Detecting platform version at run time on Symbian

Overview

The following code snippets demonstrate how to detect the Symbian/S60 platform version at run time. This article presents two different methods, one for the S60 devices prior to S60 3rd Edition, FP2, and one for devices compatible with S60 3rd Edition, FP2.

S60 3rd Edition FP2 and later

This method uses the Version Info API available from S60 3rd Edition, FP2 onwards.

MMP file

Add the following libraries:

```
LIBRARY platformver.lib
```

Source file

```
#include <versioninfo.h>  // VersionInfo
```

```
// On return, aMajor and aMinor contain the version information
TInt GetS60PlatformVersionL(RFs& aFs, TUint& aMajor, TUint& aMinor)
    {
    VersionInfo::TPlatformVersion platformVersion;
    TInt ret = VersionInfo::GetVersion(platformVersion, aFs);
    if (ret == KErrNone)
        {
        aMajor = platformVersion.iMajorVersion;
        aMinor = platformVersion.iMinorVersion;
        }
    return ret;
    }
```

```
// Connect to the file server session
RFs fsSession;
User::LeaveIfError(fsSession.Connect());
CleanupClosePushL(fsSession);

// Obtain the version number
TUint major;
TUint minor;
User::LeaveIfError(GetS60PlatformVersionL(fsSession, major, minor));
CleanupStack::PopAndDestroy();  // fsSession

// Display the version number
TBuf<255> buffer;
_LIT(KTxt, "Version: %d.%d");
buffer.Format(KTxt, major, minor);
CAknInformationNote* note = new (ELeave)CAknInformationNote(ETrue);
note->ExecuteLD(buffer);
```

S60 3rd Edition FP1 and earlier

This method detects the version number from .sis files that are installed to the device (using `TFindFile`). It is suitable for versions up to, and including, S60 3rd Edition FP1

MMP file

Add the following libraries:

```
LIBRARY efsrv.lib
```

Source file

```
#include <f32file.h>  // TFindFile
```

```
// On return, aMajor and aMinor contain the highest version number found from
// the .sis file names
void GetS60PlatformVersionL(RFs& aFs, TUint& aMajor, TUint& aMinor)
    {
    TFindFile fileFinder = aFs;
```


(C) Copyright Nokia 2013. All rights reserved.

```
    CDir* result;

    _LIT(KS60ProductIDFile, "Series60v*.sis");
    _LIT(KROMInstallDir, "z:\\system\\install\\");

    User::LeaveIfError(fileFinder.FindWildByDir(KS60ProductIDFile,
            KROMInstallDir, result));
    CleanupStack::PushL(result);

    // Sort the file names so that the newest platforms are first
    User::LeaveIfError(result->Sort(ESortByName | EDescending));

    // Parse the version numbers from the file name (e.g. Series60v3_1.sis)
    aMajor = (*result)[0].iName[9] - '0';
    aMinor = (*result)[0].iName[11] - '0';
    CleanupStack::PopAndDestroy(result);
    }


// Code to display the version
// Connect to the file server session
RFs fsSession;
User::LeaveIfError(fsSession.Connect());
CleanupClosePushL(fsSession);

// Obtain the version number
TUint major;
TUint minor;
GetS60PlatformVersionL(fsSession, major, minor);
CleanupStack::PopAndDestroy();  // fsSession

// Display the version number
TBuf<255> buffer;
_LIT(KTxt, "Version: %d.%d");
buffer.Format(KTxt, major, minor);
CAknInformationNote* note = new (ELeave)CAknInformationNote(ETrue);
note->ExecuteLD(buffer);
```

Postconditions

S60 platform version is shown on the display.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.