

Comparative analysis of different CAP File Version and Identification of version of CAP File

Abhay Kumar Trivedi

Abstract- In this paper, I have explained the functional differences of CAP File versions. Also, this paper helps the user to identify the CAP file version from the CAP file.

Index Terms- CAP, JVM, Java Card.

1 Introduction

In Telecom World where a tiny piece of chip have a capability to compute the data and take some decision base on computation. The Oracle has provided the strong mechanism that makes this tiny piece of chip more powerful via downloading the applets. The loadable form of applet is CAP file which is defined as Converted **AP**plet.

Before defining, the different CAP files version; I would like to provide the brief information about the CAP file.

A Java Card CAP file contains a binary representation of a package of classes that can be installed on a device and used to execute the package's classes on a Java Card virtual machine. A CAP file is produced by a Java Card converter when a package of classes is converted.

A CAP file can be unzipped. When we unzipped the CAP file, you get the following component:

Header.cap: Contains general information about the CAP file.

Directory.cap: Lists the size of each of the components defined in the CAP file.

Applet.cap: Contains an entry for each of applets defined in the package.

Import.cap: Lists the set of packages imported by the classes in the package.

ConstantPool.cap: Contains an entry for each of the classes, methods, and fields referenced by elements in the Method Component.

Class.cap: Describes each of the classes and interfaces defined in the package.

Method.cap: Describes each of the methods declared in this package exclude interface method declaration.

StaticField.cap: Contains all of the information required to create and initialize an image of all of the static fields defined in the package.

RefLocation.cap: Contains all of the linked information that is used in the Method component

Export.cap: Lists all static elements in the package that may be imported by classes in other packages.

Descriptor.cap: Provides sufficient information to parse and verify all elements of the CAP file.

Debug.cap: Contains all necessary meta data for debugging a package.

Out of this Component; three components are optional: Export.cap, Debug.cap, Applet.cap.

The detailed descriptions of these components are present in the JVM Specifications.

2 Why CAP File version 2.2 came into picture

Since, there are some performance and other defects are present in the given Java Card version 2.1.1. Because of this, the newer version has been launched. The launched version 2.2.2 of Java Card has the following additional functionality:

Extended Addressing Capabilities

Previous versions of the Development Kit for the Java Card platform were released as a 16-bit implementation. For the 2.2.2 release, a 32-bit version of the Development Kit has been implemented.

Java Card WDE support for Java Card RMI

Java Card Workstation Development Environment ("Java Card WDE") now supports Java Card RMI (Remote Method Invocation). Any applet that uses Java Card RMI can now be run and debugged on Java Card WDE. The RMI demos in the Development Kit can now run on Java Card WDE.

Enhancement to the Applet Deletion Mechanism

Version 2.2.2 has added an AppletEvent.uninstall() method to allow you greater flexibility in deleting applets.

Cryptography Support (Development Kit with Cryptography Extensions only)

Java Card 2.2.2 supports cryptography with the implementation of security and cryptography classes.

3 Functional Differences

Missing Debug Component: In CAP file version 2.1, there is no debug component present. This component provides the debugging capability on suitably instrumented Java Card virtual machine. The structure of debug component is as follows:

```
debug_component {  
  u1 tag  
  u2 size  
  u2 string_count
```

```
utf8_info strings_table[string_count]
u2 package_name_index
u2 class_count
class_debug_info classes[class_count]
}
```

Structure of Header Component Changed: The name of package is now included in Header Component (i.e. Version 2.2) which was not previously included. Now, the structure of Header Component is as follows:

```
header_component {
u1 tag
u2 size
u4 magic
u1 minor_version
u1 major_version
u1 flags
package_info package
package_name_info package_name - Added
}
```

Structure of Class Component Changed: The structure of Class Component is changed in the version 2.2 which look like:

```
class_component {
u1 tag
u2 size
u2 signature_pool_length
type_descriptor signature_pool[] } Added
interface_info interfaces[]
class_info classes[]
}
```

The description of added component is:

signature_pool_length
The signature_pool_length item indicates the number of bytes in the signature_pool[] item. The value of the signature_pool_length item must be zero if the package does not define any remote interfaces or remote classes.

signature_pool[]
The signature_pool[] item represents a list of variable-length type_descriptor structures. These descriptors represent the signatures of the remote methods.

Exception Handling: In CAP file version 2.2, there are two exceptions that are not handled. These exceptions are: IllegalArgumentException, IllegalStateException.

4 How we Identify Version of CAP File

Suppose, you have a .cap file. Now you want to know the version of CAP file. For finding this information, you have to follow the following steps:

Unzip the .cap file

You get a folder under which numbers of .cap files are present

Open the cap file "**Header.cap**" in HexEditor tool[Trial version of this tool can be easily available on net]

Once you open this file, you get the some byte code. For us, let's assume the following byte code:
01 00 14 DE CA FF ED 01 02 04 00 01 0A D2 76 00 00 28 51 00 22 70 01

The breaup of this byte code is as follows:

01 – Tag (value COMPONENT_Header as describe in JVM Specification)
0014 – Size. The number of bytes in Header Component.
DE CA FF ED – Magic Number, used for identifying the Java Card CAP file format; its value has been fixed.
0102- Minor and Major version of CAP i.e the version of CAP file is 2.1 i.e. 02.01
04 – According to the JVM specification, the meaning of this flag is set if the Applet Component is included in the CAP file.
Package Information:
0001- Minor and Major version of package. This value is set by different vendors
0A – Application Identifier length
D1700000295100102001 – Package AID

In this way, we get the version of CAP file. Now you can easily identify the version of CAP through CAP file.

5 Acknowledgement

I would like to acknowledge my co-workers for supporting and encouraging me throughout the course work.

6 References

java_card_kit2_1-doc
java_card_kit-2_2-doc
<http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html>

7 Author

Author Name: Abhay Kumar Trivedi

Qualification /Experience: B.Tech and 3.4 years Experience

Email Address: abhay_trivedi222@yahoo.com