

# Flexess: Installation instruction

## General notes

- The Flexess requires the **Java 5.0** or higher.
- To use Flexess Security Modeler you have to obtain the **license**. Trial license can be obtained automatically in Flexess security modeler or requested manually on the <http://www.jresearchsoft.com> site after registration.

## Runtime requirements

The Flexess can be downloaded as Standalone server or WAR distribution.

The Standalone distribution based on Jetty application server and uses the Derby database. That distribution virtually not required manual configuration and can be used in any environment with JDK 5.0 or higher.

The WAR distribution is suitable for customers with existing application server environment. It required the database creation and manual configuration (at least some deployment steps).

## Supported application servers

The following application servers were tested with Flexess server:

Jetty 6, Apache Tomcat 5.5, Apache Tomcat 6.0, JBoss AS 4.0, JBoss AS 4.5, Sun Glassfish v2, Bea Weblogic 10, IBM WebSphere 6.1, IBM WebSphere Community Edition. See [Appendix A](#) for container specific documentation.

Virtually the WAR distribution can be deployed on any servlet container that supports Servlet 2.4/JSP 2.0.

## Supported database servers

The following application servers were tested with Flexess server:

Oracle Database 10g and 9, Microsoft SQLServer 2000 and 2005, PostgreSQL 8.0 and 8.1, MySQL Community Server 5.0, Apache Derby 10.

See [Appendix B](#) for database specific information.

## Flexess server

Download Standalone (Jetty AS) or WAR (general J2EE AS) "Flexes server" package from <http://www.jresearchsoft.com/site/docList.do>.

Unpack it to any place on hard drive (e.g. c:\flexess or /usr/local/flexess).

### Standalone server:

Go to C:\flexess\server\bin (/usr/local/flexess/server/bin) and run run.bat (./run). Server will start on the port 8080

Open browser and go to <http://localhost:8080/flexess>

The default login/password admin/admin

If you are using a machine with an operating system that is a derivative of UNIX™, you may set the execute permission for the server startup script: `chmod -R +x bin/run`

To change the default port edit `etc\jetty.xml` (locate and change the following string “<SystemProperty default="8080" name="jetty.port"/>”

## Flexess server WAR:

### Folder structure

Go to `C:\flexess\server\war (/usr/local/flexess/server/war)`

After unpacking you will have the following structure on disk:

```
dist
edit
tools
webapp
build
build.bat
build.xml
```

The WAR file can be build by `build.bat (build.sh)`. All necessary changes should be performed in the **edit** folder. **Never** change the files in `webapp` folder. Copy necessary files from `webapp` to `edit`. They have the same structure. The build procedure will take content of `webapp` and then override and update it from `edit`. The `dist` folder will contain WAR file with all your settings.

### Database configuration

The application server is responsible for settling up the database connection. The things to know are JNDI name and the type of the database to create correct SQL commands. Flexess is shipped with integrated SUM, which uses the separate database. If you are going to use separate SUM then we recommends to setup two databases: one for Flexess and another for SUM.

To setup each database you need the following things:

1. Create database <dbname> on the database server.
2. Create a database user <dbuser> with the password <dbpassword>.
3. Grant to <dbuser> all the necessary access rights to <dbname>, **including the rights for schema creation and manipulation!**
4. Setup JNDI datasource in your application server. See documentation for the application server about how to do this. Basically you need to download and install the JDBC driver and specify the driver class name, jdbc url, <dbuser> and <dbpassword>.

Flexess data layer works through Hibernate object-relational mapping (<http://www.hibernate.org/>). For proper database configuration you need to edit `app.properties` (for Flexess database) and `app-sum.properties` (for SUM database) files in `edit\WEB-INF\classes` folder.

- `hibernate.dialect` – Hibernate dialect, which depends on the selected database server and specifies the particular SQL variant Hibernate generates (for example, `org.hibernate.dialect.OracleDialect`). Below we have a list of all

hibernate dialects, with which we tested the server.

- ds.jndiname – JNDI name of the datasource. (for example, java:comp/env/jdbc/uamDS).

Flexess is tested with the following dialects:

|  |                                   |   |
|--|-----------------------------------|---|
| org.hibernate.dialect.OracleDialect      | Oracle 10g                        | No problems detected  |
| org.hibernate.dialect.Oracle9Dialect     | Oracle 9                          | No problems detected  |
| org.hibernate.dialect.SQLServerDialect   | Microsoft SQLServer 2000 and 2005 | Flexess does not work with the Microsoft SQLServer 2000 Driver for JDBC. The apperared problem is described in <a href="http://support.microsoft.com/kb/824106">http://support.microsoft.com/kb/824106</a> . With jTDS JDBC driver ( <a href="http://jtds.sourceforge.net/">http://jtds.sourceforge.net/</a> ) no problems were detected. |
| org.hibernate.dialect.DerbyDialect       | Apache Derby 10                   | No problems detected  |
| org.hibernate.dialect.PostgreSQLDialect  | PostgreSQL 8.0 and 8.1            | No problems detected  |
| org.hibernate.dialect.MySQLInnoDBDialect | MySQL Community Server 5.0        | No problems detected  |

### ***User Management configuration***

Flexess include integrated User Manager out of box. You don't need to configure anything to use it.

### ***Client Library***

Download “Flexess client” pakage from <http://www.jresearchsoft.com/site/docList.do> site. The package contains the com.jresearchsoft.uam.client.api.jar and all necessary jars. Add them to your project build path.

### ***Flexess Security Modeler***

**You need to have a license to work with Flexess Security Modeler.** You can retrieve an evaluation license after modeler installation. Navigate to Window | Preferences | Flexess and press Evaluate button. The license will be installed automatically. You can also request the license on the <http://www.jresearchsoft.com> site in the customer section after simple registration and install it on the same preferences page.

“Flexess Security Modeler” is available as an Eclipse plugin. The plugin can be installed directly from Eclipse. Just add the Jresearch Discovery Site (<http://www.jresearchsoft.com/update>) to the Eclipse (see Step-by-Step Instruction below).

“Flexess Security Modeler” requires the following Eclipse.org products:

- ✓ **Platform Runtime Binary 3.2** (<http://download.eclipse.org/eclipse/downloads>)
- ✓ **JDT Runtime Binary 3.2** (<http://download.eclipse.org/eclipse/downloads>)
- ✓ **Eclipse Modeling Framework All-In-One SDK 2.2**  
(<http://www.eclipse.org/modeling/download.php?file=/modeling/emf/emf/downloads/drops/2.2.4/R200710030400/emf-sdo-SDK-2.2.4.zip>).

### **Step-by-Step installation instruction for Flexess Security Modeler**

- Download all required Eclipse products.
- Expand all these packages into the same folder
- Run Eclipse (C:\Eclipse\eclipse.exe)
- Open Find and Install wizard (Help | Software Updates | Find and Install...)
- Select "Search for new features to install"
- Add new remote site (<http://www.jresearchsoft.com/update>).
- Select the added site and press Next button.
- The Eclipse should find the Jresearch Discovery Site and show the latest Modeler.
- Select it and follow the Installation Wizard.
- After installation you should add the existing or request evaluation license in the Eclipse Preference dialog (Main Menu | Window | Preferences | Flexess).

### ***Flexess Example.***

Download “Flexess Example” package from <http://www.jresearchsoft.com/site/docList.do>.

Unzip it.

### **To open the example security model in modeler.**

- Start Flexess Security Modeler
- Create Eclipse project. In the menu "File | New | Project" select "Java project", press next and select "Create project from existing sources". Then select the folder with unpacked example.
- Right click on the project and select "Build path | Configure build path" action. On tab "Order and Export" add all the jars inside projects to the libraries.
- Set compiler compliance level. In the menu “Window | Preferences” select “Java | Compiler” and set compiler compliance level to 5.0
- Right click on the project and select "Flexess | Add Flexess Capabilities" to enable Flexess for the project.
- Double-click on "src/main/java/com.jresearchsoft.uam.examples.orders.conf file"

## To build war and deploy on the Jetty server

- Go to unpacked example and check the the hostname and port of the Flexess server in the **src\main\webapp\WEB-INF\classes\flexess-client.properties** file. By default the hostname is **localhost** and the port is **8080**.
- Execute command **ant all**
- Copy orders.war from a **build** folder (e.g. C:\Flexess\orders\build) to the server **webapp** folder (e.g. C:\Flexess\server\webapp)
- Restart Jetty.
- Open browser and go to <http://localhost:8080/orders>
- The login page has a links with predefined users. (Customer, EU Manager, Global manager). With Flexess Administrator/SUM you can create new users for Orders or modify existing.

## To deploy on a J2EE server

The example application is a general J2EE application and doesn't required any additional steps to deploy. Just build the WAR as described above (don't forget to check URL of Flexess server) and follow the standard deploy procedure for your J2EE application server.

The example uses in memory database and regenerate all data on startup. So any changes wouldn't persist between application server restarts.

## Standalone Simple User Management (SUM)

This distribution allows you to setup the standalone SUM server. The Standalone SUM use the Flexess to control access rights, so before using standalone SUM you **need** to setup **Flexess server** and create new application for Standalone SUM on it.

1. Select some application name (e.g. standalone\_sum)
2. Create on Flexess server the application with selected name (standalone\_sum), SUM model (com.jresearchsoft.uam.sum) and some existing UMI (e.g. embedded SUM)
3. Select the "standalone\_sum" application, and assign to some user administration rights.
4. Deploy the Standalone SUM as described bellow, don't forget update the flexess-client.properties file with application name (standalone\_sum) and Flexess server URL.

Now you can login to Standalone SUM with some user account (see step 3). If you selected embedded SUM and user admin you can use admin/admin account.

## To use the standalone SUM in other applications as User Management System

You need to create new UMI connection pointed to the new standalone SUM installation (see "Setup the External Simple User Management connection" section in the "Flexess Developer's Guide-1.0" for detail) and associate it with applications which will use the standalone SUN.

## To deploy standalone SUM

To setup standalone SUM you need to download it from <http://www.jresearchsoft.com/site/docList.do> and unpack it. The installation procedure is very similar with [Flexess WAR installation](#). Check the database settings. Setup JNDI on application server. **Check the Flexess URL and application name in editWEB-**

**INF\classes\flexess-client.properties** file. Build the war and deploy it. All container and database specific issues from appendixes can be applied to the standalone SUM installation with minimal correction in JNDI names, context name and URL's.

## To change JNDI Names

If you want to deploy SUM on the same server with Flexess or another SUM you may need to change JNDI names (it depends from server). There are two parts – application JNDI names and application server JNDI names.

By default the SUM uses the following JNDI name: **java:comp/env/jdbc/uam\_users\_DS**. Assume you want to change it to **java:comp/env/jdbc/uam\_users\_DS01**.

**Note:** You can choose any name, but we strongly recommended don't change the “**java:comp/env/jdbc**” part.

To change the JNDI names on the application side you need to edit the following files:

- **edit\WEB-INF\classes\app-sum.properties** – change value of *sum.datasource.jndiname* property to **java:comp/env/jdbc/uam\_users\_DS01**
- **edit\WEB-INF\web.xml** – locate the tag `<resource-ref/>` and change value of `<res-ref-name/>` subtag to **jdbc/uam\_users\_DS01**

Changing the JNDI name on the application server is very depend from application server:

- The Sun **Glassfish** v2, Bea **Weblogic** 10, IBM **WebSphere** 6.1 Express Edition provide the management tools to setup and change the JNDI resources – please, read the appropriate server documentation.
- For Apache **Tomcat** 5.5, Apache Tomcat 6.0 you need to edit the **edit\META-INF\context.xml** – change the value of attribute “*name*” to **jdbc/uam\_users\_DS01**
- For **JBoss** AS 4.0, JBoss AS 4.5 you can use the template file **stand-sum-ds.xml** from **support\jboss** folder. Change the content of `<jndi-name/>` tag to **jdbc/uam\_users\_DS01** and copy the file to Jboss deployment folder.
- For **Jetty** 6 edit the file **edit\WEB-INF\jetty-env.xml**. Change the id of `<new/>` tag and content of the first `<Arg/>` tag to **jdbc/uam\_users\_DS01**
- For IBM **WebSphere** 6.1 Community Edition edit the file **edit\WEB-INF\geronimo-web.xml**. Change the content of the `<ref-name>` tag to **jdbc/uam\_users\_DS01**

## Appendix A

### Container specific instructions

#### WebLogic 10

Web site:

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/server/>

Trial:

[http://commerce.bea.com/showproduct.jsp?family=WLS&major=10&minor=0&DL=www\\_WLS icon&WT.ac=DL\\_www\\_WLS icon](http://commerce.bea.com/showproduct.jsp?family=WLS&major=10&minor=0&DL=www_WLS_icon&WT.ac=DL_www_WLS_icon)

Assume that Flexess WAR distribution was unpacked into the **flexess\dist** folder

1. Copy suitable JDBC driver to *domain\_name\lib* folder (in my case it was

C:\bea\wlserver\_10.0\samples\domains\wl\_server\lib). For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.

2. Start Weblogic server (*domain\_name\startWebLogicEx.cmd* or *.sh*) and open the WebLogic console (<http://localhost:7001/console>)
3. Create Data Source(s) for your database: Left tree > wl\_server > Services > JDBC > Data Sources (*different providers may be used if you are going to use separate database for Flexess and integrated SUM*)
  1. Click on Lock & Edit
  2. Click on New
  3. Enter any name
  4. JNDI Name: jdbc/uamDS and jdbc/uam\_users\_DS
  5. Select Database type
  6. Click Next
  7. Click Next
  8. Database Name: e.g. flexess
  9. Host Name: e.g. localhost
  10. Database User Name: e.g. flexess
  11. Password: e.g. flexess
  12. Click Next
  13. Review, Test and Next
  14. Select the target server (e.g. exampleServer) and click Finish
  15. Activate Changes.
4. Uncomment the right Hibernate dialect (see comments inside property files) in ***flexess\dist\edit\WEB-INF\classes\app.properties*** and ***flexess\dist\edit\WEB-INF\classes\app-sum.properties*** files. Check if no more dialects are uncommented.
5. Run build.bat(.sh) in the ***flexess\dist*** folder.
6. Deploy ***flexess\dist\distflexess.war*** to WebLogic (Left tree > wl\_server > Deployments)
  1. Click on **Lock & Edit**
  2. Click on **Install** button
  3. Select Location in table: ***flexess\dist\distflexess.war*** and click **Next**
  4. Leave selection on “*Install this deployment as an application*” and click **Next**
  5. Leave all settings as are and click **Finish**.
  6. **Activate Changes**.
  7. Select flexess in Deployments table and click **Start** (Serving all requests) and then **Yes**
7. You can access Flexess at <http://localhost:7001/flexess>

## Jetty 6.1

Web site: <http://www.mortbay.org>

Download: <http://dist.codehaus.org/jetty/jetty-6.1.7>

Assume that Flexess WAR distribution was unpacked into the **flexess\dist** folder. To run Flexess on Jetty you need to **enable plus** options. To do it you need to uncomment the "webapps-plus" settings in `%jetty_home%\etc\jetty-plus.xml`, create `webapps-plus` folder in `%jetty_home%` and use `java -jar start.jar etc/jetty.xml etc/jetty-plus.xml` command to run the Jetty server (see more information on <http://docs.codehaus.org/display/JETTY/JNDI>).

1. Copy suitable JDBC driver to `%jetty_home%\lib` folder (in my case it was `C:\jetty-6.1.7\lib`). For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.
2. **Move** `xmldsig-1.0.jar` and `xmlsec-2.0.jar` to `%jetty_home%\lib` folder from `flexess\dist\webapp\WEB-INF\lib`. (Be sure that you **MOVE** these jars)
3. Edit the `flexess\dist\edit\WEB-INF\jetty-env.xml`. That file contains bindings from JNDI names to physical database. By default it contains the settings for **Postgres** database "flexess" on the `localhost` with user `flexess` and password `flexess`
4. Uncomment the right Hibernate dialect (see comments inside property files) in `flexess\dist\edit\WEB-INF\classes\app.properties` and `flexess\dist\edit\WEB-INF\classes\app-sum.properties` files. Check if no more dialects are uncommented.
5. Run `build.bat(.sh)` in the `flexess\dist` folder.
6. Deploy `flexess\dist\dist\flexess.war` to Jetty (copy it to `%jetty_home%\webapps-plus` folder) and start the server.
7. You can access Flexess at <http://localhost:8080/flexess>

## Apache Tomcat 5.5 and 6.0

Web site: <http://tomcat.apache.org>

Download: <http://tomcat.apache.org/download-55.cgi> or <http://tomcat.apache.org/download-60.cgi>

For Flexess server is enough the core distribution.

Assume that Flexess WAR distribution was unpacked into the **flexess\dist** folder.

1. Copy suitable JDBC driver to `%tomcat_55_home%\common\lib` or `%tomcat_60_home%\lib` folder. For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.
2. Edit the `flexess\dist\edit\META-INF\context.xml`. That file contains bindings from JNDI names to physical database. By default it contains the settings for **Postgres** database "flexess" on the `localhost` with user `flexess` and password `flexess`
3. Uncomment the right Hibernate dialect (see comments inside property files) in `flexess\dist\edit\WEB-INF\classes\app.properties` and `flexess\dist\edit\WEB-INF\classes\app-sum.properties` files. Check if no more dialects are uncommented.
4. Run `build.bat(.sh)` in the `flexess\dist` folder.
5. Deploy `flexess\dist\dist\flexess.war` to Tomcat (copy it to `%tomcat_home%\webapps` folder) and start the server

(*%tomcat\_xx\_home%\bin\catalina.bat(.sh) start*).

6. You can access Flexess at <http://localhost:8080/flexess>

## Jboss 4.0.5 and 4.2.2

Web site: <http://labs.jboss.com>

Download: <http://labs.jboss.com/jbossas/downloads>

Assume that Flexess WAR distribution was unpacked into the *flexess\dist* folder.

1. Copy suitable JDBC driver to *%jboss\_home%\server\default\lib* folder. For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.
2. Create the Jboss data source descriptions. You can use sample descriptions from *flexess\dist\support\jboss* as a base. Copy the data source descriptions to *%jboss\_home%\server\default\deploy*
3. Uncomment the right Hibernate dialect (see comments inside property files) in *flexess\dist\edit\WEB-INF\classes\app.properties* and *flexess\dist\edit\WEB-INF\classes\app-sum.properties* files. Check if no more dialects are uncommented.
4. Run build.bat(.sh) in the *flexess\dist* folder.
5. Deploy *flexess\dist\dist\flexess.war* to Jboss (copy it to *%jboss\_home%\server\default\deploy* folder) and start the server
6. You can access Flexess at <http://localhost:8080/flexess>

## Sun GlassFish v2

Web site: <https://glassfish.dev.java.net>

Download: <https://glassfish.dev.java.net/downloads/v2ur1-b09d.html>

Installation instruction: On download page.

Assume: Flexess WAR distribution was unpacked into the *flexess\dist* folder (e.g. *c:/flexess/dist*) and default domain of application server is **domain1**. If you have another domain just change all entrance of "*domain1*" to your.

1. Copy suitable JDBC into the *%glassfish\_home%\domains\domain1\lib\ext* folder. For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.
2. Start Glassfish server (*%glassfish\_home%\bin\asadmin.bat start-domain domain1*). By default admin interface is available at <http://localhost:4848>
3. Create Connection Pool(s) for database (different pools may be used if you are going to use separate database for Flexess and integrated SUM).  
Here are the sample steps for PostgreSQL database:
  1. Left tree>Resources>JDBC>Connection Pools
  2. Name: FlexessPool
  3. Resource Type: javax.sql.DataSource
  4. Database Vendor: PostgreSQL
  5. On the **next page** in Additional Properties table fill-in:
    1. databaseName (e.g. flexess)

2. password (e.g. flexess)
3. serverName (e.g. localhost)
4. user (e.g. flexess)
6. All other fields left as is
4. Create JDBC Resources (Left tree>Resources>JDBC>JDBC Resources) with name jdbc/uamDS and jdbc/uam\_users\_DS (use the pool(s) from the step 3)
5. Uncomment the right Hibernate dialect (see comments inside property files) in **flexess\dist\edit\WEB-INF\classes\app.properties** and **flexess\dist\edit\WEB-INF\classes\app-sum.properties** files. Check if no more dialects are uncommented.
6. Run build.bat(.sh) in the **flexess\dist** folder.
7. Deploy **flexess\dist\dist\flexess.war** to Glassfish
  1. Left tree>Applications>Web Applications>Deploy
    1. Browse to the WAR file (select **local** if you have WAR on the same computer as the Glassfish server or **upload** otherwise)
    2. Application Name: flexess
    3. Context Root: flexess
    4. All other fields left as is
  2. Press "Ok" button
  3. Wait while glassfish deploy the application.
8. You can access Flexess at <http://localhost:8080/flexess>

## IBM WebSphere 6.1.0.11

Web site: <http://www-306.ibm.com/software/websphere>

Download:

[http://www.ibm.com/developerworks/downloads/ws/wasexp/?S\\_TACT=105AGX28&S\\_CMP=TRIALS](http://www.ibm.com/developerworks/downloads/ws/wasexp/?S_TACT=105AGX28&S_CMP=TRIALS)

**Note: After deploy the Flexess or any other product with flexess-client.jar you need to switch class loader to PARENT\_LAST mode for deployed module.**

Assume: Flexess WAR distribution was unpacked into the **flexess\dist** folder, default **server1**. If you have another server just change all entrance of "server1" to your.

1. Copy suitable JDBC driver to into the  $\${WAS\_LIBS\_DIR}$  (%ws\_home%\lib) folder. For the list of recommended JDBC drivers see [Recommended JDBC drivers](#) section.
2. Start WebSphere server (**startServer.bat server1**). By default admin interface is available at <https://localhost:9043/ibm/console/>
3. Create JDBC provider(s) for your database (Left tree>Resources>JDBC>JDBC Providers). See [http://www.jresearchsoft.com/blog/flexess/entry/postgresql\\_with\\_websphere](http://www.jresearchsoft.com/blog/flexess/entry/postgresql_with_websphere) to sample PostgreSQL steps.
4. Uncomment the right Hibernate dialect (see comments inside property files) in **flexess\dist\edit\WEB-INF\classes\app.properties** and **flexess\dist\edit\WEB-**

**INFclasses\app-sum.properties** files. Check if no more dialects are uncommented.

5. Run build.bat(.sh) in the **flexess\dist** folder.
6. Deploy **flexess\dist\distflexess.war** to WebSphere
  1. Left tree>Applications>Install New Application
    1. Location: **flexess\dist\distflexess.war**
    2. Context Root: flexess
    3. Select "Prompt me only when additional information is required".
  2. Press "Next"
  3. Leave all settings unchanged (Step 1) and press "Next"
  4. Leave all settings unchanged (Step 2) and press "Next"
  5. Edit the resource mapping and authentication settings (Step 3).
    1. In the resource table browse to corresponded resource JNDI names (you can just copy value from "Resource Reference" column to input field in the "Target Resource JNDI Name" column.
    2. Then select all check boxes in the table and select appropriate authentication method. You can use the default and select "Authentication data entry" created for database access (see [http://www.jresearchsoft.com/blog/flexess/entry/postgresql\\_with\\_websphere](http://www.jresearchsoft.com/blog/flexess/entry/postgresql_with_websphere) for more information).
    3. Click "Apply" - the table should be updated.
  6. Press "Next"
  7. Leave all settings unchanged (Step 4) and press "Next"
  8. Review settings and press "Finish"
  9. Wait while WebSphere deploy the application.
  10. Click on the "save" link.
  11. Select in application list (Left tree>Applications>Enterprise Applications) deployed application, select Manage Modules link, in the table select module (there should be only one) and select "**Classes loaded with application class loader first**".
  12. Check if deployed application started (Left tree>Applications>Enterprise Applications)
7. You can access Flexess at <http://localhost:9080/flexess>

## **IBM WebSphere Application Server Community Edition 2.0**

Web site: <http://www.ibm.com/developerworks/websphere/zones/was/wasce.html>

Download: <http://www.ibm.com/developerworks/downloads/ws/wasce/>

Java tuning: <http://publib.boulder.ibm.com/wasce/V2.0.0/en/tuning-java.html>

**Note:** We recommend you to have JAVA\_OPTS environment variable defined as

JAVA\_OPTS=-Xmx384m -XX:+CMSSClassUnloadingEnabled -XX:MaxPermSize=256m  
before the running of application server to avoid the "PermGen space" error.

Assume: Flexess WAR distribution was unpacked into the **flexess\dist** folder

1. Start WebSphere CE server :

UNIX: `$SERVER_INSTALL_DIR$/bin$ ./startup.sh`

MS Windows: On the Windows task bar, click Start, All Programs, IBM WebSphere, Application Server Community Edition, Start the server. This invokes the server startup script.

A new command window opens and displays a list of ports on which the server is listening, application modules started, active web application URLs, and should end with the server started message. Do not close this window or logoff as this will stop the server.

Navigate your browser to the administrative console at <http://localhost:8080/console> and log on using:

- \* Default username = system
- \* Default password = manager

2. Select *Console Navigation:Services:Database Pools* and create a new database pool with name **flexess\_ds** using the "Using the Geronimo database pool wizard" link.
3. Uncomment the right Hibernate dialect (see comments inside property files) in **flexess\dist\edit\WEB-INF\classes\app.properties** and **flexess\dist\edit\WEB-INF\classes\app-sum.properties** files. Check if no more dialects are uncommented.
4. Run build.bat(.sh) in the **flexess\dist** folder.
5. Deploy **flexess\dist\dist\flexess.war** to WebSphere CE  
*Console Navigation:Applications:Deploy New*
  1. Select archive by location: **flexess\dist\dist\flexess.war**
  2. Select "Start app after install".
  2. Press "Install"
6. Check if deployed application started (*Console Navigation :Applications:Web App WARs*)
7. You can access Flexess at <http://localhost:8080/flexess> (In case of default server installation )

## Appendix B

### *Database specific instructions*

#### **Microsoft SQLServer 2000 and 2005**

Flexess does not work with the Microsoft SQLServer 2000 Driver for JDBC. The appeared problem is described in <http://support.microsoft.com/kb/824106>. With jTDS JDBC driver (<http://jtds.sourceforge.net/>) no problems were detected.

#### **Recommended JDBC drivers**

| Database                          | Driver   | Comment                             |
|-----------------------------------|--|-------------------------------------|
| Oracle Database 10g and 9         | Oracle JDBC Thin Driver<br>( <a href="http://download.oracle.com/otn/utilities_drivers/jdbc/10203/ojdbc14.jar">http://download.oracle.com/otn/utilities_drivers/jdbc/10203/ojdbc14.jar</a> )   | You should have account on OTN      |
| Microsoft SQLServer 2000 and 2005 | jTDS JDBC driver<br>( <a href="http://sourceforge.net/project/showfiles.php?group_id=33291">http://sourceforge.net/project/showfiles.php?group_id=33291</a> )  |                                     |
| PostgreSQL 8.0 and 8.1            | PostgreSQL JDBC 3 Driver<br>( <a href="http://jdbc.postgresql.org/download/postgresql-8.3-603.jdbc3.jar">http://jdbc.postgresql.org/download/postgresql-8.3-603.jdbc3.jar</a> or later)  |                                     |
| MySQL Community Server 5.0        | MySQL Connector/J 5.1<br>( <a href="http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.5.tar.gz/from/pick#mirrors">http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.5.tar.gz/from/pick#mirrors</a> or later) |                                     |
| Apache Derby 10                   | Embedded driver<br>( <a href="http://db.apache.org/derby/derby_downloads.html">http://db.apache.org/derby/derby_downloads.html</a> )   | JDBC Driver embeded in database JAR |