**SCM Deployment Instructions**

**Full Build/Delta Deployment instructions:**

**Build Server configurations/pre-requisite**

* 1. 3DEx 2020x env
  2. AdoptOpenJDL JDK 11.0.x, with x > 3, with OpenJ9 is a compatible platform.
  3. ANT 1.10.7 should be configured on the server.
  4. ‘JAVA\_HOME’ and ‘ANT\_HOME’ system level variables should be configured.

**Server Deployment**

1. **Script configuration for PLM Admin :** [deploy\_build.sh]
2. This file is not part of the source code. PLM deploy admin need to copy ‘deploy\_build.sh’ file to the build server. This is the entry point of the deployment process. PLM deploy admin just needs to execute ‘deploy\_build.sh to start the process.
3. PLM Deploy admin must update the following information in the script before starting the build process.
   * SERVER\_DEPLOY\_PROPERTIES\_FILE =<Path to external **deploy.properties** file>
4. It takes three parameters.
   * 1. Branch name
     2. Target tag number.
     3. Origin tag number. (Optional, required only for delta build)
5. ‘deploy \_build.sh’ will deploy the package (available in the same directory where the deploy\_build.sh file is) based on the input parameters.
6. **Environment configuration (To be maintained by PLM Admin): [deploy.properties**]

Providing this file to build process is optional, in case PLM admin choose **not to** provide this file through the SERVER\_DEPLOY\_PROPERTIES\_FILE configuration in deploy\_build.sh file, default deploy.properties file from GIT repository will used by the build process.

When external property is specified, it will override the values from the default deploy.properties file present in the GIT repo.

Create deploy.properties and add the following configuration.

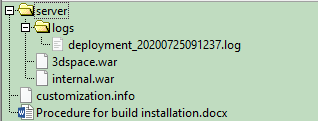
|  |  |  |
| --- | --- | --- |
| **Property Name** | **Description** | **To be updated** |
| ENV | represents the name of the environment, e.g. DEV, QA, PROD. | Yes |
| 3dspace\_nocas\_url | The No CAS URL of the environment | Yes |
| enovia.server.admin.user | Enovia admin user name (ex : creator) | Yes |
| enovia.server.admin.password | Enovia admin user password | Yes |
| SRV\_INSTALL | Server install 3DSpace path | Yes |
| enovia.sxi.config.servertags | Config.xml SERVER tag with hostname and baseport | Yes |
| platform.admin.user | Platform admin user name | Yes |
| platform.nocas.admin.password | Platform admin user password | Yes |
| platform.admin.context | Platform admin security context | Yes |
| tomee\_3dspace | Tomee 3dspace(CAS) directory full path : -> Distrib | Yes |
| tomee\_internal | Tomee internal(No Cas) directory full path : -> Distrib | Yes |
| enovia.war\_setup.cas.script | The script to generate CAS War file (BuildDeploy3DSpace\_CAS.sh) | No |
| enovia.war\_setup.nocas.script | The script to generate No Cas War file (BuildDeploy3DSpace\_NoCAS.sh) | No |
| enovia.webapp.cas.name | CAS directory name | No |
| enovia.webapp.nocas.name | No CAS directory name | No |

**Note: Above Configurations to be maintained by PLM deploy admin.**

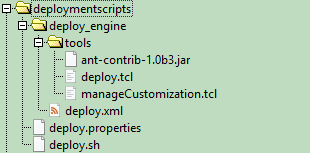
1. **Execution Output :**
   1. Script will generate deployable deployment package.
   2. Output package will be generated in same folder where ‘deploy\_build.sh’ script is present.
   3. Name of the output folder will follow below naming sequence.
      1. Full Deploy package : pwc.master.deployment.<N>.<ENV>
      2. Delta Deploy package : pwc.master.deployment.<N>\_<N + X>.<ENV>

**Note**: <N> represents the GIT Tag copied from the delta build root folder.

<ENV> represents the name of the environment, e.g. DEV, QA, PROD.



1. **Deployment Script Contents (In GIT Hub):**



1. **deploy.sh:**

deploy\_build.sh script will call ‘delpoy.sh’ file to deploy the build package available under server directory. This script intern calls the deploy.xml (ANT script).

1. **deploy.xml (ANT script):**
2. It loads the environment configurations from deploy.properties. If PLM admin configures external deploy.properties in the deploy\_build.sh, the external config file will override the default configurations.
3. As per the environment variable added in above step, this script is responsible to deploy the build package in the below order.

|  |  |  |
| --- | --- | --- |
| **#** | **Action** | **Ant Target Name** |
| 1 | Print OOTB and Custom Information and Check if the origin tag is same as the last build installed | print\_Info.build\_check |
| 2 | Generate OOTB WAR fies (CAS & NO CAS) | generate.OOTB\_war |
| 3 | Execute cleanup directory | execute.cleanup |
| 4 | Execute One Time scripts | execute.onetime\_scripts |
| 5 | Execute Pre Config scripts | execute.preconfig\_scripts |
| 6 | Import platform\_management Configuration | platform\_management.configuration |
| 7 | Import platform\_management PnO | platform\_management.PnO |
| 8 | Import 6W vocabularies | platform\_management.6WVocabularies |
| 9 | Import unified\_typing | unified\_typing |
| 10 | Execute Spinner | execute.spinner |
| 11 | Execute Post Config | execute.postconfig\_scripts |
| 12 | Register costomized information | register.customization |
| 13 | Mask compilation | mask\_compilation |
| 14 | Generate Custom WAR fies (CAS & NO CAS) | generate.custom\_war |
| 15 | JPO compilation | compile.JPOs |
| 16 | Import 3D Space Index | import.3DSpaceIndex |
| 17 | Create output package | create.output\_package |

1. **deploy.properties:**

This is environment configuration file.

1. **ant-contrib-1.0b3.jar:**

Third party library to support ANT looping mechanism.

1. **deploy.tcl:**

This TCL script has all the business logic to deploy the package.

1. **manageCustomization.tcl**

This TCL script has all the logic related to customization info.

**Note :**

* Cleanup directory will be executed only when the customization is available.
* One Time Scripts will be executed only when there is no customization.
* Make sure \*.tcl and \*.sh from package has execute access.

1. **Logs:**
   * logs’ folder will get created inside the Root Deployment directory inside the client directory.
   * ‘logs’ files will be generated in the same folder with the name of ‘deployment\_<timestamp>.log’.

**Client Deployment**

1. **Script configuration for PLM Admin :** [deploy\_client\_build.sh]
2. This file is not part of the source code. PLM deploy admin need to copy ‘deploy\_build.sh’ file to the build server. This is the entry point of the deployment process. PLM deploy admin just needs to execute ‘deploy\_ client\_build.sh to start the process.
3. PLM Deploy admin must update the following information in the script before starting the build process.
   * CLIENT\_DEPLOY\_PROPERTIES\_FILE =<Path to external **deploy.properties** file>
4. It takes three parameters.
   * 1. Branch name
     2. Target tag number.
     3. Origin tag number. (Optional, required only for delta build)
5. ‘deploy \_ client\_build.sh’ will deploy the package (available in the same directory where the deploy\_ client\_build.sh file is) based on the input parameters.
6. **Environment configuration (To be maintained by PLM Admin): [deploy.properties**]

Providing this file to build process is optional, in case PLM admin choose **not to** provide this file through the CLIENT\_DEPLOY\_PROPERTIES\_FILE configuration in deploy\_ client\_build.sh file, default deploy.properties file from GIT repository will used by the build process.

When external property is specified, it will override the values from the default deploy.properties file present in the GIT repo.

Create deploy.properties and add the following configuration.

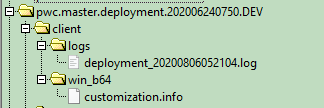
|  |  |  |
| --- | --- | --- |
| **Property Name** | **Description** | **To be updated** |
| ENV | represents the name of the environment, e.g. DEV, QA, PROD. | Yes |
| 3dspace\_nocas\_url | The No CAS URL of the environment | Yes |
| ootb.native.code.bin.dir | OOTB native code bin directory | Yes |
| shared.location | Shared location to distribute wib\_b64 | Yes |

**Note: Above Configurations to be maintained by PLM deploy admin.**

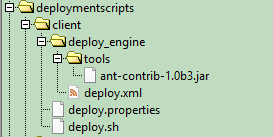
1. **Execution Output :**
   1. Script will generate deployable deployment package.
   2. Output package will be generated in same folder where ‘deploy\_client\_build.sh’ script is present.
   3. Name of the output folder will follow below naming sequence.
      1. Full Deploy package : pwc.master.deployment.<N>.<ENV>
      2. Delta Deploy package : pwc.master.deployment.<N>\_<N + X>.<ENV>

**Note**: <N> represents the GIT Tag copied from the delta build root folder.

<ENV> represents the name of the environment, e.g. DEV, QA, PROD.



**Deployment Script Contents (In GIT Hub):**



1. **deploy.sh :**

deploy\_build.sh script will call ‘delpoy.sh’ file to deploy the build package available under server directory. This script intern calls the deploy.xml (ANT script).

1. **deploy.xml (ANT script):**
2. It loads the environment configurations from deploy.properties. If PLM admin configures external deploy.properties in the deploy\_build.sh, the external config file will override the default configurations.
3. As per the environment variable added in above step, this script is responsible to deploy the build package in the below order.

|  |  |  |
| --- | --- | --- |
| **#** | **Action** | **Ant Target Name** |
| 1 | Import all the .3dxml files | import.resouce\_sets |
| 2 | Copy customization.info and distribute win\_b64 directory | copy\_customization.share\_winb64 |
| 3 | Create output package | create.output\_package |

1. **deploy.properties:**

This is environment configuration file.

1. **ant-contrib-1.0b3.jar:**

Third party library to support ANT looping mechanism.

**Note :**

* Make sure \*.tcl and \*.sh from package has execute access.

1. **Logs:**
   * ‘logs’ folder will get created inside the Root Deployment directory inside the client directory.
   * ‘logs’ files will be generated in the same folder with the name of ‘deployment\_<timestamp>.log’.