4502/> and login with admin/admin
- 2) Click **Package Share** and login using your Adobe ID username and password.
- 3) Select **Packages > Public > Adobe > Digital Enterprise Platform > 10.0 > product**.
- 4) Click **Download** next to the es-sdk-pkg-1.0.186.zip package.
- 5) Once download you need to install the packages into ADEP.
- 6) Go to <http://localhost:4502/> and click the Package icon. The **Package Manager** screen opens.
- 7) Click **Install** next to the downloaded ex-sdk-pkg-1.0.186.zip package.

- 8) Click **Install** on the confirmation screen.
- 9) NOTE: The icon next to the package will now say “Share”.

### Retrieve the Experience SDK within Flash Builder to your development system

- 1) Start Flash Builder
- 2) Select **Windows > Preferences**
- 3) In the left panel, select **Adobe > RDS Configuration**
- 4) Select **Experience Server (localhost)**
- 5) Click **Test Connection** to ensure that RDS is configured.
- 6) Select OK.
- 7) In the left panel, select **Adobe > Experience Services**
- 8) In the SDK Root Directory field, specify the location where you want to place the SDK.
  - a. This directory will be used by Flash Builder for the SDKs.
  - b. Create a folder named **ExperienceServicesSDK** under **C:\demo**
  - c. I placed mine in **C:\demo\ExperienceServicesSDK**
- 9) Click **Retrieve SDK** and Click **Ok**.

### Retrieve the Data Services JAR file.

- 1) Unzip the template that we will use to build an OSGI package.
  - a. This file is located in  
**C:\demo\ExperienceServicesSDK\riaservices\dataservices\4.5.0\dataservices-example-app-template.zip**
- 2) Unzip the dataservices example app template to a folder. I will extract mine to the following folder **c:\demo\dataservices\_template**

**NOTE:** Open the file **C:\demo\dataservices\_template\sourceCode\pom.xml** and look at the build resources **/src/main/java** below. We will place our source within the **C:\demo\dataservices\_template\sourceCode\src\main\java** folder and it will be compiled by Maven for us.

```
<resources>
  <resource>
    <directory>${basedir}/src/main/java</directory>
    <includes>
      <include>**/*.xml</include>
      <include>**/*.properties</include>
    </includes>
  </resource>
</resources>
```

- 3) The Data Services JAR file is needed when your project uses the Data Services APIs such as `flex.data.assemblers.AbstractAssembler`. In the next step we will copy the `dataservices-bundle.jar` and place this within the `lib` directory for the package.

- 4) Copy the **C:\demo\ExperienceServicesSDK\riaservices\dataservices\4.5.0\java\dataservices-bundle.jar** to the **C:\demo\dataservices\_template\lib** directory

### Create Java EchoService

- 1) Create a file named "EchoService.java" within the following folder:  
**C:\demo\dataservices\_template\sourceCode\src\main\java\com\adobe\support\example\
  - a. You will need to create the **com/adobe/support/example** folders.**

```
package com.adobe.support.example;

public class EchoService {
    public String sayHello(String val){
        return "Adobe Digital Enterprise Platform says: " +val ;
    }
}
```

- 2) Modify the **C:\demo\dataservices\_template\sourceCode\pom.xml** file with the following Export-Package like below adding **com.adobe.support.example**,

```
<instructions>
    ...
    <Export-Package>
        com.sample,
        com.adobe.support.example,
    </Export-Package>
</instructions>
```

### Build Maven Package to deploy into ADEP

- 1) Start the Command Prompt
- 2) Change your directory to **C:\demo\dataservices\_template**
- 3) Enter the following command: **mvn clean install**
- 4) This should return a result similar to the following:

```
[INFO] Reactor Summary:
[INFO]
[INFO] Data Services Example - Parent ..... SUCCESS [0.625s]
[INFO] Data Services Example - OSGi Bundle ..... SUCCESS [4.343s]
[INFO] Data Services Example - Package ..... SUCCESS [3.302s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.900s
[INFO] Finished at: Wed Oct 12 09:46:55 EDT 2011
[INFO] Final Memory: 13M/239M
```

### Deploy the Maven created package using the Package Manager to the Experience Server

- 1) To deploy this go to <http://localhost:4502/>
- 2) Enter the username and password **admin/admin**

- 3) Click Packages.
- 4) From the menu bar, click Upload Package.
- 5) Select the zip file located here **C:\demo\dataservices\_template\package\target\dataservices-example-app-package-1.0-SNAPSHOT-mypackage.zip**
- 6) Click OK.
- 7) From the new package entry menu, click Install.
- 8) In the Install Package dialog, click Install.

### Defining the server destination for the echo service within CRDDE Lite.

- 1) Go to <http://localhost:4502>
- 2) Login with username/password admin/admin
- 3) Click CRXDE Lite.
- 4) From the left-side folder structure view, open **etc->aep->config->dataservices->destinations->remoting**.
- 5) Right click the remoting folder and select Create, Create File.
- 6) Specify **com.adobe.support.example.echoservice.xml** in the Name field.
- 7) Click Ok.
- 8) This should open the file for edit if not you can ( Double click the com.adobe.support.example.xml file to edit )
- 9) Enter the XML shown replacing "null".

```
<?xml version="1.0" encoding="UTF-8"?>
<destinations>
<destination channels="my-amf" id="echoservice">
  <properties>
    <source>com.adobe.support.example.EchoService</source>
  </properties>
</destination>
</destinations>
```

- 10) From the top menu, click the **Save All** button.

### Create a Flash Builder client Project for the echo service

- 1) Start Flash Builder
- 2) Click File, New, Select **Flex Project for ADEP Experience Services**
- 3) In the Project Name box, specify a name for your project "ADEP Demo"
- 4) Under the Application Type, select **Web (runs in Adobe Flash Player)**.
- 5) Keep the **Flex SDK version 4.5** and click **Next**.
- 6) From the RDS server, select Experience Server (localhost).
- 7) Click the Configure button.
- 8) Select Experience Server.
- 9) In the Host Name box, specify the hostname "localhost" or "127.0.0.1".
- 10) In the Port Number box, specify the Port Number 4502

- 11) In the User Name, specify "admin" and password "admin".
- 12) Test to make sure the project can communicate with the server by clicking the "Test Connection" button. If the communication is successful, then a message that states that the test connection is successful is displayed.
- 13) Click OK.
- 14) Click Finish.
- 15) Within the **ADEP Demo** project edit **Main.mxml** the following MXML example into your new project.

```
<?xml version="1.0" encoding="utf-8"?>
  <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" minWidth="955"
    minHeight="600">
    <fx:Declarations>
      <mx:RemoteObject id="echoOb"
        destination="echoservice"
        result="handleResult(event);"
        fault="handleFault(event);"/>
    </fx:Declarations>

    <fx:Script>
      <![CDATA[
        import mx.controls.Alert;
        import mx.rpc.events.FaultEvent;
        import mx.rpc.events.ResultEvent;

        [Bindable]
        public var echoResultString: String = "";

        public function handleResult(event:ResultEvent):void {
          // Handle result by populating the TextArea control.
          echoResultString = event.result as String
          outputResult.text=echoResultString;
        }

        public function handleFault(event:FaultEvent):void {
          // Handle fault.
          Alert.show(event.fault.faultString, "Fault");
        }
      ]]>
    </fx:Script>

    <s:Button id="btnHello" x="296" y="28" label="Button"
      click="echoOb.sayHello(inputText.text)"/>
    <s:TextInput id="inputText" x="108" y="28"/>
    <s:Label id="outputResult" x="109" y="78" text="Label"/>

  </s:Application>
```

- 16) Save the edit.
- 17) Select Project from the menu and if "Build Automatically" is not checked, click Build All.
- 18) Select Run from the menu and click debug. This will launch the application in your browser.
- 19) Enter a name within the TextInput box and click "Button".