Standard Operating Procedure

for

Java Card Build Process

ANT build process for Java Card Applications

Version Control

Document Name	Version	Date
SOP for Java Card Build Procedure	v1.1	22-03-2022
Prepared by	Technical Support Services	22-07-2020
Approved by	Technical Manager	*Saji Krishnan*

Revision History

Issue	Date	Reason for change
1	27-07-2020	First issue
1.1	22-03-2022	Minor changes made

Table of Contents

1.0 Purpose

1.1. Introduction

This Document describe the build procedure for javacard application, it describes all the

folder used in the process and Generation of AID of the applet.

1.2. Glossary

Term	Definition
AID	Application Identifier
JC	Java Card

2.0 Overall description

2.1. Use cases

The use of this procedure is for building a cap file out of java class files. Later, this cap can

be loaded into card and perform its functionality according to events configured.

A Java Card application is what which can be loaded onto a card having an OS and perform

its functionality, JC application is executed based on standard events defined in JC standards

like Event_profile_download, Event_first_command_after_reset etc. On reception of these

events' application runs its defined functionality.

2.2. Folder Description

1. **BIN**
2. **EXT**
3. **LIB**
4. **SRC**

2.2.1 BIN

Bin folder contain the output class file that was generated after compilation of the code
written.

2.2.2 EXT

EXT folder contain export files and libraries, it contains sub folders like api_export_file and

lib. Api_export_file contain all the EXP files used in build process, if any API/Function is

introduced later or multiple applets linked together then generated EXP file must be placed

in this folder. Name of the folder present in this folder are as per their package name and

last folder name must be '**javacard**'.

Folder placing is mentioned below:

Lib folder contain all the jars required for compilation of the code, if any functionality is

introduced later , then corresponding jar file must be placed in this folder so that it avoids

any compilation error shown by compiler.

2.2.3 LIB

LIB folder contain release library like UICC or USIM.

2.2.4 SRC

SRC folder contain source code files as per standard package name.

3.0 Build.xml configurations

Build.xml is used for ant build process and it contain all the configurational related settings.

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project basedir="." default="JavaCard" name="AppletPlayground build">
3    <!-- Applet building dependencies -->
4    <property name="JC222" value="ext/java_card_kit-2_2_2"/>
5      <property name="JC221" value="ext/java_card_kit-2_2_1"/>
6    <property name="JC303" value="ext/java_card_kit-3_0_3"/>
7    <property name="OP20" value="ext/visa_openplatform-2_0"/>
8    <property name="GP211" value="ext/globalplatform-2_1_1"/>
9
10   <!-- ant-javacard task from javacard.pro -->
11   <taskdef name="javacard" classname="pro.javacard.ant.JavaCard" classpath="ext/ant-javacard.jar"/>
12   <!-- All included applets -->
13   <target name="JavaCard">
14     <javacard jckit="${JC222}">
15       <!-- MuscleApplet -->
16       <cap output="inwistk.cap" sources="src/com/musclecard" verify="no" >
17         <applet class="com.musclecard.CardEdge.MenuStk" aid="A00000019400000000000000100000000"/>
18         <import exps="${JC222}" jar="${JC222}/102241_Annex_D.jar" />
19         <import exps="${JC222}" jar="${JC222}/31130_Annex_D.jar" />
20         <import exps="${JC222}" jar="${JC222}/lib/gpapi-globalplatform.jar" />
21       </cap>
22
23     </javacard>
24   </target>
25 </project>
26
```
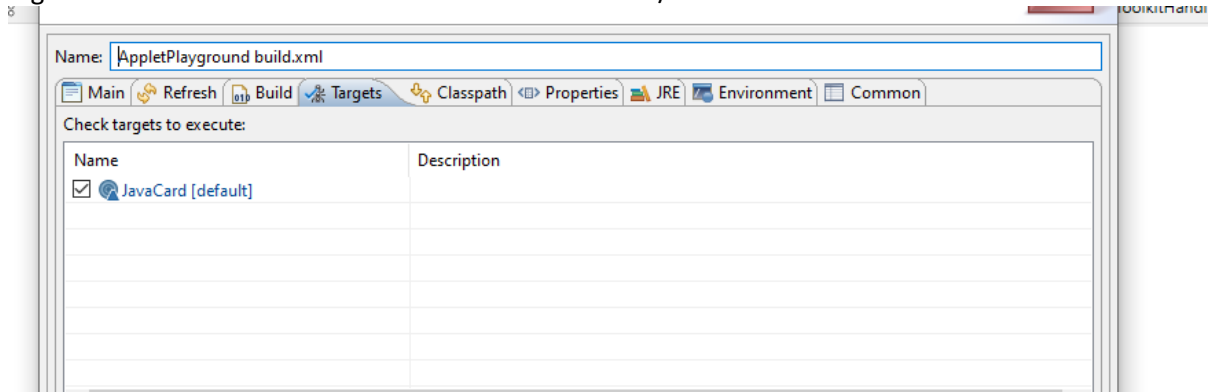
1. Property name: property name contain path for exp files against which conversion of cap will work.
2. Taskdef name: it contains the path for ant build jar and jar name.
3. Javacard jckit: it contains property name that we defined in point 1.
4. Cap output: it contains the name of output cap file
5. Sources: it contains the path for java files.
6. Verify: it verifies the cap against standard cap format, value can be yes or no, if yes then it verifies otherwise it will verify.
7. Applet class: it contains class name ex- package_name.class_name.
8. Aid: it contains applet AID and first five bytes of AID will be package id.
9. Import: it contains extra exp and jar files against which cap can be generated.

4.0 Build steps:

1. Create/import eclipse project.
2. Right click on build.xml and choose run as 2nd ant build/3rd ant

Name: AppletPlayground build.xml

Main | Refresh | Build | Targets | Classpath | Properties | JRE | Environment | Common

Check targets to execute:

Name	Description
☑ JavaCard [default]	

3.

4. Click on checkbox javacard as shown in above image and run.

TITLE	SOP_Javacard_Build_Process_v1.1
FILE NAME	SOP_Javacard_Build_Process_v1.1.pdf
DOCUMENT ID	ab8d6c161c9edb0ddc9e64024d95f2c200e9cf72
AUDIT TRAIL DATE FORMAT	DD / MM / YYYY
STATUS	● Signed

Document history

SENT	**01 / 06 / 2022** 18:18:26 UTC+4	Sent for signature to Saji Krishnan (saji.krishnan@workz.com) from tss@workz.com IP: 86.99.198.37
VIEWED	**01 / 06 / 2022** 18:19:04 UTC+4	Viewed by Saji Krishnan (saji.krishnan@workz.com) IP: 86.99.198.37
SIGNED	**01 / 06 / 2022** 18:19:23 UTC+4	Signed by Saji Krishnan (saji.krishnan@workz.com) IP: 86.99.198.37
COMPLETED	**01 / 06 / 2022** 18:19:23 UTC+4	The document has been completed.