****

**Weblogic to JBoss Migration**

**Development Guide**



**Project Name:** 삼성카드 노후서버 교체 프로젝트

**Department:** 삼성카드

**Focus Area:** Weblogic To JBoss Migration

**Product/Process:** JBoss EAP 7.0.4, JBoss EWS 3.0.3



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Man-Woong Choi | JBoss Technical Assistance |
|  |  |
|  |  |

**Project Status Report Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 1.0 | 2017/03/02 | Man-Woong Choi | Document created |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

[1. 본 문서의 목적 5](#_Toc476671038)

[1.1 작업의 범위 6](#_Toc476671039)

[2. JBoss Migration 환경 구성 7](#_Toc476671040)

[2.1 기본적인 Server 구성 Architecture 7](#_Toc476671041)

[2.2 Migration 환경 구성 8](#_Toc476671042)

[2.3 JBoss Install 8](#_Toc476671043)

[2.4 JBoss 디렉터리 기능 정보 9](#_Toc476671044)

[3. Application Migration 주의사항 10](#_Toc476671045)

[3.1 사전 고려 사항 10](#_Toc476671046)

[3.2 J2ee 변경에 따른 고려 사항 11](#_Toc476671047)

[3.3 JNDI 호출 방법에 대한 lookup 코드 변경 11](#_Toc476671048)

[3.4 JDBC 호출 방법에 대한 코드 변경 12](#_Toc476671049)

[3.5 weblogic.xml to jboss-web.xml 코드 변경 14](#_Toc476671050)

[3.6 weblogic-application.xml to web.xml 코드 변경 15](#_Toc476671051)

[4. 주요 Application Migration 변경 사항 16](#_Toc476671052)

[4.1 WebLogic Annotations 16](#_Toc476671053)

[4.2 @WLServlet Annotation 16](#_Toc476671054)

[4.2.1 AS-IS 17](#_Toc476671055)

[4.2.2 TO-BE 18](#_Toc476671056)

[4.3 @WLFilter Annotation 18](#_Toc476671057)

[4.3.1 AS-IS 19](#_Toc476671058)

[4.3.2 TO-BE 19](#_Toc476671059)

[4.4 @WLInitParam Annotation 20](#_Toc476671060)

[4.4.1 AS-IS 20](#_Toc476671061)

[4.4.2 TO-BE 21](#_Toc476671062)

[4.5 EJB Deployment Descriptor 추가 21](#_Toc476671063)

[4.5.1 기본 설정 변경 22](#_Toc476671064)

[4.5.2 Transaction Timeout 설정이 있는 경우의 설정 변경 23](#_Toc476671065)

[4.6 EAR Application 설정 변경 25](#_Toc476671066)

[4.6.1 Application Archive (jar, war)형태에 따른 정의 25](#_Toc476671067)

[4.6.2 Extends JAR (java archive ) file의 Class Loader 정의 26](#_Toc476671068)

[4.6.3 App-INF/classes의 Class Loader 정의 27](#_Toc476671069)

[4.7 Weblogic 의존 코드 변경 27](#_Toc476671070)

[4.7.1 AS-IS 27](#_Toc476671071)

[4.7.2 TO-BE 28](#_Toc476671072)

[4.8 Resource File 로딩 문제 28](#_Toc476671073)

[4.9 JSP 인코딩 수정 29](#_Toc476671074)

[4.10 JSP import를 중복 사용 30](#_Toc476671075)

[4.11 64KB를 넘는 JSP 페이지 31](#_Toc476671076)

[4.12 spring-module-jndi-lookup 설정 변경 34](#_Toc476671077)

[4.12.1 AS-IS 35](#_Toc476671078)

[4.12.2 TO-BE 36](#_Toc476671079)

[4.13 spring-modules-validation 속성 변경 37](#_Toc476671080)

[4.14 spring-dependency 오류 38](#_Toc476671081)

[4.15 struts2 library dependency 오류 40](#_Toc476671082)

[4.16 apache commons-lang library 오류 41](#_Toc476671083)

[4.17 Tag library 오류 43](#_Toc476671084)

[4.18 JSP Tag library 오류 수정 45](#_Toc476671085)

[4.19 JSP 태그 라이브러리 : TLD에 속성 선언하지 않음 46](#_Toc476671086)

[4.20 일부 한글이 깨지는 문제 48](#_Toc476671087)

[4.21 Message Resource bundle Default Language 오류 48](#_Toc476671088)

[4.22 Base64Encoder 클래스 변경 49](#_Toc476671089)

[4.23 Application 내의 log4j 설정을 위한 JBoss 설정 변경 50](#_Toc476671090)

[5. APPENDICES 52](#_Toc476671091)

[5.1 Document Guidelines 52](#_Toc476671092)

[5.2 Project Migration Report Sections Omitted 52](#_Toc476671093)

# 본 문서의 목적

본 문서의 목적은 삼성카드 노후서버 교체 프로젝트에서 사용중인 웹로직 WAS (Web Application Server)서버를 Opensource 기반의 JBoss EAP 로 이관함에 있어 원활한 서비스 전환을 위해 제공되어지는 개발자를 위한 기본 가이드 문서입니다.

## 작업의 범위

Weblogic to JBoss 작업을 위한 개발 지원이 주 목적이며, 그에 따른 Source 단에서 기본적으로 확인하여야 하는 부분에 대한 내용을 그 작업 범위로 합니다.

* Migrartion 대상 업무 장비
  + - 선택적 복지 WEB/WAS 4개 장비
    - 인터넷 공통 WEB/WAS 4개 장비
    - 인트라넷 공통 WEB/WAS 4개 장비
* Migrartion 대상 업무 시스템
  + - 선택적 복지외 9개 업무 시스템 애플리케이션 전환



# JBoss Migration 환경 구성

## 기본적인 Server 구성 Architecture

JBoss EWS #2

JBoss EWS #1

JBoss EAP #1

JBoss EAP #2

Oracle Database

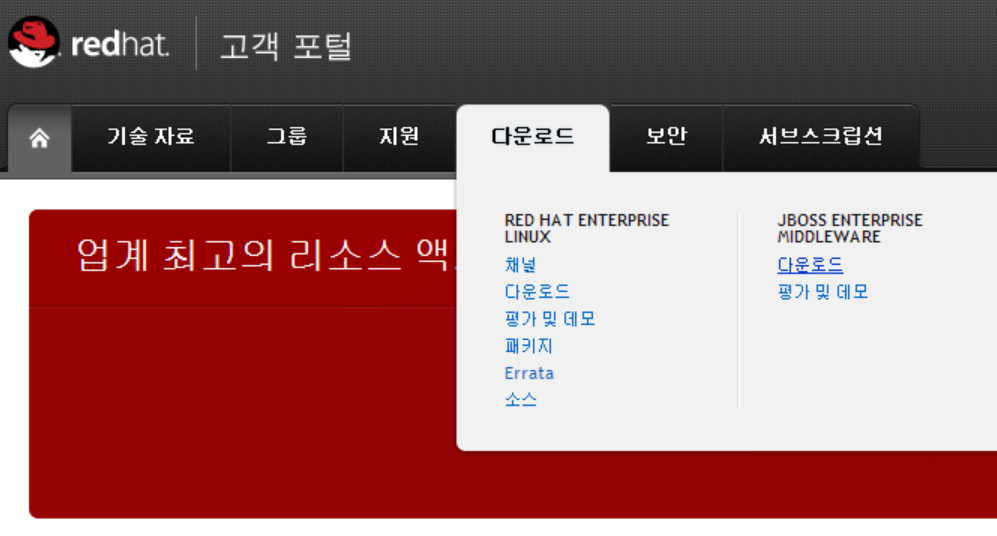
## Migration 환경 구성

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** |  |
| 운영체제 | Red Hat Enterprise Linux Server release 7.2 |  |
| 서버 IP |  |  |
| 기존 WAS 소프트웨어 | WebLogic Server 10.3.6 외 기타 버전 |  |
| 신규 WAS 소프트웨어 | JBoss Application Server 7.0.4 |  |
| Java Version | 1.8.0\_111,Sun Microsystems Inc. |  |

## JBoss Install

가장 최신 버전의 JBoss WAS는 EAP(Enterprise Application Platform) is 7.0 SP04 버전이며, Subscription 구매 고객은 [https://access.redhat.com/site/downloads/](http://network.jboss.com/) 사이트에서 다운로드가 가능하다.

JBoss 대부분의 소프트웨어들은 설치 프로그램 형태로 되어 있는 것이 아닌 압축을 풀면 해당 소프트웨어가 설치되는 간단한 방식을 취하고 있습니다.



## JBoss 디렉터리 기능 정보

각 인스턴스의 디렉토리 하위 정보는 다음과 같이 구성됩니다.

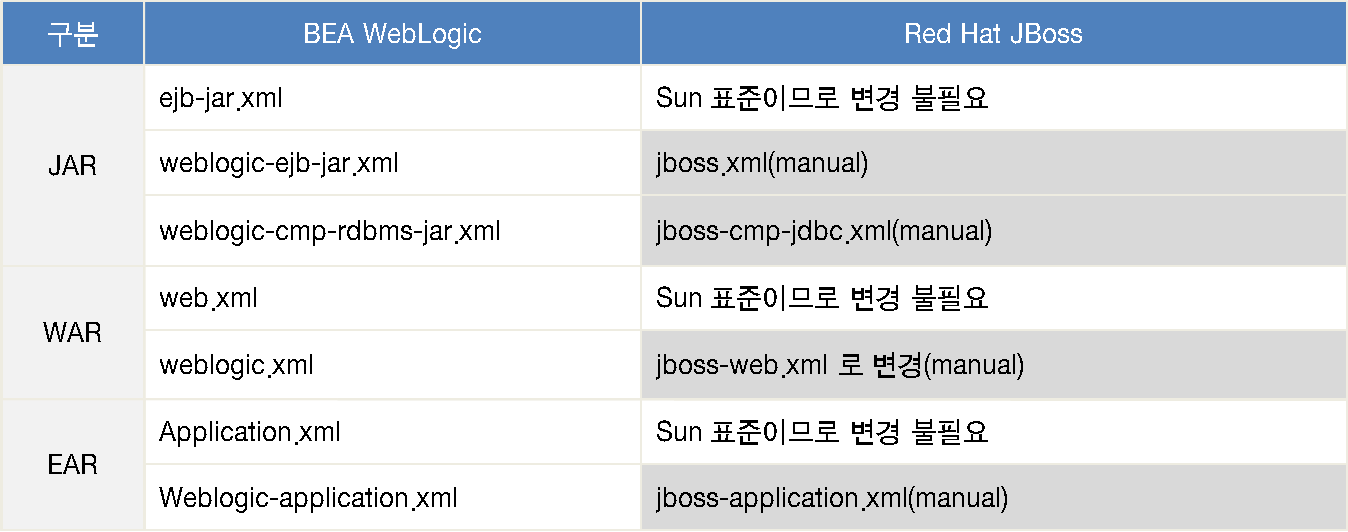
|  |  |
| --- | --- |
| **디렉터리** | **설명** |
| bin | 해당 JBoss 인스턴스 실행, 중지, 강제종료, 로그 보기, 스레드 덤프 생성등  인스턴스를 운영하기 위해 필요한 스크립트들이 보관되어 있다. |
| configuration | JBoss 메인구성 설정 파일 |
| data | JBoss 실행시 생성되는 파일들이 보관되는 디렉토리이다. |
| deployment | JBoss에 배포할 애플리케이션을 놓을 위치이다. |
| lib | 해당 인스턴스가 사용하게 될 Library 파일들이 보관되어 있다. |
| log | JBoss의 내부 로그 파일이 여기에 보관된다. ‘server.log’ 로그가 생성된다. |
| tmp | Web Application의 JSP 파일들의 java, class 파일들이 위치합니다. |

# Application Migration 주의사항

## 사전 고려 사항

|  |  |
| --- | --- |
| **구분** | **내용** |
| 일반 클래스 | * 공통적으로 사용하는 라이브러리 클래스들에 대한 분석을 통해   WAS에 의존적인 코드를 찾아 수정합니다. * 일반적으로 애플리케이션 수정 없이 재 컴파일만을 통해서 배치가 될 수 있습니다. |
| JSP/Servlet | * 개발된 JSP/Servlet 애플리케이션의 지원 스펙을 확인하여 최신 JSP/Servlet 스펙에 맞게 전환합니다. * 일반적으로 애플리케이션 수정 및 재 컴파일 없이 사용할 수 있습니다. |
| EJB | * EJB 스펙을 확인하고 WAS에 의존적인 DD.xml 파일을 JBoss용으로 전환. * EJB Bean 클래스의 비즈니스 로직을 가지고 있는 메소드에 대한 소스 수정은 없습니다. * J2EE EJB 스펙 변경에 따른 Home, Remote 인터페이스 클래스는 필요 없음. (EJB3.0으로 전환 시) * J2EE EJB 스펙 변경에 따른 메소드 별 Annotation 추가 (EJB 3.0으로 전환 시) |
| 클라이언트  프로그램 | * WAS Server의 Context를 lookup하기 위한 클라이언트 Property를 수정힙니다.   + InitialContext(), lookup() 사용 소스 수정 * 일반적인 경우 local context를 사용함으로 수정할 대상 소스가 거의 없습니다. |
| 라이브러리 | * 애플리케이션 구동에 필요한 3rd-party 라이브러리  (예:PL/SQL, Tuxedo 연동을 위한 Jolt, 아파치 log4j 등) |

## J2ee 변경에 따른 고려 사항



## JNDI 호출 방법에 대한 lookup 코드 변경

**AS-IS ( Weblogic )**

Properties props = new Properties();

props.put(Context.INITIAL\_CONTEXT\_FACTORY, “weblogic.jndi.WLInitialContextFactory”);

props.put(CONTEXT.PROVIDER\_URL, “t3://ip\_address:7001”);

InitialContext context = new InitialContext(props);

**TO-BE ( JBoss )**

<https://access.redhat.com/solutions/396853>

Properties jndiProperties = new Properties();

jndiProperties.put(Context.URL\_PKG\_PREFIXES, "org.jboss.ejb.client.naming");

jndiProperties.put("jboss.naming.client.ejb.context", true);

jndiProperties.put("java.naming.factory.initial", "org.jboss.naming.remote.client.InitialContextFactory");

jndiProperties.put("java.naming.provider.url", "remote://localhost:4447");

## JDBC 호출 방법에 대한 코드 변경

**AS-IS ( Weblogic )**

wasApps\WEB-INF\src\iit\common\util\DBUtil.java (브랜드관리)

|  |
| --- |
| import javax.naming.\*;  public static Connection getConnection() throws SQLException, NamingException {  InitialContext initialcontext = new InitialContext();  try {  Connection connection;  DataSource datasource = (DataSource)initialcontext.lookup("TxTdevDataSource");  connection = datasource.getConnection();  return connection;  } catch (NamingException ne) {  throw new EJBException(ne);  }  } |

**TO-BE ( JBoss )**

|  |
| --- |
| InitialContext ctxt = new InitialContext();  DataSource ds = (DataSource)ctxt.lookup(“java:/TxTdevDataSource”);  Con = ds.getConnection(); |

**브랜드관리 DATASOURCE JNDI 부분으로 수정하면됨**

**AS-IS ( Weblogic )**

src\scard\ejb\mall\hub\common\login\HEITMA100Bean.java(알뜰시장)

|  |
| --- |
| import com.nextel.weblogic.jdbc.JdbcConnectionManager;  public ResultSetList getUserInfo(String iden\_no) throws SQLException,Exception  {  JdbcConnectionManager jdbcconnManager = null;  Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;  ResultSetList rsList = null;  String queryStr = null;  try  {  jdbcconnManager = new JdbcConnectionManager("jdbc:weblogic:pool:MEMBER");  conn = jdbcconnManager.getConnection();  queryStr = " SELECT user\_id ,user\_password FROM ITcmmembertt WHERE iden\_no=?";  pstmt = conn.prepareStatement( queryStr );  pstmt.setString(1, iden\_no);  rs = pstmt.executeQuery();  rsList = new ResultSetList(rs);  rs.close();  pstmt.close();  } catch(SQLException sqlex) |

com.nextel.weblogic.jdbc.JdbcConnectionManager; 부분 소스 수정 여부?

**TO-BE ( JBoss )**

|  |
| --- |
| InitialContext ctxt = new InitialContext();  DataSource ds = (DataSource)ctxt.lookup(“java:ds/MEMBER”);  Con = ds.getConnection(); |

**Datasource jndi 설정하는 부분도 넣을 것**

**AS-IS ( Weblogic )**

mgrhtml\_new\src\main\java\com\ebestech\dbutil\DBConnDataSource.java (법인구매)

|  |
| --- |
| private Connection getWeblogicPool(Properties dbProps) throws Exception  {  Connection conn = null;  Driver driver = null;  try{  String lookup\_name = dbProps.getProperty("JNDI\_LOOK\_NAME");  driver = (Driver)Class.forName("weblogic.jdbc.pool.Driver").newInstance();  conn = driver.connect("jdbc:weblogic:pool:" + lookup\_name, null);  return conn;  }catch(SQLException qe){  System.err.println("DBConnDataSource Error(2)!!"+qe);  throw qe;  }  } |

**TO-BE ( JBoss )**

|  |
| --- |
| InitialContext ctxt = new InitialContext();  DataSource ds = (DataSource)ctxt.lookup(“java:ds/MEMBER”);  Conn = ds.getConnection(); |

## weblogic.xml to jboss-web.xml 코드 변경

**AS-IS ( Weblogic )**

WEB-INF/weblogic.xml

<context-root>MyAppContextRoot</context-root>

**TO-BE ( JBoss )**

WEB-INF/jboss-web.xml

<jboss-web>

    <context-root>MyAppContextRoot</context-root>

</jboss-web>

## weblogic-application.xml to web.xml 코드 변경

**AS-IS ( Weblogic )**

APP-INF/weblogic-application.xml

<application-param>

    <description>

        Web application default encoding

    </description>

    <param-name>

        webapp.encoding.default

    </param-name>

    <param-value>

        UTF8

    </param-value>

</application-param>

**TO-BE ( JBoss )**

WEB-INF/web.xml

<context-param>

    <description>

        Web application default encoding

    </description>

    <param-name>

        webapp.encoding.default

    </param-name>

    <param-value>

        UTF8

    </param-value>

</context-param>

# 주요 Application Migration 변경 사항

## WebLogic Annotations

기존의 Weblogic 에서만 사용되던 annotations 설정을 Java EE(표준)에 맞게 변경을 하여야 한다.

|  |  |  |
| --- | --- | --- |
| **WebLogic 사** | **Java EE 6 표준** | **Java EE 6 API Import** |
| @WLServlet | @WebServlet | javax.servlet.annotation.WebServlet |
| @WLFilter | @WebFilter | javax.servlet.annotation.WebFilter |
| @WLInitParam | @WebInitParam | javax.servlet.annotation.WebParam |

## @WLServlet Annotation

기존의 Weblogic 에서만 사용되던 WL\_Servlet annotations 설정을 사용하였다면, 그 내용에서 사용 되어 졌던 WL\_Servlet 내부 Attribute 항목을 Java EE(표준)에 맞게 변경을 하여야 한다.

|  |  |  |
| --- | --- | --- |
| **WebLogic Attribute Name** | **Java EE 6 표준** | **Description / Comments** |
| String displayName | String displayName | Display name for the Servlet |
| String description | String description | Servlet description |
| String icon | String smallicon -or-String largeicon | The icon location |
| String name | String name | The Servlet name |
| WLInitParam[] initParams | WebInitParam[] initParams | Servlet initialization parameters |
| int loadOnStartup | int loadOnStartup | Whether the Servlet should load on server start |
| String runAs | No equivalent attribute | The run-as User for the Servlet.  There are 3 options for replacement detailed below. |
| String[] mapping | String[] urlPatterns | The Servlet URL pattern |

### AS-IS

|  |
| --- |
| import weblogic.servlet.annotation.WLServlet;    *@WLServlet (*      name = "catalog",      runAs = "SuperEditor"      initParams = {  *@WLInitParam* (name="catalog", value="spring"),  *@WLInitParam* (name="language", value="English")       },       mapping = {"/catalog/\*"}  )  public class MyCatalogServlet extends HttpServlet { . . . } |

### TO-BE

|  |
| --- |
| import javax.annotation.security.RunAs;  import javax.servlet.annotation.WebServlet;    **@WebServlet** (      name = "catalog",      initParams = {  **@WebInitParam** (name="catalog", value="spring"),  **@WebInitParam** (name="language", value="English")      },      urlPatterns = "/catalog/\*"  )  **@RunAs**("SuperEditor")  public class MyCatalogServlet extends HttpServlet { . . . } |

## @WLFilter Annotation

기존의 Weblogic 에서만 사용되던 WLFilter annotations 설정을 사용하였다면, 그 내용에서 사용 되어 졌던 WLFilter 내부 Attribute 항목을 Java EE(표준)에 맞게 변경을 하여야 한다.

|  |  |  |
| --- | --- | --- |
| **WebLogic Attribute Name** | **Java EE 6 표준** | **Description / Comments** |
| String displayName | String displayName | Display name for the Servlet |
| String description | String description | Servlet description |
| String icon | String smallicon -or-String largeicon | The icon location |
| String name | String name | The Servlet name |
| WLInitParam[] initParams | WebInitParam[] initParams | Servlet initialization parameters |
| String[] mapping | String[] urlPatterns  String[] value | The Servlet URL pattern |
| No equivalent attribute | DispatcherTypes[] dispatcherTypes | The dispatcher types to which the filter applies |
| No equivalent attribute | String[] servletNames | The names of the servlets to which the filter applies |

### AS-IS

|  |
| --- |
| import weblogic.servlet.annotation.WLFilter;  import weblogic.servlet.annotation.WLInitParam;    *@WLFilter* (      name = "catalogFilter",      initParams = { *@WLInitParam*(name="catalog", value="winter") }      mapping = {"/catalog/\*"}  )  public class MyFilter implements Filter { . . . } |

### TO-BE

|  |
| --- |
| import javax.servlet.annotation.WebFilter;  import javax.servlet.annotation.WebInitParam;    **@WebFilter** (      filterName = "catalogFilter",      initParams = { **@WebInitParam**(name="catalog", value="winter") }      urlPatterns = {"/catalog/\*"}  )  public class MyFilter implements Filter { . . . } |

## @WLInitParam Annotation

기존의 Weblogic 에서만 사용되던 WLInitParam annotations 설정을 사용하였다면, 그 내용에서 사용 되어 졌던 WLInitParam 내부 Attribute 항목을 Java EE(표준)에 맞게 변경을 하여야 한다.

|  |  |  |
| --- | --- | --- |
| **WebLogic Attribute Name** | **Java EE 6 표준** | **Description / Comments** |
| String name | String name | Name of the initialization parameter |
| String value | String value | Value of the initialization parameter |
| No equivalent attribute | String description | Optional description of the initialization parameter |

### AS-IS

|  |
| --- |
| import weblogic.servlet.annotation.WLFilter;  import weblogic.servlet.annotation.WLInitParam;    *@WLFilter* (      name = "catalogFilter",      initParams = { *@WLInitParam*(name="catalog", value="winter") }      mapping = {"/catalog/\*"}  )  public class MyFilter implements Filter { . . . } |

### TO-BE

|  |
| --- |
| import javax.servlet.annotation.WebFilter;  import javax.servlet.annotation.WebInitParam;    **@WebFilter** (      filterName = "catalogFilter",      initParams = { **@WebInitParam**(name="catalog", value="winter") }      urlPatterns = {"/catalog/\*"}  )  public class MyFilter implements Filter { . . . } |

## EJB Deployment Descriptor 추가

EJB 2.x 버전에서는 EJB Deployment Descriptor가 필요하다.

ejb-jar.xml 파일은 표준 EJB Deployment Descriptor 파일이며, 각 WAS 마다 별도의 Descriptor를 설정하여야 한다.

WebLogic의 경우엔 weblogic-ejb-jar.xml 파일에 바인딩될 JNDI이름등을 지정하게 된다.

JBoss에서는 jboss.xml 파일을 사용한다.

<https://docs.jboss.org/ejb3/docs/reference/build/reference/en/html/jboss_deployment_descriptor.html>

### 기본 설정 변경

#### AS-IS

* 변환할 weblogic-ejb-jar.xml 파일

META-INF/weblogic-ejb-jar.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <weblogic-ejb-jar>  <description><![CDATA[Generated by XDoclet]]></description>    <weblogic-enterprise-bean>  <ejb-name>BisMainEjb</ejb-name>  <stateless-session-descriptor>  </stateless-session-descriptor>  <reference-descriptor>  </reference-descriptor>  <enable-call-by-reference>True</enable-call-by-reference>  <jndi-name>BmsBisMainFacadeHome</jndi-name>  <local-jndi-name>BmsBisMainFacadeLocalHome</local-jndi-name>  </weblogic-enterprise-bean>  </weblogic-ejb-jar> |

#### TO-BE

* 변환된 jboss.xml 파일

META-INF/jboss-ejb3.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <jboss>  <enterprise-beans>  <session>  <ejb-name>BisMainEjb</ejb-name>  <jndi-name>BmsBisMainFacadeHome</jndi-name>  <local-jndi-name>BmsBisMainFacadeLocalHome</local-jndi-name>  <call-by-value>false</call-by-value>  </session>  </enterprise-beans>  </jboss> |

### Transaction Timeout 설정이 있는 경우의 설정 변경

#### AS-IS

* 변환할 weblogic-ejb-jar.xml 파일

META-INF/weblogic-ejb-jar.xml

|  |
| --- |
| <?xml version="1.0"?>  <weblogic-ejb-jar>  <weblogic-enterprise-bean>  <ejb-name>BmsStaFacadeEJB</ejb-name>  <stateless-session-descriptor>  </stateless-session-descriptor>  *<transaction-descriptor>*  *<trans-timeout-seconds>180</trans-timeout-seconds>*  *</transaction-descriptor>*  <reference-descriptor>  </reference-descriptor>  <enable-call-by-reference>True</enable-call-by-reference>  <jndi-name>BmsStaFacadeHome</jndi-name>  <local-jndi-name>BmsStaFacadeLocalHome</local-jndi-name>  </weblogic-enterprise-bean>  </weblogic-ejb-jar> |

#### TO-BE

* 변환된 jboss.xml 파일

META-INF/jboss-ejb3.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <jboss>  <enterprise-beans>  <session>  <ejb-name>BmsStaFacadeEJB</ejb-name>  <jndi-name>BmsStaFacadeHome</jndi-name>  <local-jndi-name>BmsStaFacadeLocalHome</local-jndi-name>  <call-by-value>false</call-by-value>  <clustered>False</clustered>  <method-attributes>  <method>  <method-name>\*</method-name>  **<transaction-timeout>180</transaction-timeout>**  </method>  </method-attributes>  </session>  </enterprise-beans>  </jboss> |

## EAR Application 설정 변경

JBoss에서는 EAR ( Enterprise Application Archive) 형태의 애플리케이션의 확장자는 .ear

WAR (Web Application Archive ) 형태의 애플리케이션의 확장자는 .war 이어야 한다.

따라서, EAR 내의 bmsWeb WebApp의 확장자를 bmsWeb.war 로 변경한 후 이에 대한 application.xml 파일의 설정을 변경하여야 한다.

### Application Archive (jar, war)형태에 따른 정의

* 변환할 application.xml 파일

META-INF/application.xml

|  |
| --- |
| <!DOCTYPE application PUBLIC '-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN' 'http://java.sun.com/dtd/application\_1\_3.dtd'>  <application>  <display-name>BmsEar</display-name>  <description>Bms Ear</description>  **<module>**  **<web>**  **<web-uri>bmsWeb.war</web-uri>**  **<context-root>/bms</context-root>**  **</web>**  **</module>**  <module>  <ejb>drmEjb.jar</ejb>  </module>  <module>  <ejb>bisEjb.jar</ejb>  </module>  </application> |

### Extends JAR (java archive ) file의 Class Loader 정의

EAR내에서 APP-INF 하위 디렉토리의 파일을 클래스로 읽어들이는 것은 표준 스펙이 아닌 WebLogic에만 한정된 기능이다.

APP-INF/lib 디렉토리의 jar 파일을 로딩하기 위하여 아래 jboss-app.xml 파일을 추가하였다.

* 변환할 jboss-app.xml 파일

META-INF/jboss-app.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <jboss-app>  **<library-directory>APP-INF/lib</library-directory>**  </jboss-app> |

### App-INF/classes의 Class Loader 정의

APP-INF/classes 아래의 class 파일을 로딩하기 위하여 MANIFEST.MF 파일을 추가하였다.

* 변환할 MANIFEST.MF 파일

META-INF/ MANIFEST.MF

|  |
| --- |
| Manifest-Version: 1.0  **Class-Path: bms.ear/APP-INF/classes** |

## Weblogic 의존 코드 변경

웹로직에 의존적인 코드를 사용할 경우 표준 Application 동작에 문제가 있을 수 있으므로 검색을 통하여 해당 의존성을 없애도록 한다.

* weblogic.server.Server 사용 코드

### AS-IS

|  |
| --- |
| import weblogic.server.Server;  if(wasport.equals("")) wasport = Integer.toString(Server.getConfig().getListenPort()); |

### TO-BE

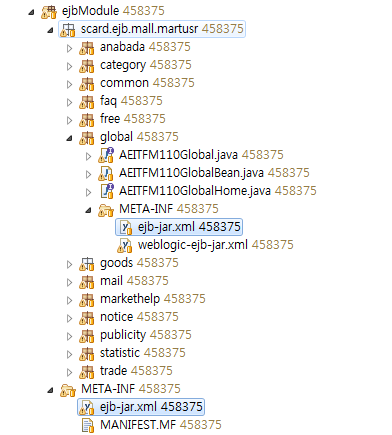
|  |
| --- |
| Import javax.servlet.http.HttpServletRequest;  HttpServletRequest **req** = (HttpServletRequest) request;  if(wasport.equals("")) wasport = Integer.toString(**req**.getServerPort()); |

## EJB 변경 필요 내용 ( ex: 임직원 알뜰시장 )

현재 Weblogic 에서 사용되던 EJB 를 변경하기 위해서는 다음의 3가지 항목등을 변경 수정하여야 합니다.

1. ejb-jar.xml 의 통합
2. 기존 JNDI 호출 이름 확인
3. JNDI 호출 방식 변경

### ejb-jar.xml 파일 내용의 통합



기존 ejb-jar.xml 의 분리된 deployments descriptor 항목을 하나의 ejb-jar.xml 항목으로 수정하여야 함.

#### AS-IS



#### TO-BE

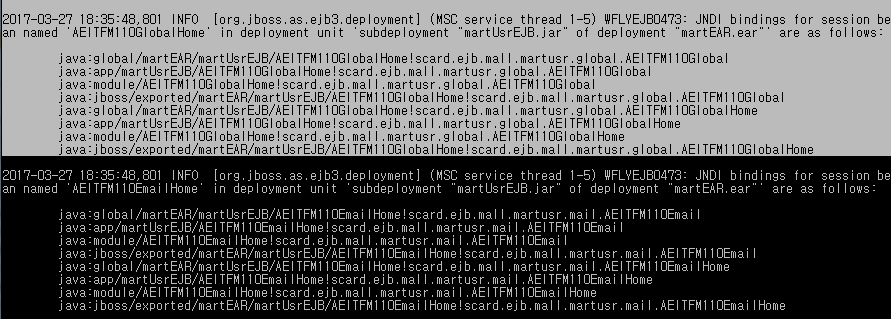


### 기존 JNDI 호출 Name 확인

#### 기존 ejb-jar.xml 확인

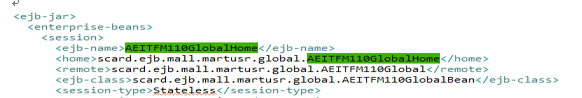


#### 신규 등록 JNDI 확인



### JNDI 호출 방식 변경

#### AS-IS



|  |
| --- |
| <ejb-name>AEITFM110GlobalHome</ejb-name>  Object BeanHome = ctx.lookup("AEITFM210CategoryHome");  AEITFM210Category beanRemote = (AEITFM210Category)home.create(); |

#### TO-BE

|  |
| --- |
| ctx = **new** InitialContext();  AEITFM210CategoryHome home =  (AEITFM210CategoryHome)ctx.lookup("java:app/martUsrEJB/AEITFM210CategoryHome! scard.ejb.mall.martusr.category.AEITFM210CategoryHome");  AEITFM210Category beanRemote = (AEITFM210Category)home.create(); |

## Resource File 로딩 문제

보통 Resource번들이나 혹은 properties 파일들을 읽어들이기 위하여 InputStream 을 이용하여 일반적으로 파일을 로드하게 설정을 한 경우, 실제로 “/” 로 시작하는 경로에서 읽어 들이도록 설정어 있는 case 가 있을 수 있다, 그와 같은 경우에는 다음과 같이 설정 변경이 필요하며, 그와 마찬가지로 클래스 패스에서 파일을 읽어들이기 위해서는 아래와 같이 코딩 변경이 필요하다.

Solution :

|  |
| --- |
| ***AS-IS***  is = getClass().getResourceAsStream(propFile);  **TO-BE**  **is = getClass().getClassLoader().getResourceAsStream(propFile);** |

코딩 방식에 대한 참고 자료 :

<http://stackoverflow.com/questions/1464291/how-to-really-read-text-file-from-classpath-in-java>

* 설정 파일 경로 변경

|  |
| --- |
| ***AS-IS***  <init-param>  <param-name>quartz-jobs-file</param-name>  <param-value>**/config/bms-quartz-config.xml**</param-value>  </init-param>  **TO-BE**  <init-param>  <param-name>quartz-jobs-file</param-name>  **<param-value>config/bms-quartz-config.xml</param-value>**  </init-param> |

## JSP 인코딩 수정

JSP 단의 한글 인코딩 및 MessageSource 의 오류 발생 가능성이 있으며, 그 부분을 완화하기 위하여 web.xml 부분에 다음의 내용을 추가하도록 한다.

* 변환할 web.xml 파일

WEB-INF/web.xml

|  |
| --- |
| <jsp-config>  <taglib>  <taglib-uri>http://jakarta.apache.org/taglibs/string-1.1</taglib-uri>  <taglib-location>/WEB-INF/tlds/taglib.tld</taglib-location>  </taglib>  <jsp-property-group >  <url-pattern>\*.jsp</url-pattern>  <page-encoding>UTF-8</page-encoding>  </jsp-property-group>  </jsp-config> |

## JSP import를 중복 사용

Solution : 에러 내용

|  |
| --- |
| 2016-11-07 10:24:06,544 ERROR [org.apache.catalina.core.ContainerBase] Servlet.service() for servlet jsp threw exception  org.apache.jasper.JasperException: Unable to compile class for JSP:  An error occurred at line: 7 in the generated java file  Syntax error on token "import", delete this token  Stacktrace:  at org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:92)  at org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:330)  at org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:415)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:312)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:290)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:277)  at org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:566)  at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:311)  at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:320)  at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:266) |

Solution :

<%@ page import=” 내에서 import를 중복 선언하여 문제 발생 page import 구문을 사용하므로 이후 의 import 명령은 삭제 조치합니다.

|  |
| --- |
| ***AS-IS***  <%@ page import="import bigbang.mvc.CacheManager" %>  **TO-BE**  **<%@ page import="bigbang.mvc.CacheManager" %>** |

## 64KB를 넘는 JSP 페이지

|  |
| --- |
| 2011-11-04 16:35:38,174 INFO [STDOUT] 11/04 16:35:38, DEBUG (Compiler ) : Generated /home/jjeon/jboss/eap/domains.43/s1/work/jboss.web/localhost/bms//org/apache/jsp/bms/dct/submit/BmsAddReportC\_jsp.java total=90 generate=16 validate=73  2011-11-04 16:35:38,405 ERROR [org.apache.catalina.core.ContainerBase] Servlet.service() for servlet jsp threw exception  org.apache.jasper.JasperException: Unable to compile class for JSP:  An error occurred at line: 178 in the generated java file  The code of method \_jspService(HttpServletRequest, HttpServletResponse) is exceeding the 65535 bytes limit  Stacktrace:  at org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:92)  at org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:330)  at org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:415)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:312)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:290)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:277)  org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)  at org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:687)  org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:230)  at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:173)  at org.jboss.web.tomcat.security.SecurityAssociationValve.invoke(SecurityAssociationValve.java:182) at  org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)  at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447)  at java.lang.Thread.run(Thread.java:662)  2011-11-04 16:35:38,406 INFO [STDOUT] 11/04 16:35:38, ERROR (BmsController ) : ServletException : Unable to compile class for JSP:  An error occurred at line: 178 in the generated java file  ***The code of method \_jspService(HttpServletRequest, HttpServletResponse) is exceeding the 65535 bytes limit***  Stacktrace:  2011-11-04 16:35:38,406 INFO [STDOUT] 11/04 16:35:38, ERROR (BmsController ) : org.apache.jasper.JasperException: Unable to compile class for JSP:  An error occurred at line: 178 in the generated java file  ***The code of method \_jspService(HttpServletRequest, HttpServletResponse) is exceeding the 65535 bytes limit***  Stacktrace:  at org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:92)  at org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:330)  at org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:415)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:290)  at org.apache.jasper.compiler.Compiler.compile(Compiler.java:277)  … 중략…  .web.tomcat.service.jca.CachedConnectionValve.invoke(CachedConnectionValve.java:157)  at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)  at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:241)  at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:844)  at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)  at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447)  at java.lang.Thread.run(Thread.java:662) |

Solution :

“The code of method \_jspService(HttpServletRequest, HttpServletResponse) is exceeding the 65535 bytes limit”

이 문제는 사이즈가 매우 큰 JSP 파일에서 생성되는 Java 메소드의 크기가 64KB을 넘어서 발생하는 문제로, JVM에서 64KB 가 넘는 메소드를 사용할 수 없어 발생한다. 아래와 같이 많은 이슈들이 제기되었지만 JVM에서 해결되어야 하는 문제이다. JVM 스펙의 제한점 때문이다.

<https://issues.apache.org/bugzilla/show_bug.cgi?id=39309>

<https://issues.apache.org/bugzilla/show_bug.cgi?id=39089>

<http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6294277>

|  |
| --- |
| 4.10 Limitations of the Java Virtual Machine and class File Format  \* The following limitations in the Java Virtual Machine are imposed by this version of the Java Virtual Machine specification:  \* The per-class constant pool is limited to 65535 entries by the 16-bit constant\_pool\_count field of the ClassFile structure (§4.1). This acts as an internal limit on the total complexity of a single class.  \* The amount of code per method is limited to 65535 bytes by the sizes of the indices in the exception\_table of the Code attribute (§4.7.4), in the LineNumberTable attribute (§4.7.6), and in the LocalVariableTable attribute (§4.7.7). |

해결할 수 있는 방법은 JSP 파일의 크기를 줄이는 방법으로 다음과 같이 <jsp:include>를 사용하여야 한다.

|  |
| --- |
| <jsp:include page="URI" flush="true"/> |

@include는 JSP파일을 하나로 합쳐 하나의 Java 파일을 생성하여 컴파일하지만, <jsp:include>의 경우에는 각각의 JSP 파일을 따로 컴파일하고 출력 결과만 합쳐서 출력하게 된다.

## spring-module-jndi-lookup 설정 변경

기존 Weblogic JNDI을 통해 Lookup 하던 부분을 JBoss JNDI를 Lookup하도록 변경하여야 한다.

Solution : 에러 내용

|  |
| --- |
| 18:40:25,592 ERROR [org.springframework.web.context.ContextLoader] (ServerService Thread Pool -- 69) Context initialization failed: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jdbcTemplate' defined in ServletContext resource [/WEB-INF/config/spring/applicationContext-cosmos-core.xml]: Cannot resolve reference to bean 'dataSource' while setting bean property 'dataSource'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'dataSource' defined in ServletContext resource [/WEB-INF/config/spring/applicationContext-datasource.xml]: Invocation of init method failed; nested exception is javax.naming.NamingException: JBAS011843: Failed instantiate InitialContextFactory weblogic.jndi.WLInitialContextFactory from classloader ModuleClassLoader for Module "deployment.slp.war:main" from Service Module Loader [Root exception is java.lang.ClassNotFoundException: weblogic.jndi.WLInitialContextFactory from [Module "deployment.slp.war:main" from Service Module Loader]]  at  org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:913) [org.springframework.context-3.1.1.RELEASE.jar:3.1.1.RELEASE]  at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:464) [org.springframework.context-3.1.1.RELEASE.jar:3.1.1.RELEASE]  at org.springframework.web.context.ContextLoader.configureAndRefreshWebApplicationContext(ContextLoader.java:385) [org.springframework.web-3.1.1.RELEASE.jar:3.1.1.RELEASE]  at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:284) [org.springframework.web-3.1.1.RELEASE.jar:3.1.1.RELEASE]  at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:111) [org.springframework.web-3.1.1.RELEASE.jar:3.1.1.RELEASE]  @ |

### AS-IS

WEB-INF/config/spring/applicationContext-datasource.xml

|  |
| --- |
| 중략   1. **기존 Weblogic jndi Lookup 부분 주석 처리**   <!-- Weblogic JNDI Configuration -->  <!--  <bean id="dataSource" class="org.springframework.jndi.JndiObjectFactoryBean">  <property name="jndiName" value="${jndi.name}"/>  <property name="jndiTemplate">  -->  <!-- dataSourceTemplate -->  <!--  <ref local="dataSourceTemplate"/>  </property>  </bean>  --> |

### TO-BE

WEB-INF/config/spring/applicationContext-datasource.xml

|  |
| --- |
| 중략   1. **jee NameSpace 등록**   <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:jee="http://www.springframework.org/schema/jee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.0.xsd">   1. **jee datasource 태그 내용 추가**   <jee:jndi-lookup id="dataSource" jndi-name="${jndi.name}"/>  중략 |

## spring-modules-validation 속성 변경

valang.tld 파일의 parse XML descriptor 의 dependency의 엄격성으로 인하여 발생을 하는 것으로 실제 에러 내용은 다음과 같습니다.

Solution : 에러 내용

|  |
| --- |
| at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:895) [rt.jar:1.6.0\_45]  at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:918) [rt.jar:1.6.0\_45]  at java.lang.Thread.run(Thread.java:662) [rt.jar:1.6.0\_45]  Caused by: org.jboss.as.server.deployment.DeploymentUnitProcessingException: JBAS018014: Failed to parse XML descriptor "/jboss/docs/deployments/aps/aps.war/WEB-INF/lib/spring-modules-validation-0.9.jar/META-INF/valang.tld" at [13,23]  at org.jboss.as.web.deployment.TldParsingDeploymentProcessor.parseTLD(TldParsingDeploymentProcessor.java:143)  at org.jboss.as.web.deployment.TldParsingDeploymentProcessor.processTlds(TldParsingDeploymentProcessor.java:122)  @ |

해결 내용 spring-modules-validation-0.9.jar 내의 valang.tld 내부의 <body-content>속성 변경 처리.

기존의 <body-content>None</body-content>부분을

<body-content>empty</body-content> 형식으로 변경후 jar파일 재배포 필요.

* 파일내 속성 설정 변경

spring-modules-validation-0.9.jar 내의 valang.tld

|  |
| --- |
| org.springmodules.validation.valang.javascript.taglib.ValangCodebaseTag  ***AS-IS***  <body-content>None</body-content>  **TO-BE**  <body-content>**empty**</body-content> |

Jar 파일 재 배포

## spring-dependency 오류

spring 프레임웍 내의 Version 오류로 인한 dependency 제약 사항 발생

Solution : 에러 내용

|  |
| --- |
| 16:41:57,488 ERROR [org.apache.catalina.core.ContainerBase.[jboss.web].[default-host].[/]] (ServerService Thread Pool -- 77) JBWEB000287: Exception sending context initialized event to listener instance of class org.springframework.web.context.ContextLoaderListener: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jdbcTemplate' defined in ServletContext resource [/WEB-INF/config/spring/applicationContext-cosmos-core.xml]: Cannot resolve reference to bean 'dataSource' while setting bean property 'dataSource'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'dataSource': Post-processing of the FactoryBean's object failed; nested exception is java.lang.IllegalArgumentException: warning no match for this type name: cosmos.comm.svc.CommSVC [Xlint:invalidAbsoluteTypeName]  @ |

해결 :

1. WEB-INF/lib 폴더의 spring 모듈의 Version 부분이 통일되어 있지 않아 발생하는 case 로 많이 발생하며, spring framework library version 통일하면 정상화 된다.

|  |
| --- |
| -rw-r--r-- 1 weblogic weblogic 378476 Apr 22 14:15 org.springframework.orm-3.1.1.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 231922 Apr 22 14:15 org.springframework.transaction-3.0.5.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 246719 Apr 22 14:15 org.springframework.transaction-3.1.1.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 544291 Apr 22 14:15 org.springframework.web-3.1.1.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 573701 Apr 22 14:15 org.springframework.web.servlet-3.1.1.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 31879 Apr 22 14:15 org.springframework.web.struts-3.0.0.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 482529 Apr 22 16:41 spring-modules-validation-0.9.jar  -rw-r--r-- 1 weblogic weblogic 359154 Apr 22 14:15 spring-web-3.0.0.RELEASE.jar  -rw-r--r-- 1 weblogic weblogic 386749 Apr 22 14:15 spring-webmvc-3.0.0.RELEASE.jar  를 모두  Spring Framework Version 을 모두 3.0.0.RELEASE.jar 에서 3.1.1.RELEASE.jar  로 통일 조치 |

1. WEB-INF/jboss-deployment-structure.xml 파일 생성으로 dependency 우회 조치

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <jboss-deployment-structure>  <deployment>  <dependencies>  <module name="org.jboss.ironjacamar.jdbcadapters" slot="main"></module>  </dependencies>  </deployment>  </jboss-deployment-structure> |

## struts2 library dependency 오류

stuts2 프레임웍 내의 Version 오류로 인한 dependency 제약 사항 발생

Solution : 에러 내용

|  |
| --- |
| **14:11:01,607 WARN [com.opensymphony.xwork2.util.FileManager] (ServerService Thread Pool -- 75) Could not create JarEntryRevision for [vfs:/jboss/docs/deployments/blg/blg.war/WEB-INF/lib/struts2-core-2.2.3.1]!: java.util.zip.ZipException: error in opening zip file**  at java.util.zip.ZipFile.open(Native Method) [rt.jar:1.6.0\_45]  at java.util.zip.ZipFile.<init>(ZipFile.java:127) [rt.jar:1.6.0\_45]  at java.util.jar.JarFile.<init>(JarFile.java:136) [rt.jar:1.6.0\_45]  at java.util.jar.JarFile.<init>(JarFile.java:100) [rt.jar:1.6.0\_45]  ….  중략  ….  at java.lang.Thread.run(Thread.java:662) [rt.jar:1.6.0\_45]  at org.jboss.threads.JBossThread.run(JBossThread.java:122)@ |

해결 :

WEB-INF/lib 폴더의 Struts모듈의 낮은 버전의 사용으로 인하여 발생하는 것으로 JBoss jee spec 에서 필요로 하는 다음의 library로 version 업그레이드 하는 것을 권고합니다.

|  |
| --- |
| ***변경전 라이브러리 version***  struts2-core-2.2.3.1.jar  struts2-spring-plugin-2.2.3.1.jar  struts2-tiles-plugin-2.2.3.1.jar  xwork-core-2.2.3.1.jar  **변경 후 라이브러리 version**  **struts2-core-2.3.4.jar**  **struts2-spring-plugin-2.3.4.jar**  **struts2-tiles-plugin-2.3.4.jar**  **xwork-core-2.3.4.jar** |

## apache commons-lang library 오류

서비스 프레임웍 내의 struts2 에서 사용하는 dependency library 모듈인 commons-lang library 에 StringUtils 이 클래스가 없어서 발생할 수 있습니다.

기존의 common-lang 모듈만 적용되어 있다면, commons-lang3-3.1.jar 모듈을 추가하여야 합니다.

Solution : 에러 내용

|  |
| --- |
| 14:30:31,974 ERROR **[org.apache.catalina.core.ContainerBase.[jboss.web].[default-host].[/]] (ServerService Thread Pool -- 83) JBWEB000284: Exception starting filter struts2: java.lang.NoClassDefFoundError: org/apache/commons/lang3/StringUtils**  at com.opensymphony.xwork2.config.providers.XmlConfigurationProvider.register(XmlConfigurationProvider.java:209) [xwork-core-2.3.4.jar:2.3.4]  at org.apache.struts2.config.StrutsXmlConfigurationProvider.register(StrutsXmlConfigurationProvider.java:102) [struts2-core-2.3.4.jar:2.3.4]  at com.opensymphony.xwork2.config.impl.DefaultConfiguration.reloadContainer(DefaultConfiguration.java:210) [xwork-core-2.3.4.jar:2.3.4]  at com.opensymphony.xwork2.config.ConfigurationManager.getConfiguration(ConfigurationManager.java:66) [xwork-core-2.3.4.jar:2.3.4].  …중략…  org.jboss.modules.ConcurrentClassLoader.performLoadClassUnchecked(ConcurrentClassLoader.java:459) [jboss-modules.jar:1.3.3.Final-redhat-1]  at org.jboss.modules.ConcurrentClassLoader.performLoadClassChecked(ConcurrentClassLoader.java:447) [jboss-modules.jar:1.3.3.Final-redhat-1]  at org.jboss.modules.ConcurrentClassLoader.performLoadClassChecked(ConcurrentClassLoader.java:414) [jboss-modules.jar:1.3.3.Final-redhat-1]  at org.jboss.modules.ConcurrentClassLoader.performLoadClass(ConcurrentClassLoader.java:389) [jboss-modules.jar:1.3.3.Final-redhat-1]  at org.jboss.modules.ConcurrentClassLoader.loadClass(ConcurrentClassLoader.java:134) [jboss-modules.jar:1.3.3.Final-redhat-1]  ... 20 more |

해결 :

WEB-INF/lib commons-lang3-3.1.jar library 추가하였습니다.

|  |
| --- |
| 추가 라이브러리  **commons-lang3-3.1.jar** |

## Tag library 오류

J2ee Spec의 높아짐에 따란 taglibs 모듈의 경로 인식이 엄격해 짐으로 인하여 오류로 발생할 수 가 있습니다.

Solution : 에러 내용

|  |
| --- |
| **14:44:10,895 ERROR [org.apache.catalina.core.ContainerBase.[jboss.web].[default-host].[/blg].[jsp]] (http-/121.252.241.77:8780-1) JBWEB000236: Servlet.service() for servlet jsp threw exception: org.apache.jasper.JasperException: JBWEB004113: The absolute uri: http://jakarta.apache.org/taglibs/string-1.1 cannot be resolved in either web.xml or the jar files deployed with this application**  **at org.apache.jasper.compiler.DefaultErrorHandler.jspError(DefaultErrorHandler.java:53) [jbossweb-7.3.1.Final-redhat-1.jar:7.3.1.Final-redhat-1]**  **at** org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:147) [jbossweb-7.3.1.Final-redhat-1.jar:7.3.1.Final-redhat-1]  at org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:88) [jbossweb-7.3.1.Final-redhat-1.jar:7.3.1.Final-redhat-1]  at org.apache.jasper.compiler.TagLibraryInfoImpl.generateTLDLocation(TagLibraryInfoImpl.java:207) [jbossweb-7.3.1.Final-redhat-1.jar:7.3.1.Final-redhat-1]  at org.apache.jasper.compiler.TagLibraryInfoImpl.<init>(TagLibraryInfoImpl.java:122) [jbossweb-7.3.1.Final-redhat-1.jar:7.3.1.Final-redhat-1  …중략…  at org.apache.tiles.servlet.context.ServletTilesRequestContext.dispatch(ServletTilesRequestContext.java:222) [tiles-servlet-2.1.4.jar:2.1.4]  at org.apache.tiles.renderer.impl.TemplateAttributeRenderer.write(TemplateAttributeRenderer.java:44) [tiles-core-2.1.4.jar:2.1.4]  at org.apache.tiles.renderer.impl.AbstractBaseAttributeRenderer.render(AbstractBaseAttributeRenderer.java:103) [tiles-core-2.1.4.jar:2.1.4]  at org.apache.tiles.impl.BasicTilesContainer.render(BasicTilesContainer.java:669) [tiles-core-2.1.4.jar:2.1.4] |

해결 :

/WEB-INF/classes/META-INF/taglib.tld 경로를 다음의 경로 /WEB-INF/tlds/taglib.tld 변경한 후 web.xml 부분에 경로 부분을 추가합니다.

/WEB-INF/web.xml 추가

|  |
| --- |
| 추가 설정 내용  **<jsp-config>**  **<taglib>**  **<taglib-uri>http://jakarta.apache.org/taglibs/string-1.1</taglib-uri>**  **<taglib-location>/WEB-INF/tlds/taglib.tld</taglib-location>**  **</taglib>**  **</jsp-config>** |

## JSP Tag library 오류 수정

JSP 태그 라이브러리가 표준에 맞지 않게 코딩되어 사용한 경우에 발생할 소지가 있다.

|  |
| --- |
| 011-11-04 15:58:59,510 ERROR [org.apache.catalina.core.ContainerBase] Servlet.service() for servlet jsp threw exception  **org.apache.jasper.JasperException: /bms/dct/list/BmsToDoL.jsp(792,4) Unable to find setter method for attribute: menunm**  at org.apache.jasper.compiler.DefaultErrorHandler.jspError(DefaultErrorHandler.java:40)  at org.apache.jasper.compiler.ErrorDispatcher.dispatch(ErrorDispatcher.java:407)  at org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:148)  at org.apache.jasper.compiler.Generator$GenerateVisitor.evaluateAttribute(Generator.java:2736)  at org.apache.jasper.compiler.Generator$GenerateVisitor.generateSetters(Generator.java:2965)  at org.apache.jasper.compiler.Generator$GenerateVisitor.generateCustomStart(Generator.java:2169)  at org.apache.jasper.compiler.Generator$GenerateVisitor.visit(Generator.java:1689)  at org.apache.jasper.compiler.Node$CustomTag.accept(Node.java:1525)  at org.apache.jasper.compiler.Node$Nodes.visit(Node.java:2357)  at org.apache.jasper.compiler.Node$Visitor.visitBody(Node.java:2407)  at org.apache.jasper.compiler.Generator$GenerateVisitor.visit(Generator.java:1705)  at org.apache.jasper.compiler.Node$CustomTag.accept(Node.java:1525)  at org.apache.jasper.compiler.Node$Nodes.visit(Node.java:2357)  at org.apache.jasper.compiler.Node$Visitor.visitBody(Node.java:2407)  at org.apache.jasper.compiler.Node$Visitor.visit(Node.java:2413)  at org.apache.jasper.compiler.Node$Root.accept(Node.java:495)  at org.apache.jasper.compiler.Node$Nodes.visit(Node.java:2357) |

커스텀 태그 라이브러리를 구현한 Java 코드에서는 setMenuNm() 와 같이 함수를 정의하였지만, TLD 파일의 설정은 menunm 으로 모두 소문자로 설정되어 있음.

***TLD 파일의 설정을 menuNm 과 같이 변경하고 이를 사용하는 모든 JSP 파일에서 menuNm 을 Attribute로 사용***

해야 하지만, 변경을 최소화하기 위해서, Java 코드에 다음과 같이 소문자 함수를 추가하는 것도 방법일 수 있다.

|  |
| --- |
| public void setMenuNm(String menuNm) {  this.menuNm = menuNm;  }  public void setMenunm(String menunm) {  this.menunm = menunm;  }  public void setMenuId(String menuId) {  this.menuId = menuId;  }  public void setMenuid(String menuId) {  this.menuId = menuId;  } |

## JSP 태그 라이브러리 : TLD에 속성 선언하지 않음

filename 속성이 TLD에 정의되어 있지 않아 문제가 발생함.

|  |
| --- |
| 2011-11-04 18:42:12,720 ERROR [org.apache.catalina.core.ContainerBase] Servlet.service() for servlet jsp threw exception  org.apache.jasper.JasperException: /bms/dct/env/BmsDctFormL.jsp(625,88) ***Attribute filename invalid for tag link according to TLD***  at org.apache.jasper.compiler.DefaultErrorHandler.jspError(DefaultErrorHandler.java:40)  at org.apache.jasper.compiler.ErrorDispatcher.dispatch(ErrorDispatcher.java:407)  at org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:236)  at org.apache.jasper.compiler.Validator$ValidateVisitor.checkXmlAttributes(Validator.java:1198)  at org.apache.jasper.compiler.Validator$ValidateVisitor.visit(Validator.java:819)  at org.apache.jasper.compiler.Node$CustomTag.accept(Node.java:1525)  at org.apache.jasper.compiler.Node$Nodes.visit(Node.java:2357)  at org.apache.jasper.compiler.Node$Visitor.visitBody(Node.java:2407)  at org.apache.jasper.compiler.Node$Visitor.visit(Node.java:2413)  at org.apache.jasper.compiler.Node$Root.accept(Node.java:495)  at org.apache.jasper.compiler.Node$Nodes.visit(Node.java:2357)  … 중략 …  org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:433)  at org.jboss.web.tomcat.security.JaccContextValve.invoke(JaccContextValve.java:84)  at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:128)  at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:104)  at org.jboss.web.tomcat.service.jca.CachedConnectionValve.invoke(CachedConnectionValve.java:157)  at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)  at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:241)  at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:844)  at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)  at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447) |

filename 속성이 TLD에 정의되어 있지 않아 문제가 발생함.

bmsWeb/WEB-INF/file-attach.tld

|  |
| --- |
| <attribute>  <name>filename</name>  <required>false</required>  <rtexprvalue>true</rtexprvalue>  </attribute>  … 중략 …  </tag> |

## 한글이 깨지는 문제

### 일부 한글 깨지는 문제

한글이 Get방식의 Parameter를 통해 UTF-8로 전달되어지는 경우 한글이 깨지는 현상이 발생할 수 있으며, 그에 따라 서버 기동시에 –D옵션의 System Property 값을 추가하여 환경 변수를 추가합니다.

**해결** :

|  |
| --- |
| export JAVA\_OPTS=”$JAVA\_OPTS -Dorg.apache.catalina.connector.URI\_ENCODING=UTF-8 -D org.apache.catalina.connector.USE\_BODY\_ENCODING\_FOR\_QUERY\_STRING=true |

### 화면 한글 깨져 보이는 현상

< 브랜드 관리 에서 >

AS-IS 소스에서 Encoding 을 하여 한글이 깨져 보입니다.

|  |
| --- |
| <td><%=**new** String(entity.req\_userid.getBytes("8859\_1"),"KSC5601")%></td>  <td><%=**new** String(entity.job\_userid.getBytes("8859\_1"),"KSC5601")%></td> |

Solution :

TO-BE 에서는 아래 처럼 encoding 부분을 없애거나 euc-kr 로 수정하면 됩니다.

|  |
| --- |
| <td><%=entity.req\_userid%></td>  <td><%=**new** String(entity.job\_userid.getBytes("euc-kr"),"euc-kr")%></td> |

## Message Resource bundle Default Language 오류

Solution : 에러 내용

|  |
| --- |
| 14:06:39,798 WARN [org.apache.commons.validator.ValidatorResources] (ajp-/121.252.241.77:8109-1) Form '/lpm/cmm/login/executeLogin.do' not found for locale 'en\_US' |

Solution : 에러 원인

Linux 사용자 Locale 설정이 en\_US.utf8 설정이 되어 있다면, 서버의 기본 locale 설정이 틀려 지게 되므로 서버에서 접속 Locale 이 틀려지게 되는 경우가 있다.

Solution : 에러 해결

유저 계정의 .bash\_profile 혹은 profile 부분에 locale 계정의 LANG 설정을 ko\_KR.utf8 로 변경한다.

|  |
| --- |
| 추가 설정 내용  **export LANG=ko\_KR.utf8** |

## Base64Encoder 클래스 변경

|  |
| --- |
| Access restriction: The type BASE64Encoder is not accessible due to restriction on required library /usr/java/jdk1.5.0\_22/jre/lib/rt.jar |

위의 에러 메시지는 sun.misc.BASE64Encoder 클래스 인데, 아래 URL과 같이 rt.jar 파일의 클래스를 사용해선 정상적으로 동작이 안된다.

<http://www.oracle.com/technetwork/java/faq-sun-packages-142232.html>

하여 아래와 같이 apache commons의 Base64 인코더를 사용하도록 변경하여야 한다.

|  |
| --- |
| import org.apache.commons.codec.binary.Base64;  import org.apache.commons.lang.Validate;  import org.xml.sax.SAXException;  //import sun.misc.BASE64Encoder;  … 생략 …  // resultXML base64 encondig  //BASE64Encoder enc = new BASE64Encoder();  //resultXml = enc.encodeBuffer(resultXml.getBytes());  resultXml = new String(Base64.encodeBase64(resultXml.getBytes())); |

## Application 내의 log4j 설정을 위한 JBoss 설정 변경

Application 내에서 log4j 에 대한 설정이 정상적으로 적용이 되지 않는 경우가 있을 수 있다.

그것을 회피하기 위해서는 JBoss를 위한 Deployment Structure Descriptor 파일인 jboss-deployment-structure.xml 파일을 아래 디렉토리에 추가하여야 한다.

webApplication 의 log4j 설정

/WEB-INF/jboss- deployment-structure.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <jboss-deployment-structure>  <deployment>  **<exclusions>**  **<module name="org.apache.log4j" />**  **</exclusions>**  <dependencies>  <module name="org.jboss.ironjacamar.jdbcadapters" slot="main"></module>  </dependencies>  </deployment>  </jboss-deployment-structure> |



# APPENDICES

## Document Guidelines

<https://docs.jboss.org/author/display/AS72/How+do+I+migrate+my+application+from+WebLogic+to+JBoss+EAP+6>

## Project Migration Report Sections Omitted

N/A

