Archived:Porting from S60 to UIQ3

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*}}.

UIQ3 no longer available or widely supported. It is possible, but unlikely that developers would want to use this article.

List of problems

- **Bringing the console application to the foreground**

The console application has the icon displayed on the main screen, while it is switch to the foreground without any problems on S60, on UIQ3 is each time started new process, so when I

```
GLDEF_C TInt E32Main()
{
  RMutex mutex;
    if ( mutex.OpenGlobal( KConsoleTestMutex ) == KErrNone )
  {
    CTrapCleanup* cleanup = CTrapCleanup::New();
    TRAPD( err, BringMeToForegroundL() );
    delete cleanup;
    mutex.Close();
    return error;
  }
  else
  {
    mutex.CreateGlobal( KConsoleTestMutex );
  }
  ...
}
void BringMeToForegroundL()
{
  TInt err = KErrNone;
  RWsSession ws;
  err = ws.Connect();
  if ( err != KErrNone )
  {
    User::LeaveIfError( err );
  }
  TApaTaskList taskList( ws );
  TApaTask task = taskList.FindApp( KConsoleAppName );
  if ( task.Exists() )
  {
    task.BringToForeground();
    err = KErrNone;
  }
  else
  {
    err = KErrAlreadyExists;
  }
  ws.Close();
  User::LeaveIfError( err );
}
```

Start console - notice the name **KConsoleAppName**:

```
...
CConsoleBase* console = Console::NewL( KConsoleAppName,
  TSize(KConsFullScreen,KConsFullScreen));
```

- **Winsock IAP is missing in CommDb and wrong RConnection capability**

To establish the connection from the emulator could be sometimes little tricky. I usually use RConnection class to establish the connection and the Winsock IAP on the S60 emulator. Unfortunately it does not exist on in the commDb on the UIQ emulator. The **uiqenv –eth** tool must be executed to create the new APN for LAN bearer in the database first. Details how this tool has to be used could be found here: http://developer.uiq.com/downloads/public/EnvConfig.pdf. In my case everything worked fine on the first attempt. The newly created profile name is **Emulator Ethernet**.

Another surprising thing was that the **RConnection::Start()** method needs the NetworkServices capability on UIQ3, while on S60 it runs without it.

(C) Copyright Nokia 2014. All rights reserved.

Limited SVG support

The UIQ3 platform surprisingly does not support SVG (UIQ3.1 already supports it), so all bitmaps have to be exported into bmp and consequently transformed into compound mbm file. There is possibility how to detect the SVG support on runtime mentioned here (the snippet just checks if the file **UIQSvgSupport.sis** is located in the directory **z:\ system\install**):
http://developer.uiq.com/devlib/uiq_31/SDKDocumentation/doc_source/guide/uiqhowtos/UIQVarious/UIQ_SVG/Detect_SVG_supported.guide.html#Detect_SVG_supported%2eguide

The disadvantage of missing SVG support will touch every ported S60 application at least once – when the application icon will be defined. The application icon on UIQ must be provided in three different sizes (64x64, 40x40 and 18x18), so when the **RESOURCE LOCALISABLE_APP_INFO** is defined, instead of putting into **icon_file** just the mif file name with the svg as on S60 platform, there should be the mbm with 3 icons provided with different sizes

```
RESOURCE LOCALISABLE_APP_INFO
{
  short_caption = "My application";
  caption_and_icon =
  {
    CAPTION_AND_ICON_INFO
    {
      caption=" My application ";
      number_of_icons=3;
      icon_file= "\\resource\\apps\\MyAppIcons.mbm";
    }
  };
}
```

The best way to define the sources for mbm is the mmp file (or mk file for gnumake):

```
START BITMAP MyAppIcons.mbm
HEADER
TARGETPATH Resource\Apps
SOURCEPATH      ..\gfx
SOURCE c24 Icon_Small.bmp
SOURCE 1 Icon _Small_mask.bmp
SOURCE c24 Icon _Large.bmp
SOURCE 1 Icon _Large_mask.bmp
SOURCE c24 Icon _xLarge.bmp
SOURCE 1 Icon _xLarge_mask.bmp
END
```

Transparent window

UIQ3 supports displaying transparent windows, while the method **RWindow::SetTransparencyFactor()** is not supported on S60 3rd (I have tested couple MR devices and N95, which is FP1 device) .

Choosing between different builds

There is quite a lot differences between S60 and UIQ applications and sometimes it could be useful to have different project files for both platforms. Unfortunately there is missing some defined mechanism, how to differentiate between different SDKs during build time (i.e. there are not some macros, which can be easily used in build time). I found useful (especially for build scripts) to differentiate on the **bld.inf** level the MMP file to be used:

```
PRJ_MMPFILES
#if defined(UIQ_UMTS_AVAILABLE) && defined(EKA2)
  // UIQ3
  Application_uiq3.mmp
#else
  Application_S60.mmp
#endif
```


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.