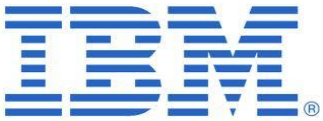


IBM WebSphere Application Server (z/OS)

Implementing Enhanced Form Based Authentication with Servlet Filters in Java EE Applications

Overview

An application protected with standard form based authentication has limited options in handling scenarios that involve more than just simple userid/password authentication.

For instance, consider the following conditions:

- expired password
- invalid password
- invalid userid
- revoked userid
- userid not defined to OMVS

With standard form based authentication, all of the conditions listed will produce the same result: being passed to the error page. Unfortunately, there is no indication of the exact error condition, leaving the user in the dark as to why they were sent to the error page.

This document describes an enhancement to standard form based authentication, complete with the ability to change a user's password from an HTML form and provide error messages based on specific conditions, including the ones listed above.

This implementation of enhanced form based authentication makes use of servlet filters to perform preprocessing of the userid and password. It should work both in a base application server and in a network deployment configuration.

Note: This document assumes prior knowledge of all the requirements needed for configuring standard form based authentication, both in the application and on the server. Before proceeding, ensure that an application server is available to which applications using standard form based authentication can be deployed. This includes having global security functioning properly.

Getting Started

Before using the enhanced form based authentication application, the application server must be configured for standard form based authentication. For testing purposes, a sample application using standard form based authentication has been included (StandardFBASample.ear) with this document.

This application uses a security role called “AuthorizedUser.”

Use this application to test standard form based authentication before proceeding to enhanced form based authentication. Using any type of authentication mechanism in WebSphere Application Server requires turning on global security. This document assumes the user will be using IBM® Rational® Application Developer for WebSphere® Software (hereafter referred to as RAD) to build their application.

Using the Form Based Authentication Sample Without SSL

Testing standard form based authentication without SSL can be beneficial in diagnosing any problems that may arise due to SSL. However, running an application that uses form based authentication without SSL in a production environment is not recommended.

The sample application included with this document (StandardFBASample.ear) uses a security role called "AuthorizedUser." There must be a profile called AuthorizedUser (case sensitive) in the EJBROLE class in RACF. For the purposes of this document, we assume you have given universal access of READ to AuthorizedUser. In a production environment, it is recommended that you consult with your local security administrator before doing this.

After deploying StandardFBASample.ear, point your web browser to the following URL.

http://example.com/StandardFBA/protected/index.jsp

The browser should display an HTML page asking for a userid and password. Give any valid userid and password defined to RACF and the browser should be redirected to the JSP.

Using the Form Based Authentication Sample With SSL

Using SSL with the sample application is an easy task after getting it to work without SSL. Simply point a web browser to the SSL port. For example, use the following URL as a template:

https://example.com/StandardFBA/protected/index.jsp

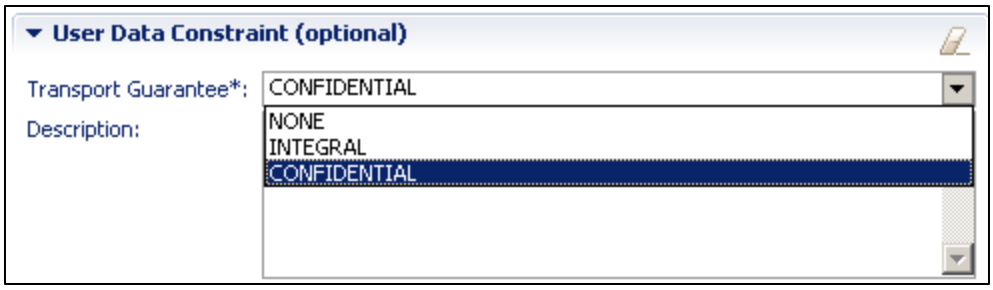
Forcing the Form Based Authentication Sample to Use SSL

WebSphere Application Server includes the ability to redirect a request from the HTTP port to the HTTPS port.

To make the sample require SSL, add or modify the following lines to the <security-constraint> tag in the deployment descriptor (web.xml).

```
<user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

In RAD, this can be done in the Web Application Deployment Descriptor Editor.



Alternatively, there is an ear file (StandardFBASample-secure.ear) supplied with this document that is already set to do this. Simply deploy this application and point your web browser to the following URL:

http://example.com/StandardFBASecure/protected/index.jsp

The application server should redirect the request to the secure port.

Infrastructure Configuration for Enhanced Form Based Authentication

At this point, the standard form based authentication application should be working with SSL. Using the enhanced form based authentication functionality will require modifications to be made to the application server to which it is deployed.

The following files in the installation HFS must be marked as program controlled. You can use the extattr command in UNIX® System Services to do so. Keep in mind that your SMPE HFS must be mounted read/write in order change the extended attributes.

Future WebSphere maintenance may require marking other modules as program controlled. The list below is current at the time of publication.

```
$SMPE_HOME/lib/modules/bbgsr
$SMPE_HOME/lib/modules/bbgutil
$SMPE_HOME/lib/modules/bbgcorba
$SMPE_HOME/lib/modules/bbgras
$SMPE_HOME/lib/modules/bbgmwenv
$SMPE_HOME/lib/modules/bbglrt
$SMPE_HOME/lib/modules/bbg3mvs
$SMPE_HOME/lib/modules/bbgoxfcn
$SMPE_HOME/lib/modules/bbgoslip
$SMPE_HOME/lib/modules/bbgrjras
$SMPE_HOME/lib/modules/bbgboa
$SMPE_HOME/lib/modules/bbgorb
$SMPE_HOME/lib/modules/bbgcf
$SMPE_HOME/lib/modules/bbgoxmem
$SMPE_HOME/lib/modules/bbgcomm
$SMPE_HOME/lib/modules/bbgsec
$SMPE_HOME/lib/modules/bbgtots
$SMPE_HOME/lib/modules/bbgglog
$SMPE_HOME/lib/modules/bbgdasr
$SMPE_HOME/lib/modules/bbgujuu
```

```
$SMPE_HOME/lib/modules/bbgosmf
$SMPE_HOME/lib/bbgosib.so
$SMPE_HOME/lib/bboosrmr
$SMPE_HOME/lib/modules/bbgcdbbm
$SMPE_HOME/lib/modules/bbgclvti
$SMPE_HOME/lib/modules/bbgenus
$SMPE_HOME/lib/modules/bbgjutil
$SMPE_HOME/lib/modules/bbglpmis
$SMPE_HOME/lib/modules/bbgubinf
$SMPE_HOME/lib/modules/bbguph
$SMPE_HOME/lib/modules/bborsmct
$SMPE_HOME/lib/modules/bbo3sbcp
```

For WebSphere XD Compute Grid v6.1.1, the following modules will also need to be marked program controlled.

```
$WCG_SMPE_HOME/lib/libProcessCPU64.so
$WCG_SMPE_HOME/lib/libNodeDetect64.so
$WCG_SMPE_HOME/lib/libJobSync64.so
```

For WebSphere XD Compute Grid v8, the following modules will also need to be marked program controlled.

```
$WCG_SMPE_HOME/lib/libWCGProcessCPU64.so
$WCG_SMPE_HOME/lib/libWCGNodeDetect64.so
$WCG_SMPE_HOME/lib/libJobSync64.so
```

In TSO, you will also need to issue the following commands:

```
RALT PROGRAM * ADDMEM('SYS1.CPP.SCLBDLL2'//NOPADCHK)
SETROPTS WHEN(PROGRAM) REFRESH
```

Failure to do so may result in error messages similar to the following:

```
BPXP015I HFS PROGRAM /wasv7config/h2cell/h2nodeb/AppServer/lib/bboosrmr IS NOT MARKED
PROGRAM CONTROLLED.
```

Application Modification

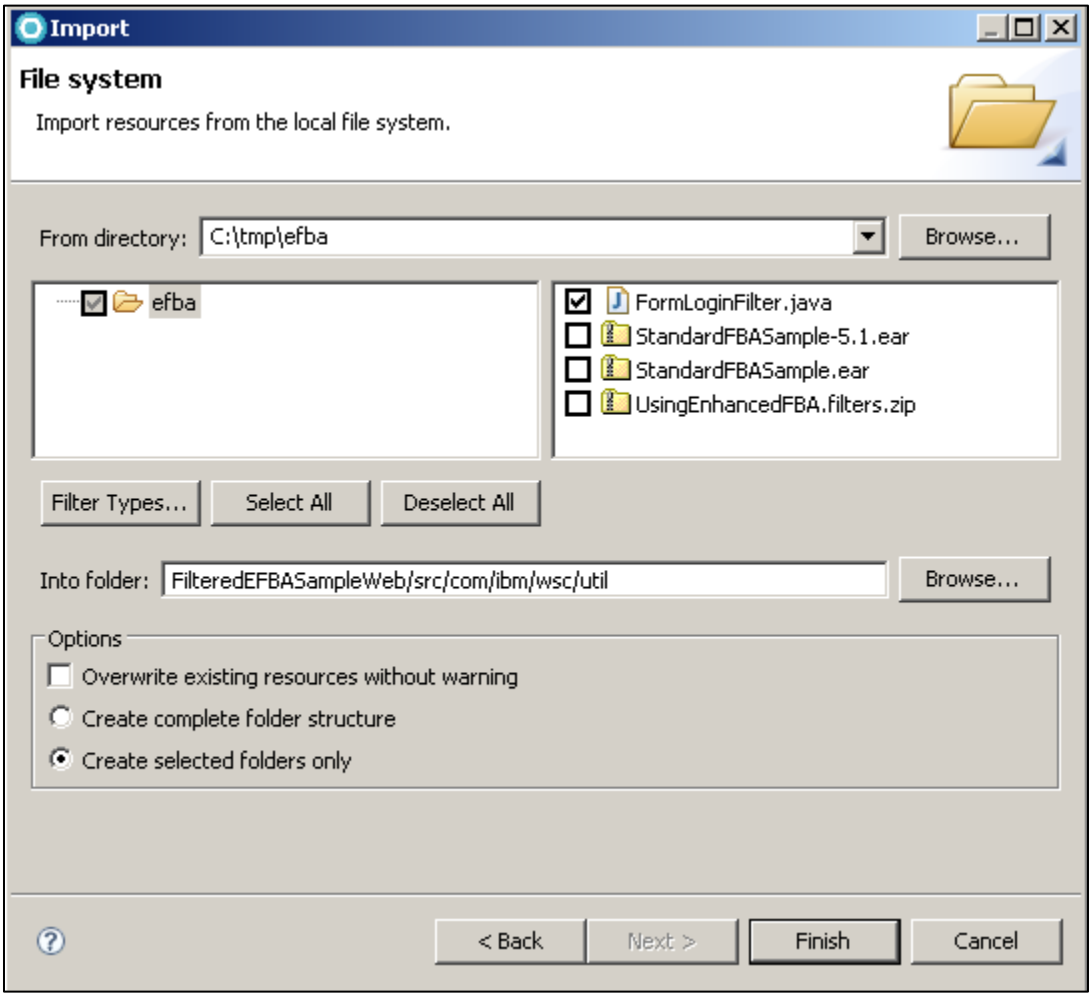
An application to be protected with enhanced form based authentication needs several modifications.

- Filter must be inserted in the application
- Error pages must be inserted in the application
- Login page may be modified
- Deployment descriptor must be modified

For the remainder of this document, it is assumed that the developer will be using RAD.

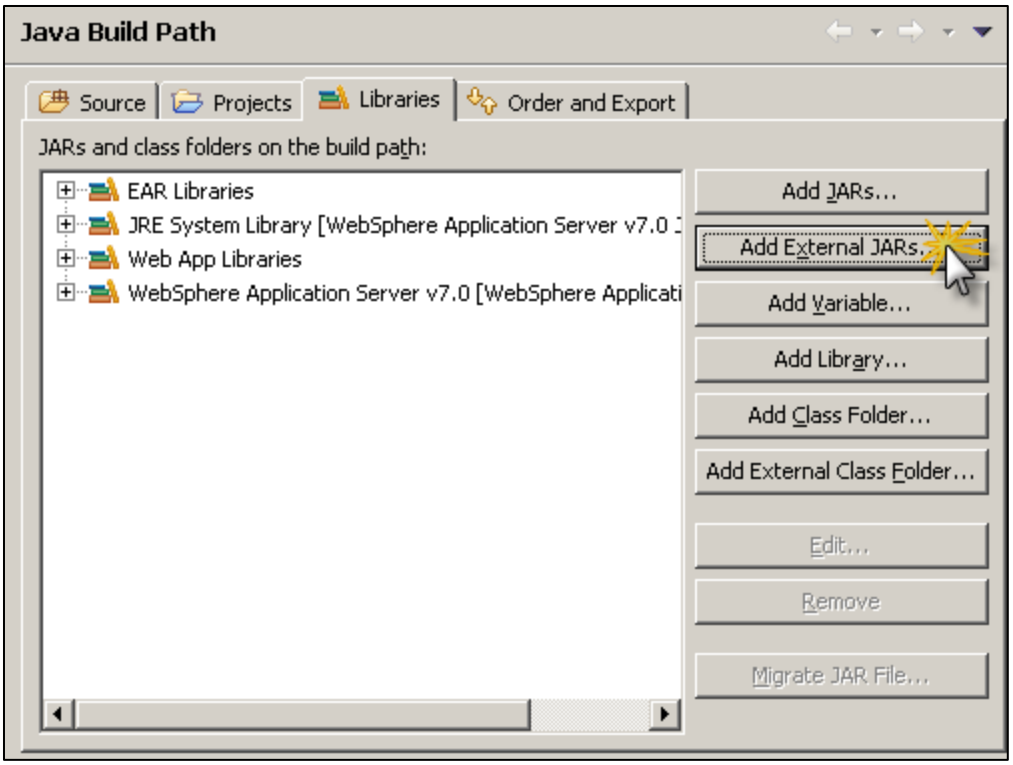
Import Filter

Import filter into the application. The filter is in package com.ibm.wsc.util. This can be changed if necessary.



RAD will report errors because some required classes are not available to the application. To resolve this, you will need to FTP **RACF.jar** to your local workstation. This jar file is typically found in **$JAVA_HOME/lib**.

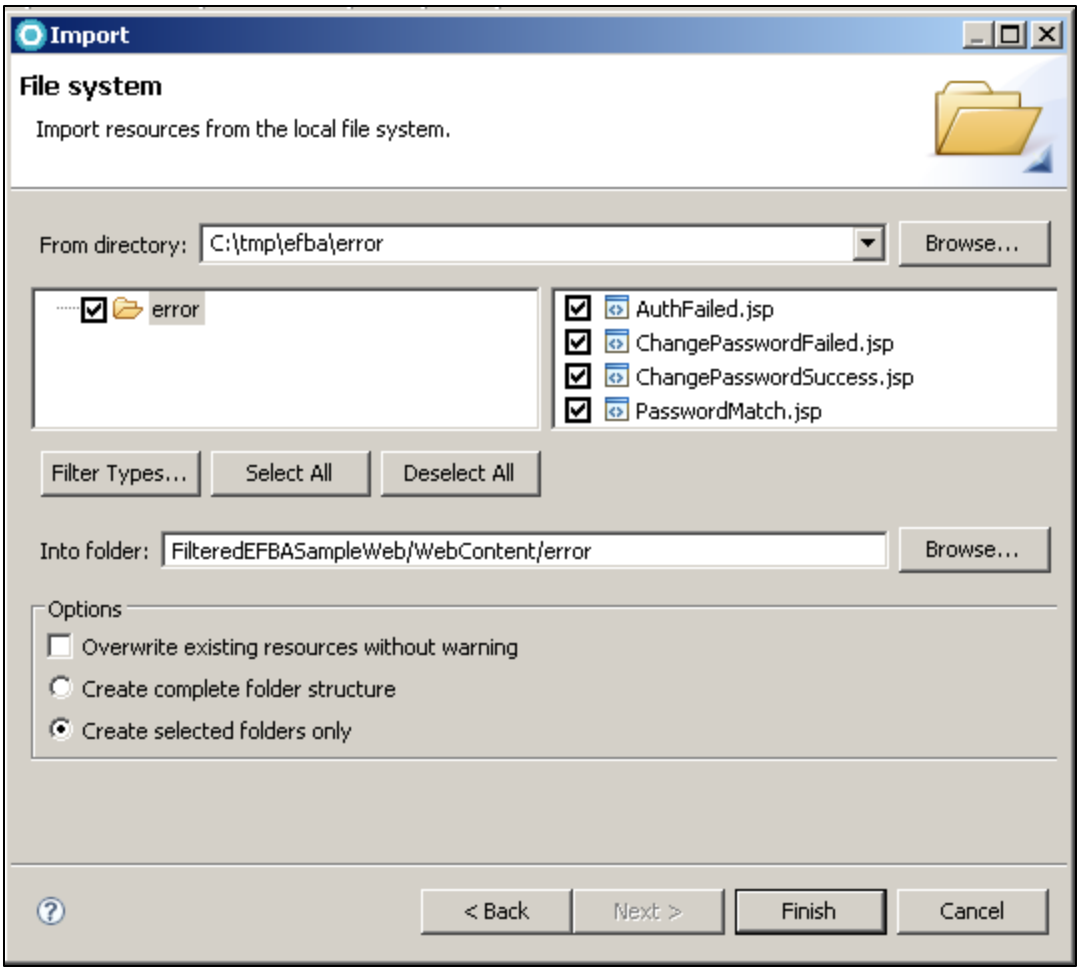
Once you have the jar file on your local workstation, you need to add it to the build path. You can do this by right-clicking on your web module and selecting Build Path→Configure Build Path… Then click on the Libraries tab and click Add External Jars…



Navigate to RACF.jar and click OK. This should resolve any errors associated with the filter.

Import Error Pages

Import the error pages into the "Web Content" folder. The four error pages should stay under the "error" subfolder.



Modify Login Page

If the application needs the ability to let the user change their password, then the login page must be modified. Otherwise, it will work as is.

The following is a vanilla HTML form used in standard form based authentication.

```
<form method="post" action="j_security_check">
	Username <input type="text" name="j_username" maxlength="8" size="8">
	<br/>
	Password <input type="password" name="j_password" maxlength="8" size="8">
	<br/><br/>
	<input type="submit" value="login">
	<br/>
</form>
```

Add the following lines to give users the ability to change their password.

```
<form method="post" action="j_security_check">
	Username <input type="text" name="j_username" maxlength="8" size="8">
	<br/>
```

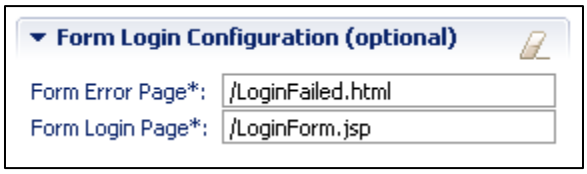
```
        Password <input type="password" name="j_password" maxlength="8" size="8">
        <br/>
        New Password <input type="password" name="newPassword" maxlength="8" size="8">
        <br/>
        Confirm Password <input type="password" name="confirmPassword" maxlength="8" size="8">
        <br/><br/>
        <input type="submit" value="login">
        <br/>
</form>
```

Alternatively, this document comes with a login page and error page to replace the existing ones. To do this, you will need to import LoginPage.jsp and LoginFailed.jsp into the WebContent folder of your web application.

Then in the deployment descriptor, you will change the value of the <form-login-page> tag.

```
        <form-login-config>
                <form-login-page>/LoginForm.jsp</form-login-page>
                <form-error-page>/LoginFailed.html</form-error-page>
        </form-login-config>
```

This can also be done in RAD in the Web Application Deployment Descriptor Editor, under Login Configuration.



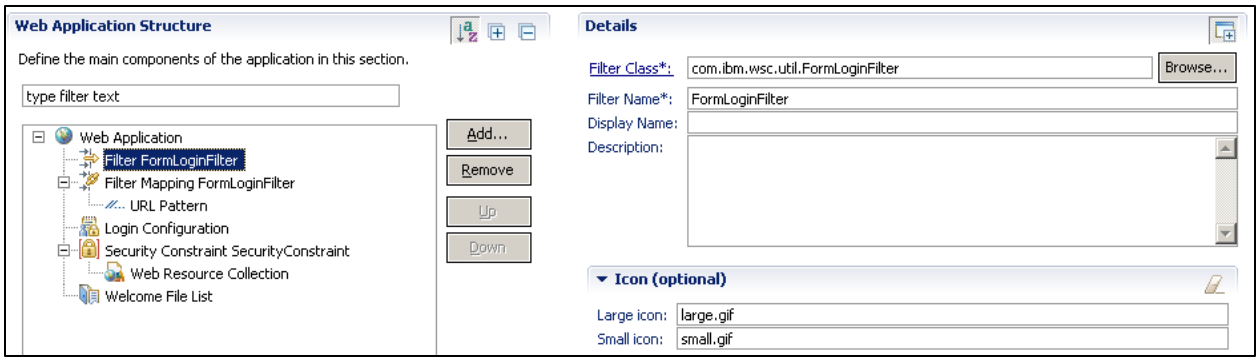
Adding Filter to Deployment Descriptor

The deployment descriptor must be modified to tell the application server when to use the filter.

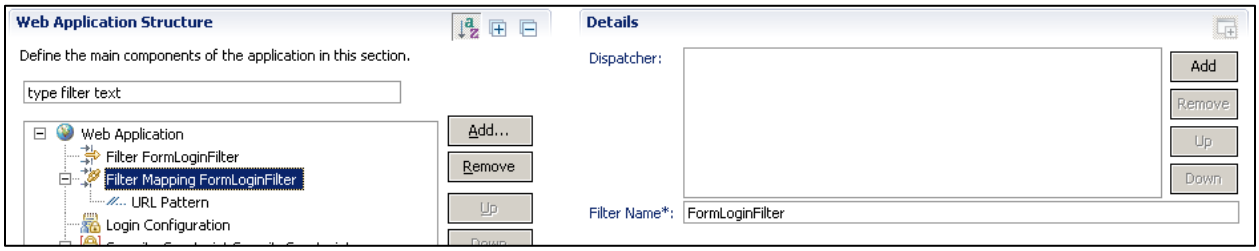
Add the following lines to the deployment descriptor. This can be anywhere inside the <web-app> tag. This will tell the application server to run the filter before j_security_check.

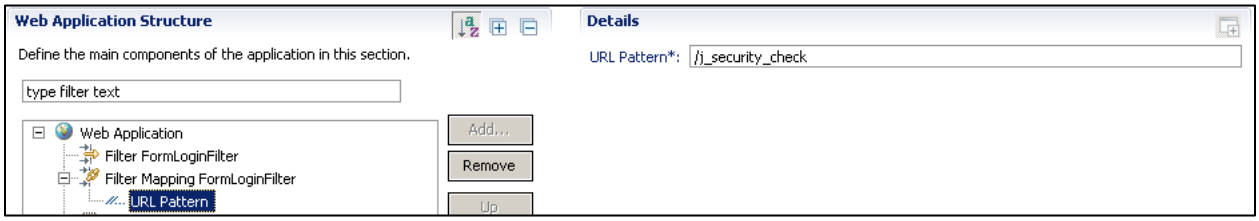
```
        <filter>
                <filter-name>FormLoginFilter</filter-name>
                <display-name>FormLoginFilter</display-name>
                <filter-class>com.ibm.wsc.util.FormLoginFilter</filter-class>
        </filter>
        <filter-mapping>
                <filter-name>FormLoginFilter</filter-name>
                <url-pattern>/j_security_check</url-pattern>
        </filter-mapping>
```

This can also be done in RAD in the Web Application Deployment Descriptor Editor. First, add a filter. The sample filter associated with this document is com.ibm.wsc.util.FormLoginFilter.



Then add a filter mapping and a URL pattern. Make sure the URL pattern is set to /j_security_check. This will ensure that the filter is run prior to authentication.





Sample Enhanced Form Based Authentication with Servlet Filters

A sample application (FilteredEFBASample.ear) using enhanced form based authentication with servlet filters is provided with this document. This application also requires an EJBROLE called AuthorizedUser. The user data constraint is confidential so SSL is required.

If desired, deploy the sample application and point a web browser to the following URL:

http://<server>[:port]/FilteredEFBA/protected/index.jsp

Test various scenarios such as invalid userid and incorrect password to ensure that the application is functioning.

Using the Servlet Filter in a Shared Library

It is possible to have the servlet filter separate from the application. The servlet filter can be placed in a shared library. Thus, any application using that shared library will also have access to the servlet filter. The developer does not have to package the servlet filter with each application.

However, the developer will still need to modify the deployment descriptor and package the error pages with each application.

The real advantage is having only one servlet filter to manage. If changes to the servlet filter are made, the developer can simply replace the servlet filter jar file, rather than repackaging every application and redeploying.

See the WAS documentation for details on creating and using a shared library.

This document comes with a jar file (EFBAFilter.jar) containing only the servlet filter. This jar file can be used in a shared library.

Sample Applications

This document is accompanied by sample applications. They are built against the Java EE 5.0 specification level.

FilteredEFBASample.ear	Sample EFBA Application
EFBAFilter.jar	Jar file containing the EFBA servlet filter. This can be used as part of a shared library.
StandardFBASample.ear	Sample standard form-based authentication application.
StandardFBASample-secure.ear	Sample standard form-basedauthentication application – SSL required

Questions or Comments?

Please direct any correspondence to:

Lee-Win Tai – tai@us.ibm.com
Michael Kearney – kearney@us.ibm.com
Mike Loos – mikeloos@us.ibm.com