|  |  |
| --- | --- |
| AMCAD Logo RGB 72 dpi.png  www.amcad.com | **AMCAD eCommerce Deployment Guide**  **va_court_2** |

# Document Revisions

|  |  |  |
| --- | --- | --- |
| **Date** | **Editor** | **Description of Change** |
| 12/7/2011 | Vivekanand  Alampally | 1. Encrypting & Decrypting of Configuration properties using Jasypt. 2. Configuration Properties to be changed in ecom at the time of Deployment. 3. Configuration Properties to be changed in ecomadmin at the time of Deployment. |
|  |  |  |
|  |  |  |

# 

**Table of Contents**

Document Revisions 2

0 Overview 4

1 Prerequisites: 4

2 Configuration Settings of server.properties in ecom 5

3 Configuration Settings of server.properties in ecomadmin 8

4 Encrypting & Decrypting of Configuration Properties using Jasypt 8

# Overview

The purpose of this document is to explain the steps involved, in the deployment of ecomadmin and ecom projects. Before doing the deployment, it is needed to change configuration properties in each project suitable to the environment and this document thus mainly explains the following things.

1. Configuration Properties to be changed in ecom at the time of Deployment.
2. Configuration Properties to be changed in ecomadmin at the time of Deployment.
3. Encrypting & Decrypting of configuration properties using Jasypt.

**Important Note:** In this document where ever words are marked with RED then those properties should be changed at the time of Deployment.

# Prerequisites:

1. Install latest version of Java
2. Set **JAVA\_HOME** environment variable. It should refer the base folder of java.

For example, **C:\Program Files\Java\jdk1.7.0\_02**.

1. Modify the existing **PATH** environmental variable by adding the following at the end.

**;C:\Program Files\Java\jdk1.7.0\_02\bin;**

Open a command prompt and type **java –version** to veify that the PATH variable has been set

1. Install JCE jars for unlimited Key stength .
2. Download Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7.
3. Extract the downloaded file. You will find **local\_policy.jar** , **US\_export\_policy.jar**.
4. Replace the jar files present in underlying jre’s lib\security folder with the jar files in extracted folder. For example, **C:\Program Files\Java\jdk1.7.0\_02\jre\lib\security.**
5. Install Apache Tomcat as a service.
6. Download latest version of Apache Tomcat.
7. Extract that in C:. Open command prompt with administrative privileges and then go to bin folder in extracted folder

For example, **cd C:\apache\tomcat-7.0.23\bin**

1. Now type,

**service.bat install Tomcat7**

where Tomcat7 is the service name.

1. Install Apache ActiveMQ as a service.
2. Download latest version of ApacheMQ.
3. Extract that in C: . Then go to bin folder in extracted folder. In bin you will see Win32 and, in Win32 open **wrapper.conf** and add the following line at the end of file.

**wrapper.java.additional.10=-Djava.io.tmpdir="%ACTIVEMQ\_BASE%"/temp**

1. Now run **InstallService.bat** file and MQService will be installed.

# Configuration Settings of server.properties in ecom

This Section explains what settings in server.properties needs to be changed before deploying ecom. Here you will notice that some properties like jdbc.username, jdbc.password, and etc are encrypted. To get the encrypted value of a string, here we are using an open source Jasypt library. The process of encrypting a string is explained in **section 4**.

In ecom, inside the **applicationContext.xml**, find the below configuration. Here, we can mention the algorithmand **password** of your choice. Here in this case, we are using ***PBEWITHSHA256AND256BITAES-CBC-BC***algorithm and ***dookudu*** aspassword. *PBEWITHSHA256AND256BITAES-CBC-BC*is a very strong & efficient algorithm, so keep that as it is and if you want to customize the behavior change the password instead.

<bean id=*"propertyPlaceholderConfigurer"* class=*"org.jasypt.spring.properties.EncryptablePropertyPlaceholderConfigurer"*>

<constructor-arg>

<bean class=*"org.jasypt.encryption.pbe.StandardPBEStringEncryptor"*>

<property name=*"config"*>

<bean class=*"org.jasypt.encryption.pbe.config.EnvironmentStringPBEConfig"*>

<property name=*"algorithm"* value=*"PBEWITHSHA256AND256BITAES-CBC-BC"* />

<property name=*"password"* value=*"****dookudu****"* />

</bean>

</property>

<property name=*"provider"*>

<bean class=*"org.bouncycastle.jce.provider.BouncyCastleProvider"* />

</property>

</bean>

</constructor-arg>

<property name=*"locations"* >

<list>

<value>/WEB-INF/conf/spring/server/properties/server.properties</value>

</list>

</property>

</bean>

What the above configuration actually does is as soon as application get deployed, it looks for **server.properties** and it fetches all the properties with values starting with **“ENC(“** and decrypts those strings with the above specified algorithm and password.

Now in server.properties of ecom

1. The JDBC settings are specified as follows.

jdbc.drivername=com.microsoft.sqlserver.jdbc.SQLServerXADataSource

jdbc.url=jdbc:sqlserver://NV-DEVSQL7.amcad.com;instanceName=VELMA;databaseName=WAM\_DEV

jdbc.username=ENC(Ap4KwSD2iFtDGapTzAkDElprqo/aQSsLiJIA3p0ASS8=)

jdbc.password=ENC(cCvVPAl2sPMUP/PTnoJPB/xJST6tnBEDaLzklBmNs7c=)

jdbc.minPoolSize=1

jdbc.poolSize=2

jdbc.maxPoolSize=3

jdbc.borrowConnectionTimeout=60

jdbc.url is the connection URL and here you should specify the database name which you would like your application to point to.

Here we are pointing to **WAM\_DEV** database.

And, jdbc.username & jdbc.password are username & password of underlying database. It is better if these properties are encrypted(Look **Section 4**), so that application will be more secure.

1. The Scheduler settings are as follows,

Below settings follows cron format. “0 0 \* \* \* ?” means scheduler runs at start of every hour.

scheduler.ecom.verifypaypalprofile=0 0 \* \* \* ?

scheduler.ecom.cancelpaypalprofile=0 0 \* \* \* ?

scheduler.ecom.deletewebpurchases=0 0 \* \* \* ?

1. These are the E-Mail Settings, Here emailservice.username takes **AD username** (i.e. For example: jsmith not [jsmith@amcad.com](mailto:jsmith@amcad.com))

emailservice.emailserver=mail.amcad.com

emailservice.emailport=25

emailservice.username=ENC(URfFvSrlSWFcF9B6GH3W7M8psMNunSCxAb0dyAgfte0=)

emailservice.password=ENC(mm9k16CPzKcJSQqpBjB3BBKVqaV3zoDbCicnKLHYXTg=)

Note: Sometimes emailservice.username & emailservice.password can be omitted. If omitted, remove these values from applicationContext.xml also.

For Example:

<bean name=*"mailSender"* class=*"org.springframework.mail.javamail.JavaMailSenderImpl"*

p:host =*"${emailservice.emailserver}"*

p:port =*"${emailservice.emailport}"*

~~p:username=~~*~~"${emailservice.username}"~~*

~~p:password=~~*~~"${emailservice.password}"~~*

p:javaMailProperties-ref=*"javaMailProperties"*/>

1. These are the Web service settings. It is protected with username & password, so that general public cannot access this.

webservice.username=ENC(Ap4KwSD2iFtDGapTzAkDElprqo/aQSsLiJIA3p0ASS8=)

webservice.password=ENC(cCvVPAl2sPMUP/PTnoJPB/xJST6tnBEDaLzklBmNs7c=)

1. Paypal Settings can be specified here.

ecommerce.paypal.paypaladminemailid=valampally@amcad.com

ecommerce.paypal.paypalhostaddress=pilot-payflowpro.paypal.com

ecommerce.paypal.paypalhostport=443

ecommerce.paypal.paypaltimeout=60

ecommerce.paypal.paypallogfilename=C\:\\logs\\PayFlow.log

#5 MB Log File Size

ecommerce.paypal.paypallogfilesize=5120000

1. activemq.url should be changed to tcp://localhost:61616

**Note:** If the above property is set with tcp: then MQService should be started before deploying the project. If you don’t want to start MQService set the above property as vm://localhost:61616

# Configuration Settings of server.properties in ecomadmin

In ecomadmin both in **applicationContext.xml** & **sdl-servlet.xml** you need to change the algorithm and **password.**

Now in **server.properties of ecomadmin**, You need to change the following settings apart from above mentioned settings.

1. Here, localhost should be replaced by DNS name of server where ecomadmin is deployed.

email.useractivation.requesturl= <http://localhost:7001/ecomadmin/>

1. And also, you need to specify the correct URL where ecom is deployed,

ecom.facadeservice.wsdl=http://localhost:8080/ecom/EComFacadeService?wsdl

ecom.facadeservice.address=http://localhost:8080/ecom/EComFacadeService

Same thing here also, localhost should be replaced by DNS name of server where ecom is deployed.

# Encrypting & Decrypting of Configuration Properties using Jasypt

To encrypt a string,

1. First, identify java\_home, whether the system is using jdk or jre and find the folder where it is installed.

This can be done by issuing the following command in the command prompt.

**echo %JAVA\_HOME%**

If it references the jdk directory here, then system is using JDK, Otherwise jre.

1. Second, Copy bountycastleprovider jars (**bcmail-jdk16-146.jar, bcprov-jdk16-146.jar** - these are present in EncryptDecrypt\lib\bounty-castle folder) to above identified java path and under jre/lib/ext folder.
2. Add the following line in java.security file under List of providers which is in java installation folder under jre/lib/security folder.

**security.provider.11=org.bouncycastle.jce.provider.BouncyCastleProvider**

1. Go to bin folder of EncryptDecrypt

Issue the following command on command prompt,

Syntax:

**encrypt input=**<Plain text message that has to be encrypted>  **password**=<Some password which should be remembered> **algorithm=** **PBEWITHSHA256AND256BITAES-CBC-BC**  **providerName=BC**

For example:

C:\Projects\EncryptDecrypt\bin>**encrypt input=developer password=dookudu algorithm=PBEWITHSHA256AND256BITAES-CBC-BC providerName=BC**

**----ENVIRONMENT-----------------**

**Runtime: Sun Microsystems Inc. Java HotSpot(TM) 64-Bit Server VM 20.2-b06**

**----ARGUMENTS-------------------**

**algorithm: PBEWITHSHA256AND256BITAES-CBC-BC**

**input: developer**

**password: dookudu**

**providerName: BC**

**----OUTPUT----------------------**

**fzr3qlQOPbJQqUGE6RAGOMKynIW8G+a1vEePZ3FP2nQ=**

**C:\Projects\EncryptDecrypt\bin>**

It is to be noted that, while specifying the encrypted string in properties file, specify it in **ENC().**

**Example : ENC(fzr3qlQOPbJQqUGE6RAGOMKynIW8G+a1vEePZ3FP2nQ=)**

The Output generated here is encrypted string of “**developer**”.

Now if you want to specify **developer** as the username of database, simply copy & paste the above generated string beside **jdbc.username** in server.properties file. Follow the same procedure for other configuration properties which you wish to encrypt, by keeping algorithm, password & providerName as constant.

In the same way, decryption can also be done, with decrypt command.

**decrypt input=<Encrypted String > password=<password which is supplied at the time of encryption> algorithm= PBEWITHSHA256AND256BITAES-CBC-BC providerName=BC**

For example:

C:\Projects\EncryptDecrypt\bin>**decrypt input=fzr3qlQOPbJQqUGE6RAGOMKynIW8G+a1vEePZ3FP2nQ= password=dookudu algorithm=PBEWITHSHA256AND256BITAES-CBC-BC providerName=BC**

**----ENVIRONMENT-----------------**

**Runtime: Sun Microsystems Inc. Java HotSpot(TM) 64-Bit Server VM 20.2-b06**

**----ARGUMENTS-------------------**

**algorithm: PBEWITHSHA256AND256BITAES-CBC-BC**

**input: fzr3qlQOPbJQqUGE6RAGOMKynIW8G+a1vEePZ3FP2nQ=**

**password: dookudu**

**providerName: BC**

**----OUTPUT----------------------**

**developer**

**C:\Projects\EncryptDecrypt\bin>**