Archived:Identifying Java ME platform version

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

This information is now in the Java Developer Library: Determining the Java Runtime for Symbian version.

Overview

This code snippet demonstrates how to identify the platform version of the device using Java ME.

The platform version can be retrieved just like any system property using the method `System.getProperty`. The method returns a string containing the specified property.

Source

```
/**
 * Executes the snippet.
 * Gets platform version and prints it to the log.
 */
private void executeSnippet() {
    String platformVersion = System.getProperty("microedition.platform");
    if (platformVersion != null) {
        printString("Version: " + platformVersion);
    } else {
        printString("Version NOT found!");
    }
}
```

Postconditions

The platform version is displayed on the screen.

Supplementary material

This code snippet is part of the stub concept, which means that it has been patched on top of a template application in order to be more useful to developers. The version of the Java ME stub application used as a template in this snippet is v1.1.

- The patched, executable application that can be used to test the features described in this snippet is available for download at Media:IdentifyingPlatformVersion.zip.
- You can view all the changes that are required to implement the above-mentioned features. The changes are provided in unified diff and colour-coded diff (HTML) formats in Media:IdentifyingPlatformVersion.diff.zip.
- For general information on applying the patch, see Using Diffs.
- For unpatched stub applications, see Example app stubs with logging framework.