Archived:UIDs for some Symbian apps are device dependent (Known Issue)

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*}}.

UIDs for certain applications, for example Camera/Camcorder, can be different between devices within a S60 Edition (or even within a Feature Pack). This is relevant when trying to launch external application views, or using APIs that require application UIDs to be specified, such as the New File Service Client API.

Description

The New File Service API for S60 3rd Edition can be used for recording video, audio, and capturing images. This API is not part of the official S60 SDK but can be found in the Extensions plug-in package #1 for S60 3rd Edition SDK for Symbian OS, for C++, MR. Note that Nokia is not giving any binary compatibility promise for these APIs; neither between platform versions or within a platform version.

The Camera/Camcorder application UID is not the same for all S60 3rd Edition devices. This causes problems when using the `CNewFileServiceClient::NewFileL()` function. An example of API usage:

```
CDesCArray* selectedFiles = new (ELeave) CDesCArrayFlat(4);
CNewFileServiceClient* fileClient = NewFileServiceFactory::NewClientL();
TBool createOK = EFalse;
TBool multiFileSelection = EFalse;
//...

CNewFileServiceClient* fileClient = NewFileServiceFactory::NewClientL();
createOK = fileClient->NewFileL(
       KUidCamcorder,
        *selectedFiles,
        paramList,
        ENewFileServiceImage, // Note: ENewFileServiceImage is also used for video recording.
        multiFileSelection );
//...
```

UIDs for the different camera applications are:<

- Nokia N73 (and other Nseries devices): 0x101FFA86
- Basic S60 Camcorder app. UID: 0x101F857A, Required capability: WriteDeviceData

Solution

The UID for a application supporting a specified MIME type can be resolved at run time with RApaLsSession API, see the documentation for:

```
TInt RApaLsSession::AppForDataType(const TDataType &aDataType, TUid &aAppUid);
```

See also: "S60 Platform: Application views" available at Nokia Developer

(c) Copyright Nokia 2013. All rights reserved.

 Copyright Nokia 2013. All rights reserved.