Automation Deployment Guide

Version History:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>02/08/15</td>
<td>Maha Hussein Sallam</td>
<td>Initial version</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://www.linkedin.com/in/mahahusseinsallam">https://www.linkedin.com/in/mahahusseinsallam</a></td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

Purpose: ...................................................................................................................................................... 3

Scope: .......................................................................................................................................................... 4

Prerequisites: ............................................................................................................................................... 4

1. Overview on Tools required for Automation............................................................................................ 6

   1.1. Version Control System .................................................................................................................. 6

       1.1.1. Subversion Edge ....................................................................................................................... 6

           1.1.1.1. Restart Subversion admin GUI & create your Repository .............................................. 6

       1.1.2. TortoiseSVN: .......................................................................................................................... 7

           1.1.2.1. Connect SVN client with the Repository URL and commit the code............................. 7

   1.2. Jenkins ............................................................................................................................................ 7

       1.2.1. Enrich Jenkins Functionality by adding Jenkins Plugins......................................................... 7

       1.2.2. Restart Jenkins ....................................................................................................................... 7

   1.3. ApacheAnt: ..................................................................................................................................... 8

       1.3.1. Advanced Ant Tasks /Jar required for Automation : ............................................................. 8

       1.3.2. Overview on Ant's buildfiles ( build.xml ) ........................................................................... 8

           1.3.2.1. Target Examples ............................................................................................................... 8

2. Getting Started with Jenkins ................................................................................................................... 13

   2.1. Prerequisites ................................................................................................................................... 13

       2.1.1. Add the build.xml inside your Application ............................................................................ 13

       2.1.2. Modify property values in build.xml with the ENV & application details ......................... 13

       2.1.3. Check-in your code ................................................................................................................. 13

   2.2. Getting started with Jenkins .......................................................................................................... 14

       2.2.1. Login to Jenkins (http://as746:1800/) ................................................................................. 14

       2.2.2. Manage Jenkins .................................................................................................................... 14

           2.2.2.1. Add New Tap for any new Env ....................................................................................... 14

           2.2.2.2. Create a new user for Jenkins ......................................................................................... 15

           2.2.2.3. Create a new job for you application ............................................................................. 17

           2.2.2.4. Configure Jenkins for your Application job .................................................................. 18

           2.2.2.5. Execute the Automation activity for your Application .............................................. 20

           2.2.2.6. Monitor the Automation status through the Console ............................................... 20

3. References: ........................................................................................................................................... 22
**Purpose:**

The purpose of this guide is to automate the process of Building and Deploying EAR for TIBCO Business works application across TIBCO Administrator Environment.

By integrating SVN, Jenkins & Ant Script, the below activities will be done by just one click in Jenkins:

1. Use BuildEar utility to build the EAR from BW Application code.
2. Use Appmanage utility with export option to export XML from the above Generated EAR.
3. Update the exported XML values from property File that contains the values of the environment (ex: DEV).
4. Copy the generated EAR and updated XML to the remote machine that you will deploy on it.
5. Create any deployment prerequisites that will be used by the application like (Creating EMS Q, DB Table, etc).
6. Use Appmanage utility to Deploy the BW application in the environment.
**Scope:**
This guide will help you to setup, configure & troubleshoot the tools and scripts that will help you to run the Automation POC.

Automation POC will be done with the below components & Software version

**The key components of Automation are:**

- People (Developers)
- BW Application code (on Developer’s desktop with windows 7)
- SVN repository & Continuous Integration (CI)server (Separate Windows server)
- Deployment Environment (Unix sever with Solaris 10)

**Software version:**

<table>
<thead>
<tr>
<th>Serial NO</th>
<th>Software List existing</th>
<th>Where to install</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TIB_rv_8.3.2</td>
<td>CI windows server and also on the Deployment Unix Server</td>
</tr>
<tr>
<td>2</td>
<td>TIB_tra_5.7.3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TIB_ems_6.3.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TIB_bw_5.9.3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TIB_adadb_7.0.0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TIB_tibcoadmin_5.7.0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>JRE</td>
<td>CI windows server</td>
</tr>
<tr>
<td>8</td>
<td>CollabNetSubversionEdge-4.0.5_setup-x86_64</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Jenkins-1.553</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>jdk1.6.0_27</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>apache-ant-1.9.3-bin</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TortoiseSVN-1.8.5.25224-x64-svn-1.8.8</td>
<td>Developer’s desktop with windows 7</td>
</tr>
</tbody>
</table>
Prerequisites:

- A prior knowledge of SVN , Ant Script & TIBCO BW script deployment is assumed.
- The below software need to be downloaded & installed in the server that will act as SVN repository & continuous integration server:
  1- **TIBCO** (RV,TRA,BW,EMS, any other TIBCO Installation required for building Ear correctly)
  2- **JRE** then **Subversion Edge** (ex: CollabNetSubversionEdge-4.0.5_setup-x86_64)
  3- **Jenkins** (ex: jenkins-1.553) and -Enrich Jenkins Functionality by Adding some plugins (ex: discard-old-build)
  4- **JDK** (ex:jdk1.6.0_27)then **ANT** (ex:apache-ant-1.9.3-bin)
  5- Enrich Ant Functionality by Adding external JARs inside ANT lib folder (ex:”xmltask.JAR” and ant-contrib-0.6 JAR)
1. Overview on Tools required for Automation

1.1. Version Control System

1.1.1. Subversion Edge

Free web a Version Control Systems that simplifies administration and configuration of subversion server by providing the below capabilities

- **Repository Management:** Create, manage, browse, and local backup
- **User Management:** Manage roles, permissions and access rules
- **Administration:** Authentication, server management and health analytics

If you are interested to know more about tool you can visit to this URL:
http://www.collab.net/products/subversion

And Here Is the subversion Help http://help.collab.net/

1.1.1.1. Restart Subversion admin GUI & create your Repository

-CollabNet Subversion Edge server edition uses apache2 server for http connection and it provides a Web-based interface for configuring the subversion server and managing the repositories. the default port for accessing the GUI is 3343. so, by entering http://localhost:3343 in the address bar, you will be brought to the login page (admin/admin).

-If the page is down, This usually indicates the CollabNet Subversion Edge service has stopped, which can be verified by entering services.msc in the search box to go to the local service window, and make sure that you restart the below service.

-After making GUI up, create your repository and save your Repo URL, as we will connect to it later

-Make sure that the Repo URL is correct and no issue in the port, to avoid “Unexpected HTTP status 400 ‘Bad Request’ while commit “ ex: http://as746:18080/svn/Middleware_Repository
1.1.2. **TortoiseSVN:**

a free open-source Windows client for the Apache Subversion, manages files and directories over time, developed as an extension for the Windows Explorer.

Kindly visit the below link for more info about the SVN Features: http://tortoisesvn.net/docs/release/TortoiseSVN_en/tsvn-preface-features.html

1.1.2.1 Connect SVN client with the Repository URL and commit the code.

I am assuming that your already aware of this part

1.2. **Jenkins**

Jenkins is Continuous Integration server with GUI that monitors executions of repeated jobs, support building and testing virtually any project.

Kindly visit this link for more info http://jenkins-ci.org/content/about-jenkins-ci

1.2.1. **Enrich Jenkins Functionality by adding Jenkins Plugins**

To Enrich Jenkins Functionality you can download the below plugins from this site (https://updates.jenkins-ci.org/download/plugins/)

- **discard-old-build**: This plugin provides a post-build step where you can discard old build results in detailed configuration (ex. keep only builds with specified status).

- Sometimes when you download plugins you may get (.zip) files then just rename with (.hpi) and then extract all the plugins and move to `<jenkinsHome>/plugins/` directory , Afterwards you probably need to restart Jenkins.

1.2.2. **Restart Jenkins**

Open CMD in CI windows server and go to Jenkins directory then run the below commands

```
C:\ProgramFiles\Jenkins> jenkins.exe stop
C:\ProgramFiles\Jenkins> jenkins.exe start
C:\ProgramFiles\Jenkins> jenkins.exe restart
```
1.3. ApacheAnt:
Free software tool for automating software build processes.

Kindly visit this link for more info  [http://ant.apache.org/](http://ant.apache.org/)

1.3.1. Advanced Ant Tasks / Jar required for Automation:
1-Xmeltask required the usage of Xmeltask.jar to allow you for XML manipulation activities (ex: replace xml value)

- Kindly visit the below link for more info

2-<for> task required the usage contrib-0.6.jar to allow you to do looping activities

- Kindly visit the below link for more info

1.3.2. Overview on Ant's buildfiles (build.xml)
Apache Ant's buildfiles are written in XML. Each buildfile contains one project and at least one (default) target. A target can depend on other targets. Targets contain task elements, A task is a piece of code that can be executed.

1.3.2.1. Target Examples

1.3.2.1.1. Clean

```xml
<target name="Clean"

description="Delete EAR directory in the CI Windows server and create a new empty one">

<delete dir="${ear.dir}" failonerror="false"/>

<mkdir dir="${ear.dir}"/>

</target>
```
1.3.2.1.2. **Build.EAR**

- If you are using Design Time library in your code, so please make sure that you open designer in CI server just one time to fix any DTL broken reference. Otherwise, the build step will fail.

- You need to create a folder in CI server that contains DTL property file and put all used DTL inside this folder.

- You need to add the below property in your `build.xml` file:

```xml
<property name="DTL.Property.Path" value="E:\ProgramFiles\Jenkins\workspace\DTL\BuildEar_DTL.Properties.txt"/>

<target name="Build.EAR" depends="Clean" description="Validate and Build EAR for the BW project with DTL">

  <exec executable="${tibco.BuildEar.path}" failonerror="true">
    <env key="PATH" path="${tibco.tra.home}"/>
    <arg value="-x"/>
    <arg value="-v"/>
    <arg value="-p"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
    <arg value="-o"/>
  </exec>

</target>
```
1.3.2.1.3. Export.XML

```xml
<target name="Export.XML" depends="Build.EAR" description="Extracts the XML config file from the generated ear file with GV that defined inside the EAR ">
  <exec executable="C:\tibco\tra\5.7\bin\AppManage.exe" failonerror="true">
    <env key="PATH" path="C:\tibco\tra\5.7\bin"/>
    <arg value="-export"/>
    <arg value="-out"/>
    <arg value="${Exported.XML.Path}"/>
    <arg value="-ear"/>
    <arg value="${Remote.EAR.Path}"/>
  </exec>
</target>
```

1.3.2.1.4. UpdateXML

- Update Global variable values in the XML based on property File

```xml
<target name="UpdateXML" description="update the generated XML configuration values based on ENV Property file values " depends="init">
  <property file="${propertyFile.Name}" prefix="x"/>
  <local name="propertyList"/>

- loop on the values on the property file and replace the XML with these values

The below XMLTask & For Task will loop on the values on the property file and replace the XML with these values

```xml
<for list="${propertyList}" param="sequence">
  <sequential>
    <xmltask dest="${Updated.xml.Path}"
      fileset="${Updated.xml.Path}"
      replace
      path="${x.@{sequence}.xpath}"
      withText="${x.@{sequence}.value}"/>
  </sequential>
</for>
```
1.3.2.1.5. Duplicate the XML Binding Node in case of Load Balance deployment
This is optional only if load Balance deployment is required
We need to duplicate the Binding Tag

```xml
<target name="AddBindingNode_LB" depends="Export.XML,init" description="Add Binding Node">
    <xmltask source="${Updated.xml.Path}"
        <copy path="//:application/:services/:bw/:bindings/:binding" buffer="initBindingBuffer"/>
    </xmltask>
    <xmltask source="${Updated.xml.Path}" dest="${Updated.xml.Path}"
        <paste path="/:application/:services/:bw/:bindings/:binding" position="after" buffer="initBindingBuffer"/>
    </xmltask>
</target>
```

1.3.2.1.6. Duplicate NVpairs inside the Each Binding for Each GV marked on Service level
This is optional only if load Balance deployment is required
We need to duplicate the NVPaires Tag

```xml
<target name="AddNVPairsNode_LB" depends="Export.XML,init,AddBindingNode_LB" description="Add Binding Node">
    <xmltask source="${Updated.xml.Path}"
        <copy path="//:application/:services/:bw/:NVPairs[:NameValuePair/:name='ConnectionDetails/JMS_Connections/CLE/ClientId']" buffer="initNVPairsBuffer"/>
    </xmltask>
    <xmltask source="${Updated.xml.Path}" dest="${Updated.xml.Path}"
        <paste path="/:application/:services/:bw/:bindings/:binding/:shutdown" position="after" buffer="initNVPairsBuffer"/>
    </xmltask>
</target>
```

1.3.2.1.7. Copy Deployment Files from CI server to Deployment Unix Machine

```xml
<target name="CopyDeploymentFilestoRemoteMachine" description="Copy the generated XML and updated Ear to ENV UNIX Machine" depends="UpdateXML">
    <sshexec trust="true" host="${tibco.server.host}" username="${ssh.user}" password="${SSH.Password}" command="mkdir ${Server.Config.Folder.Path};" failonerror="false"/>
    <scp trust="true" file="${Remote.EAR.Path}" todir="${SSH.User}:${SSH.Password}@${tibco.server.host}:${SSH.Location}"/>
    <scp trust="true" file="${Updated.xml.Path}" todir="${SSH.User}:${SSH.Password}@${tibco.server.host}:${SSH.Location}"/>
</target>
```
1.3.2.1.8. Deploy

- You need to add the below properties first in your build.xml to make your deployment comes with description

```
<property environment="env"/>
<property name="appmanage.deployment.description" value="${envJOB_NAME}-Jenkins${envBUILD_NUMBER}-SVN${envSVN_REVISION}"/>
```

```
<target name="deployEar" depends="CreateDeployment Prerequisites" description="Deploy EAR & XML using Appmange and don't start application after deployment ">
  <sshexec trust="true" host="${tibco.server.host}" username="${SSH.User}" password="${SSH.Password}"
  command="cd ${Server.tibco.tra.home};
  ./AppManage -deploy -app ${tibco.admin.dir} -domain ${tibco.domain}
  -deployconfig ${tibco.app.xml.config} -ear ${tibco.app.earfile}
  -user ${tibco.admin.user} -pw ${tibco.admin.password} -desc ${appmanage.deployment.description} -nostart"
  failonerror="false"/>
</target>
```

1.3.2.1.9. Complete Sample for the Folder Setup & Configuration

![WinRAR archive]
2. Getting Started with Jenkins

2.1. Prerequisites

2.1.1. Add the build.xml inside your Application
Example on the Folder structure:

Code Folder
    |__________ Your Application Code Folder
    |__________ SupportResources
          |____Jenkins
                |____build.xml

2.1.2. Modify property values in build.xml with the ENV & application details

2.1.3. Check-in your code
2.2. Getting started with Jenkins

2.2.1. Login to Jenkins ([http://as746:1800/](http://as746:1800/))

2.2.2. Manage Jenkins

2.2.2.1. Add New Tap for any new Env

1-press (+) option

2-Write the Environment name then press next

3-if you need to take replica from jobs that currently exists in another Env,
Just pick the job names that you need.

<table>
<thead>
<tr>
<th>Name</th>
<th>PRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

Filter build queue  ☐
Filter build executors  ☐

**Job Filters**

- Status Filter: All selected jobs
- Recurse in subfolders: ☐
- Jobs: ☐
  - BBServiceStatus
  - SIMVoucherStatus
  - Test

☐ Use a regular expression to include jobs into the view

**Columns**

- Status
- Weather
- Name

[OK] [Apply]

---

2.2.2.2. *Create a new user for Jenkins*

1-Double click on “Manage Users”
2-then choose “create User”

3-update the user Authority permissions
4- you need to write the new user name and then press add

2.2.2.3. Create a new job for your application

1- Select the ENV Tab that you need to create your Job Under It (Ex: UAT)

2- Then select New Item from the left list

2- write your Application name (please mention the ENV Name at the end of App name: Ex: APPName_DEV)

If it is new Application, you can mark this option “Build a free-style software project”

Else if this application is currently exists you can mark “Copy existing Item”
2.2.2.4. Configure Jenkins for your Application job

1- Double click on your job name then click on “configure” option.

2- Under “Source code Management” section, add the SVN location for your code.
3- To Invoke the Ant script in build.xml in your job, you need to choose Invoke Ant under the “Build Section”

As per our SVN setup all Application URL will be like the below:

http://as746/svn/Middleware/Development/<ApplicationName>/Code

4-the default is to leave the below Target Empty, it will Invoke by default any build.XML file inside your Code Folder if you follow the folder structure mentioned before.

-Press advanced in the build File mention build.xml location relative to the above SVN URL

-All application should share the same relative location .\SupportResources\Jenkins\build.xml

- To Discard/truncate Failed Jobs, we can configure which status to discard from the application jobs, this will help to minimize storage used by jenkins
2.2.2.5. Execute the Automation activity for your Application

1- Select “build Now” to execute your Job

2- A new job will be created on the “Build History”

2.2.2.6. Monitor the Automation status through the Console

1- To monitor the execution of your job, click on the new created job in Build History

2- Then click on the “Consol Output” in the left list,

You will find the log of the execution for each target mentioned in the Build.xml
Finally You need to make sure that Ear has been built & deployed successfully
Jenkins

Console Output

�行由 user Name 提供

Build is underway. E:\Program Files\Jenkins\workspace\MMServiceStatus


New Build サーバー 110

Not change for http://a740.ae-tel2.01.etelco.com:8080/job/MMServiceStatus/90/MMServiceStatus since the previous build

Buildfile: E:\Program Files\Jenkins\workspace\MMServiceStatus\build.xml

Clean:

  deleting directory E:\Program Files\Jenkins\workspace\MMServiceStatus

configure:

  configuring...

configure Ant:

  configuring...

configuration erfolgt...

Finished configuration

configure application configuration...

Finished configuring application configuration

Finished successfully in 7 seconds

Last:

Reconfigurer

  connecting to domo:22

  home

  mounting agent /#application/services/hs/bindings/bindings/machine/texts

  mounting

  mounting domo:22

  A

  connecting to domo:22

  A

  connecting to domo:22

http://a740.ae-tel2.01.etelco.com:8080/job/MMServiceStatus/90/console

Local internet | Protected Mode: off
3. References:

