

Automation of Ant Build

Anuradha Singhal, Tarun Chander

Abstract— In today’s software world where developers try to automate everything , there should be some mechanism by which build.xml used for generation of enterprise archives can be build automatically . It will save developers effort and time.

Index Terms— ant build ,build file , jars , workspace , build file ..

I. INTRODUCTION

This document can be used for generating build.xml for any workspace provided name of open projects included with location on workstation are provided.

II. STEPS FOR DESIGNING

Let’s say, the workspace has many projects for which ear/war/jar need to be made .This utility helps to develop basic build.xml for deployment where all projects are compiled and jars of all are made and copied in one folder . To achieve this, we need to perform following steps.

- Write the name of workspace and also open projects in it along with their locations.
- Read the BuildInput.properties
- Search for a given file in given list of folders.
- Obtain .class path , .project , application.xml
- Parse the files to get
 - Dependent Projects
 - Dependent Libraries
 - Name of source folder
 - Name of output folder
- Store the result obtained in list and then sort the list on basis of dependent projects
- Write build.xml file.

III. REASON

For every new workspace developer needs to write new build.xml for deployment. This is wastage of time and effort .By this utility this issue can be resolved

IV. DETAILED SOLUTION

Unzip the required folder. Copy the module in workspace. The developer needs to make entry in Build Input. properties file like explained in Fig 1.

```
#workspace name along with location
workspace=C:\Practice

#list of projects in workspace along with location
project1=C:\Practice\DeployableApplication
project2=C:\Practice\MessageConversionModule
project3=C:\Practice\FirstSWTApplication
project4=C:\Practice\JavaPractice
project5=C:\Practice\org.eclipse.helloworldplugin
project6=C:\Practice\org.eclipse.swt
project7=C:\Practice\Servers
project8=C:\Practice\StrutsPractice
```

Fig. 1 A sample Input. properties showing entries

TABLE I
FUNCTIONS PROVIDED BY UTILITY CLASS

Function Name	Function Description
functiontoRead PropertiesFile	Read Properties File
gettingSource	Getting source code of classpath, .project application.xml for parsing and storing the source in list
settingForParsing	Parsing these files to get <ul style="list-style-type: none"> • Dependent Projects • Dependent Libraries • Name of source folder • Name of output folder
sortingOnbasisof NumberOfDependentProjects	Sorting the list
writingBuildFile	Writing to build.xml

Manuscript received March, 2014

Anuradha Singhal, Department of Computer Science , Delhi University /Deen Dayal Upadhyaya College/Delhi , India, 09650600697

Tarun Chander, Department of Computer Science, Kurukshetra University, Delhi , India , 9899880116,

V. CONCLUSION

This Utility is for generation for build.xml which provides basic infrastructure for making of deployable elements for developers without making effort for writing new build.xml for every new application .Meant Only for java based projects which uses ANT. It avoids Redundant coding and save in effort .

ACKNOWLEDGMENT

I wish to acknowledge my peers who helped me develop this module and used it in projects for better development

REFERENCES

- [1] <http://ant.apache.org/>
- [2] www.google.com

Anuradha Singhal MS(Software Systems) Bits Pilani , Assistant Professor , Deen Dayal Upadhyaya College , Delhi University , Worked as IT Analyst in IBM India .

Tarun Chander B.E Kuruksheetra University