# CMS Build Manual

## Version 1.1

PSL TEAM

7/18/2012

# Contents

# 1 Technologies Used in CMS

This document briefs about the technologies used in CMS and high level explanation about how each technology is used

## 1.1 Technologies

1. **Spring framework(2.5) (http://www.springsource.org/documentation)**

    a. Used for as MVC frame work for web based application development
    Various spring based web controllers are entry point for of processing of a user request. Following are few controllers CMS have used –

    - AddNewUserController – Collects the information provided by used on "Add user" screen. Delegates it to appropriate service class to add user into the system
    - ExecuteCommandController – Collects command and its parameters entered by user on UI, and delegates the call to appropriate service class to execute a command.

    b. Provides security to web application

2. **Blaze DS (4.0)( http://opensource.adobe.com/wiki/display/blazeds/Developer+Documentation)**

    a. Used to integrate flex with java based web application and is used to update the data on various flex based widgets in CMS. This library is used only for integration purposes and does not generate any output files as such. Refer to following link for step by step guide about integrating BlazeDS into web application –

    http://sujitreddyg.wordpress.com/2009/04/07/setting-up-blazeds/

3. **Hibernate (3.2.6)( http://www.hibernate.org/docs)**

    a. ORM framework, used for rapid development of database interactions with application. This framework is used to hide the database level complexities from application, facilitating rapid application development. This acts as an interface between application and actual database. Various DAO classes in CMS make use of this framework.

    Following are few configuration files used in CMS along with brief usage description –

    - Alarm.hbm.xml – Maps the Alarm object from application domain to "t_alarm" table on database side
    - User.hbm.xml - Maps the User object from application domain to "t_alarm" table on database side

4. **Flex 4.0 (http://www.adobe.com/devnet/flex.html)**

   It is used to create various widgets on CMS UI, which are then updated with dynamic data using push technology provided by BlazeDS.

   One can find files of following types in CMS for flex technology –

   a. Mxml files – These files end with extension ".mxml". These files are used to generate various flex based UI components in CMS. Following are such few files from CMS and corresponding flex components they correspond to –
      - PolarPlot.mxml – Used to create polar plot
      - ReceiverStatus.mxml – Used to generate Receiver status component
      - DetailedParamsEngg.mxml – Used to generate detailed parameters section in engineering UI

   b. Action Script files – These files are used to represent a data container object and are used inside mxml files to display the data. Following are such few files from CMS and corresponding flex components they correspond to –

      - HeaderObj.as – Used to hold the data related to CMS header component
      - TrackingStatusObj.as - Used to hold the data related to tracking status component

   All these files are compiled and packaged to create ".swf" extension files. These files are then embedded in html code to display corresponding component in CMS.

5. **Sleep Utility**
   a. A customized batch scripting framework. Modified to suit CMS requirements like
      - Executing commands from batch file
      - Conditional, looping logic

6. **Physhun 0.5.1 (http://physhun.sourceforge.net/)**
   a. Used for building and executing processes as finite State Models in J2SE and J2EE environments in CMS.

7. **MySQL (5.1)( http://dev.mysql.com/doc/refman/5.1/en/index.html)**
   a. Used as persistent store by CMS. CMS interacts with database via Hibernate framework through various DAO (Data Access Object) classes in CMS. Following are few DAOs used in CMS
      - AlarmDaoImpl – Contains code to create, update, read alarm records from database
      - UserDaoImpl - Contains code to create, update, read user records from database

8. **Maven 3.0 (http://maven.apache.org/)**
    a. Build tool to build entire application.

9. **Active MQ 5.4.3**
    a. Messaging framework used to en-queue and de-queue the messages to increase the scalability of overall CMS. Please refer to following URL for step by step information about Spring, ActiveMQ configuration.

       http://blog.springsource.com/2006/08/11/message-driven-pojos/

       Following are the classes from CMS involved in interaction with ActiveMQ –
        - Producer – Enqueues the message in ActiveMQ
        - RequestConsumer/ ResponseConsumer – Triggers the processing the message fetched from ActiveMQ

10. **Tomcat 6.0.x**
    a. Web server used to deploy and run web application

11. **Java 1.6.x**
    a. Java platform to develop java based application

12. **Style-sheet transformation (XSLT)**
    a. To transform XML documents into HTML web page, thus adding capability to modify the page contents by simple change in XML document. Please refer to "Dynamic UI Generation" section in Developer Guide for further details about this.

13. **Castor 1.3.2 (http://castor.org/xml-framework.html)**
    a. It is an XML data binding framework, to convert xml to java objects and vice versa. It uses xml configuration files to convert a java object into xml and vice versa.
       Following are few of such files used in CMS –

        - subsystem-mapping.xml – Contains mappings for ncra-subsystemconfig.xml file to generate corresponding java objects
        - castor-mapping.xml – Contains command configuration for all the *_commands.xml files.

14. **JNA – Java Native Access (http://jna.java.net/)**
    a. Used to invoke native APIs from java, this is used in CMS to call various apis in tact calculation library

# 2 Maven

This section explains the process of building cms-web.war using maven

## 2.1 Pre-Requisites

1. Maven 3.0 should be installed or your machine, or you can download http://maven.apache.org/download.html
2. Set the environment variable up to <installed directory>/maven/bin

## 2.2 Explicit Jar Installation

The jar files which are cannot be downloaded by maven need to be added in maven repository explicitly. Following are the steps to install the jars in maven:

1. List of the jars which are not downloaded by maven are available at <source code directory>/src/main/lib directory.
2. Open the terminal and go to the <source code directory>
3. Execute the following commands
    a. mvn install:install-file -DgroupId=antlr -DartifactId=antlr-runtime -Dversion=3.2 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib**/antlr-runtime-3.2.jar
    b. mvn install:install-file -DgroupId=drools -DartifactId=drools-api -Dversion=5.2.0.M1 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib**/drools-api-5.2.0.M1.jar
    c. mvn install:install-file -DgroupId=drools -DartifactId=drools-compiler -Dversion=5.2.0.M1 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib/**drools-compiler-5.2.0.M1.jar
    d. mvn install:install-file -DgroupId=drools -DartifactId=drools-core -Dversion=5.2.0.M1 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib/**drools-core-5.2.0.M1.jar
    e. mvn install:install-file -DgroupId=eclipse -DartifactId=jdtcore -Dversion=3.4.2.v_883_R34x -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib/**jdtcore-3.4.2.v_883_R34x.jar
    f. mvn install:install-file -DgroupId=groovy -DartifactId=groovy -Dversion=1.5.4 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib/**groovy-1.5.4.jar
    g. mvn install:install-file -DgroupId=mvel2 -DartifactId=mvel2 -Dversion=2.1.RC1 -Dpackaging=jar -Dfile**=<source code directory>/src/main/lib**/mvel2-2.1.RC1.jar
    h. mvn install:install-file -DgroupId=physhun -DartifactId=physhun -Dversion=0.5.1 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib**/physhun-0.5.1.jar
    i. mvn install:install-file -DgroupId=castor -DartifactId=castor -Dversion=1.3.2 -Dpackaging=jar -Dfile=**<source code directory>/src/main/lib**/castor-1.3.2.jar

    **Note:** Please only use "castor-1.3.2.jar" given by PSL, available on FTP in following location "**castorLibrary**"

## 2.3 Build Process

### 2.3.1 Maven Online mode

In this mode maven downloads all the required jar mentioned into build file (pom.xml) and builds the project.

1. Open the terminal and go to the CMS Code Base.

   NCRA
   |cms-validator
   |cms-core
   |cms-batch-processor
   |cms-state-machine
   |cms-web
   |pom.xml
2. Execute the command “mvn clean install”
3. This command reads the pom.xml and downloads the jars required to compile the entire projects and generate the cms.war at NCRA/cms-web/target/cms-web.war
4. You can use this cms-web.war and place it into <installed tomcat directory>/webapps/ for deployment.
5. To build the project please refers the maven integration for any IDE.
   E.g. Execute the “mvn eclipse:eclipse” command to build the eclipse project

### 2.3.2 Maven Offline mode

In this mode maven used the local repository for all the required jar mention in build file (pom.xml) to build the project.

To make it offline mode, set the local repository path as below

1. Go to <installed director>/maven/conf directory
2. Open the “settings.xml” and search for the “<localRepository>” tag and put your path as below:

   <localRepository><installed directory>/maven/repository</localRepository>
3. Save this file.

Steps:

a. Open the terminal and go to the source code directory.

   NCRA
   |cms-validator
   |cms-core
   |cms-batch-processor
   |cms-state-machine
   |cms-web
   |pom.xml

b. Execute the command “mvn –o clean install”
c. This command execute the maven into offline mode, so it read the pom.xml and take the local repository jars to compile the entire projects and generate the cms.war at < Source code directory >/cms-web/target/cms-web.war
d. You can use this cms-web.war and put it into <installed tomcat directory>/webapps/ for deployment.
e. To build the project please refers the maven integration for any IDE.
   E.g. Execute the “mvn –o eclipse:eclipse” command to build the eclipse project