Identifying the Nokia platform in Java ME

This article explains how to identify the Nokia platform on which an application is running.

Introduction

Applications running on different Nokia Java ME platforms can have different behaviour - for example, applications built for the new Asha 501 should not show an EXIT command on the screen or we might want to set an OK command as an `IconCommand` in full touch devices like the Asha 311.

As a result, when we need to target both Series 40 and Asha platforms, it is useful to be able to identify the current platform and write conditional code where needed.

In this article we will demonstrate a way of programmatically identifying the Nokia platform on which a Java ME application is running.

The code

To identify the platform we will check the kind of keyboard the device has by using the `System.getProperty()` function asking for the value of the `com.nokia.keyboard.type` property. In the code below you can see an implementation of a function we could use for that purpose.

```
public static final int TouchType=0;
public static final int FullTouch=1;
public static final int OneKeyBack=2;
public static int currentPlatform=TouchType;

public static void identifyCurrentPlatform()
{
    String type=System.getProperty("com.nokia.keyboard.type");

    if (type!=null)
        type=type.toLowerCase();
    else
    {
        currentPlatform=TouchType;
        return;
    }

    if (type.equals("none"))
        currentPlatform=FullTouch;
    else if (type.equals("onekeyback"))
        currentPlatform=OneKeyBack;
    else
        currentPlatform=TouchType;
}
```

We should call this function at the application start-up and get the platform from the `currentPlatform` static variable whenever we need to differentiate our code according to the platform it is running on. In this case we need to target Touch&Type, Full Touch and New Asha devices. That's why the code above defaults and also falls back to Touch&Type.

Targeting more platforms

When targeting more platforms we need our check to be more complex. For example, both Symbian and Full Touch will return

(C) Copyright Nokia 2014. All rights reserved.

“none” as the keyboard type. In cases like this we can use the `Class.forName` function to identify the existence of a class in the platform. In our case we could pass `"com.nokia.mid.ui.IconCommand"` as the functions parameter. In Symbian we will get a `ClassNotFoundException`.

```
try
{
    Class.forName("com.nokia.mid.ui.IconCommand");
    //Full Touch
}
catch (ClassNotFoundException cnfe)
{
    //Symbian
}
```


(C) Copyright Nokia 2014. All rights reserved.