

Setup Guide for ARIS Server

This guide will explain in detail how to setup the server and game editor for an ARIS server. More information about ARIS can be found at <http://arisgames.org>. This server serves two functions: to allow for the creation and editing of ARIS games and to serve as a gateway for the client applications (mobile devices) playing an ARIS game. This guide is broken up into two parts: the setup for the server, and for the editor.

The server:

The main components of the server:

- Apache webserver: this utilizes PHP extensively for all server roles.
- AMFPHP: this allows for communication between the flash application and the server's php files.
- MySQL: the database where game, editor and player data is stored.

Implementation for the server:

- CentOS 5.9
- MySQL 5.0.95
- Apache 2
- PHP 5.1.6

Server setup notes:

- All server functions used by ARIS can be tested in the service browser. Any data changed through this browser will be changed in the database as well, so use at your own risk!
- We had serious issues getting the server to function properly using virtual hosting in Apache, and consequently would not recommend it for testing purposes.

Server setup guide:

1. Install the required packages:

```
[root@aris]# yum install mysql mysql-server httpd php subversion
```

2. Create the MySQL database for ARIS:

```
[root@aris]# service mysqld start
[root@aris]# mysql -u root
mysql>create database aris;
mysql>use mysql;
mysql>INSERT INTO user (Host,User,Password) VALUES('%','aris','yourpassword');
mysql>grant usage on *.* to 'aris'@'localhost' identified by 'yourpassword';
mysql>flush privileges;
```

3. Create the directories used by the webserver:

```
[root@aris]# mkdir /var/www/html/server
[root@aris]# cd /var/www/html/server
```

4. Use subversion to obtain the current server package from ARIS' repository:

```
[root@aris]# svn checkout https://arisgames.googlecode.com/svn/trunk/server .
```

5. Copy ~server/config.class.php.template to ~server/config.class.php and edit it as follows:
(mysqlBinPath variable may differ depending on your system)

```
...

const dbUser = 'aris';

const dbPass = 'yourpassword';

const dbSchema = 'aris';

const dbHost = 'localhost';

const adminEmail = 'youremail@yourdomain.com; //This email address, if
specified, will be notified of server errors

const WWWPath          = 'http://yourdomain.com;

const serverWWWPath     = 'http:// yourdomain.com /';

const gamedataWWWPath   = 'http:// yourdomain.com /server/gamedata';

const playerdataWWWPath = 'http:// yourdomain.com /server/playerdata';

const gamedataFSPath    = '/var/www/html/server/gamedata'; //Make Sure this
directory is writable

const playerdataFSPath  = '/var/www/html/server/playerdata'; //Make Sure this
directory is writable

const serverErrorLog    = '/var/log/httpd/aris_error_log'; //Make sure this
file is writable

const gameMediaSubdir   = ''; //if this is used, you must include the trailing
slash

const mysqlBinPath      = '/usr/bin';

...
```

6. Import aris_db_install.sql into the database (do this while in the ~server/ directory):

```
[root@aris]# mysql -u aris -p aris < aris_db_install.sql
```

7. Set file permissions for the server directories:

```
[root@aris]# cd /var/www/html/
[root@aris]# chown -R apache.apache server
[root@aris]# chmod -R 775 server
[root@aris]# chmod -R u+s server/gamedata
```

8. Ensure both httpd and mysqld are started:

```
[root@aris]# service httpd restart
[root@aris]# chkconfig httpd on
[root@aris]# service mysqld restart
[root@aris]# chkconfig mysqld on
```

9. The service browser can be used to test the server, and is accessible at <http://yourdomain.com/server/browser/>

The editor:

The main components of the editor:

- main.html: this page executes the flex-based editor in a web browser.
- MySQL: the database where game, editor and player data is stored.

Implementation for the editor:

- Flash-enabled web browser
- Flashbuilder 4.7

Editor setup notes:

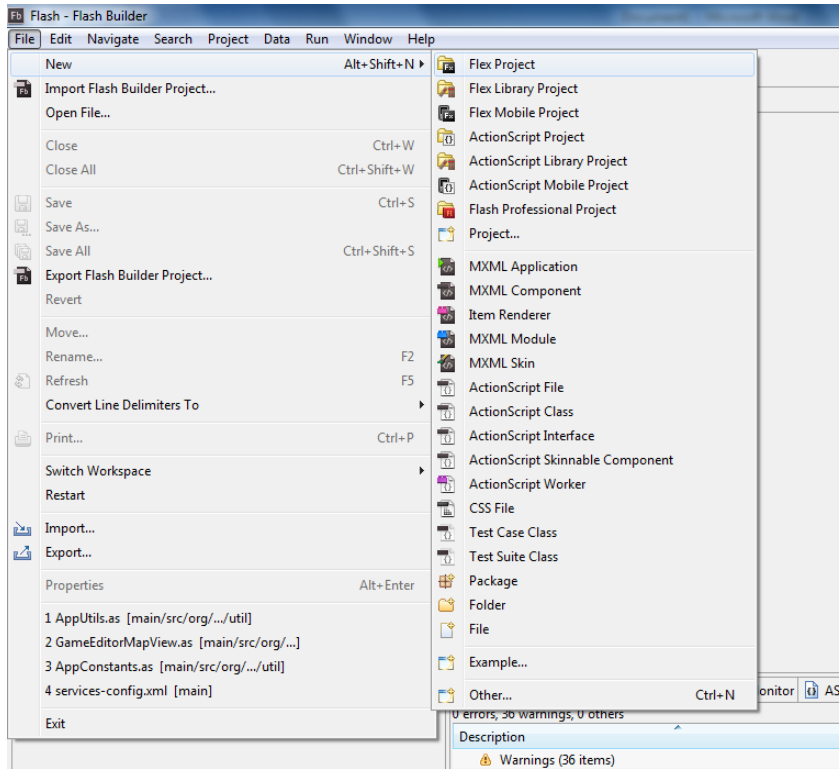
- Update(4/30/2013) The editor successfully compiles and runs locally but cannot be run from a webserver due to the Google Map Flash API key. Unless you already have a Flash API key, you will be unable to get the editor to function except when compiled and run on a local machine. Google has deprecated the Flash API key, thus no new keys are being issued and old Flash API keys will expire sometime in 2014.
- We used Flashbuilder 4.7 to compile the Flex project, however one could use the Flex SDK and compile the project through the command line.

Editor setup guide:

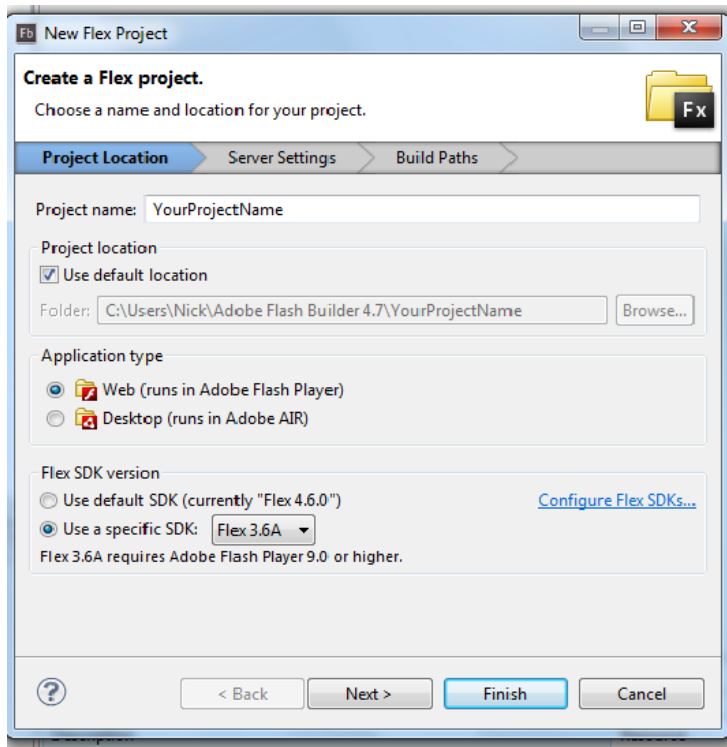
1. Use subversion to obtain the current server package from ARIS' repository:

```
[root@aris]# svn co https://arisgames.googlecode.com/svn/trunk/flexEditor
```

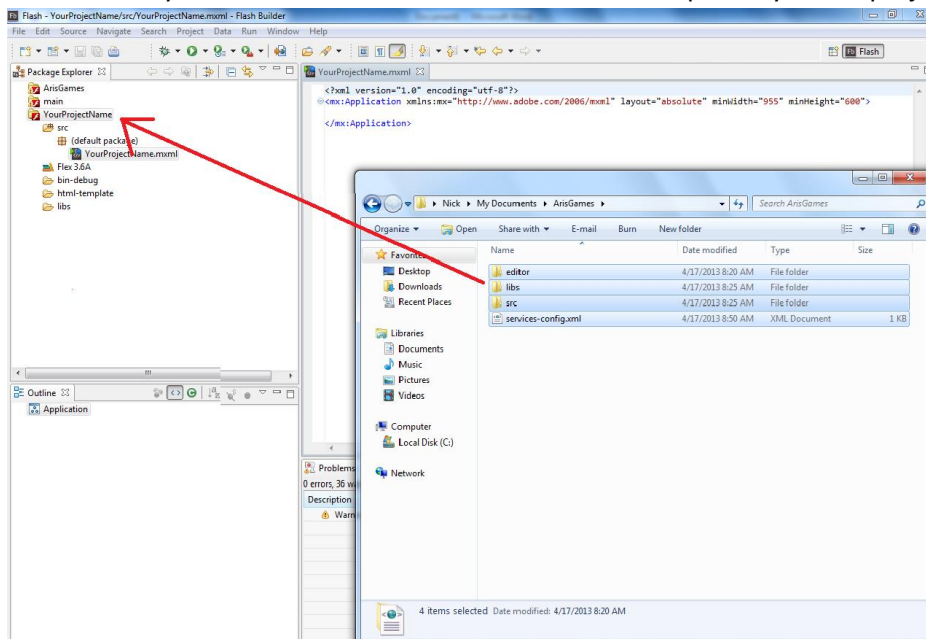
2. In Flashbuilder, create a new Flex project:



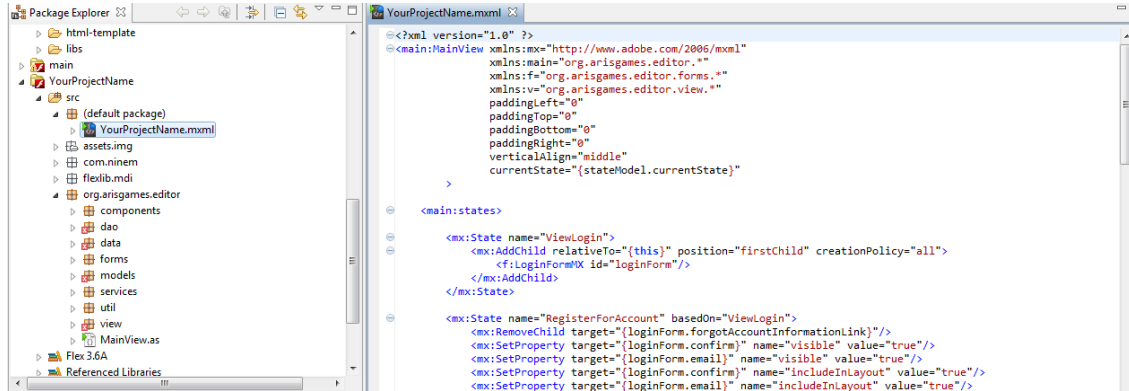
3. Choose a name for your project and under “Flex SDK version” select Flex 3.6A (although any 3.x version should work)



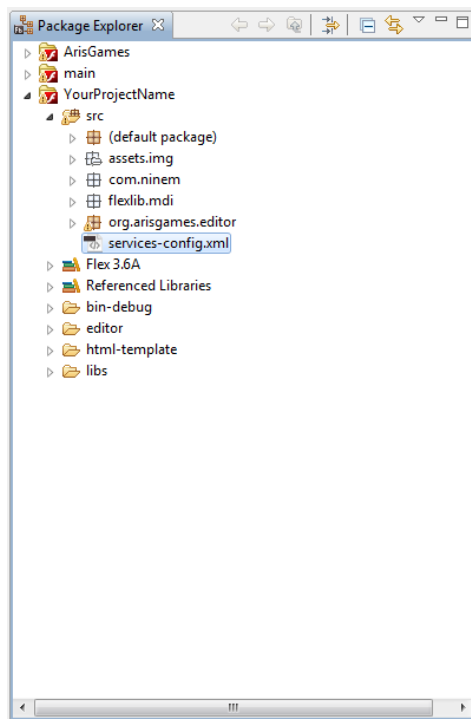
4. Add the files you downloaded from the ARIS flexEditor repository to the project:



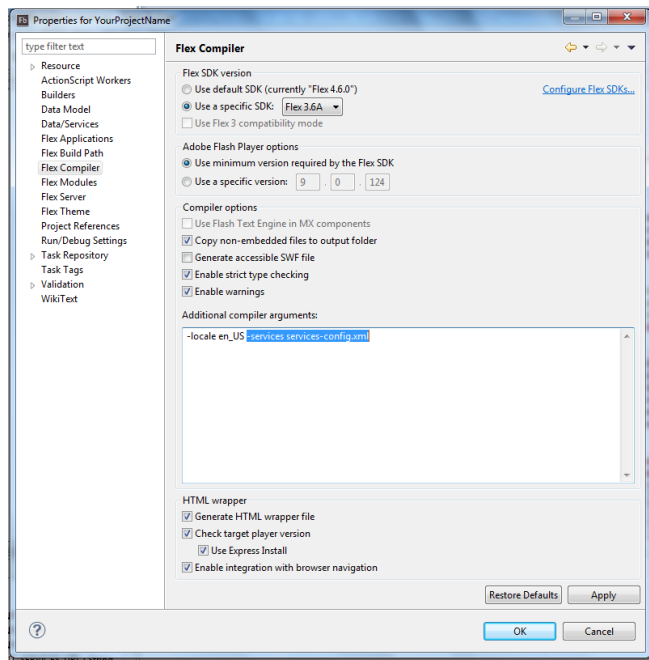
5. The main mxml file from the ARIS repository is named “main.mxml”. In the src directory of the flex project, remove the “YourProjectName.mxml” file and rename “main.mxml” to “YourProjectName.mxml”:



6. Verify that “services-config.xml” is inside the src/ directory of the project:



7. In the same window, right-click “Flex 3.6A” and select “Properties”. Then select “Flex Compiler” from the list on the left panel, and add “-services services-config.xml” to the “Additional compiler arguments” textbox:



8. Now open `org.arisgames.editor/util/AppConstants.as`, and alter the following constant:
APPLICATION_ENVIRONMENT_ROOT_URL and
The ROOT_URL should be the web address for the server, aka “`http://yourdomain.com/server`”.

9. Now open `src/services-config.xml` and change line 22 so it points to your server:

```
...  
<endpoint uri="http://yourdomain.com/server/gateway.php" ...  
...
```

10. All the necessary changes have now been made, and you need only change the output directory in the project settings to where you want it compiled. Then simply run or compile the project