Archived:Profiles Engine Active Profile Settings Symbian API

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*}}.

Note: :This API is not part of the public SDK. It can be found in the SDK API Plug-in.

Purpose

The following links point to the Profile Engine APIs for S60 3rd Edition, Maintenance Release:

Profiles Engine API

Profiles Engine Extended API

The S60 3rd Edition, Feature Pack 2 plug-in package has Central Repository keys for

- Retrieving and setting the DefaultRingingTone, DefaultMessageTone, DefaultImTone, and RingingToneMaxSize.
- Checking if a particular feature is supported or not.
- Retrieving the set of MIME types that are not allowed as Alert Tones to be excluded from the list of tones in the Tone Selection dialog, (for example, DRM protected, demo played).

Header files

```
centralrepository.h
ProfileEngineDomainCRKeys.h
ProfilesVariant.hrh
```

Libraries

```
centralrepository.lib
```

Code snippets

```
iRepository = CRepository::NewL(KCRUidProfileEngine);
```

Retrieving the default `RingingTone`:

```
TBuf<50> buffer;
User::LeaveIfError( iRepository->Get( KProEngDefaultRingingTone, buffer ) );
CEikonEnv::InfoWinL(_L("default ringing tone"), buffer);
```

Retrieving the default `MessageTone`:

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngDefaultMessageTone, buffer ) );
CEikonEnv::InfoWinL(_L("default Message tone"), buffer);
```


(C) Copyright Nokia 2014. All rights reserved.

Retrieving the maximum size of the `RingingTone`:

```
TInt size;
User::LeaveIfError( iRepository->Get( KProEngRingingToneMaxSize, size ) );
buffer.Zero();
buffer.AppendNum(size);
CEikonEnv::InfoWinL(_L("RingingToneMaxSize"), buffer);
```

KCRUidProfilesLV, Profiles local variation CR file.

```
iRepository = CRepository::NewL(KCRUidProfilesLV);
```

Retrieving the Standard bitmask type of key used for local variation of profiles:

```
TInt featureBitmask( 0 );
User::LeaveIfError(iRepository->Get( KProfilesLVFlags, featureBitmask ));

if  ( featureBitmask & KProEngFeatureIdVTRingingTone )
{
// This feature is ON, so act accordingly
CEikonEnv::InfoWinL(_L("ProEngFeatureIdVTRingingTone"), _L("Supported"));
}
else
{
CEikonEnv::InfoWinL(_L("ProEngFeatureIdVTRingingTone"), _L("Not Supported"));
}
if  ( featureBitmask & KProEngFeatureIdForceSilentMode )
{
// This feature is ON, so act accordingly
CEikonEnv::InfoWinL(_L"KProEngFeatureIdForceSilentMode"), _L("Supported"));
}
else
{
CEikonEnv::InfoWinL(_L("KProEngFeatureIdForceSilentMode"), _L("Not Supported"));
}
if  ( featureBitmask & KProEngFeatureIdTimedProfiles )
{
// This feature is ON, so act accordingly
CEikonEnv::InfoWinL(_L("KProEngFeatureIdTimedProfiles"), _L("Supported"));
}
else
{
CEikonEnv::InfoWinL(_L("KProEngFeatureIdTimedProfiles"), _L("Not Supported"));
}
if  ( featureBitmask & KProEngFeatureIdTTSRingingTone )
{
// This feature is ON, so act accordingly
CEikonEnv::InfoWinL(_L("KProEngFeatureIdTTSRingingTone"), _L("Supported"));
}
else
{
CEikonEnv::InfoWinL(_L("KProEngFeatureIdTTSRingingTone"), _L("Not Supported"));
}
```


(C) Copyright Nokia 2014. All rights reserved.

Retrieving the TonesViewFeatureIdHac if it is enabled or disabled:

```
TInt flag = 0;
User::LeaveIfError( iRepository->Get( KProEngTonesViewLVFlags, flag ) );
buffer.Zero();
buffer.AppendNum(flag);
CEikonEnv::InfoWinL(_L("KProEngTonesViewLVFlags"), buffer);
```

Retrieving the set of MIME types that are not allowed as alert tones.

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngBlockedTones, buffer ) );
CEikonEnv::InfoWinL(_L("KProEngBlockedTones"), buffer);
```

Retrieving the the set of MIME types that are not allowed as alert tones (not DRM protected):

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngBlockedUnprotectedTones, buffer ) );
CEikonEnv::InfoWinL(_L("KProEngBlockedUnprotectedTones"), buffer);
```

Retrieving the the set of MIME types that are not allowed as alert tones (DRM protected):

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngBlockedProtectedTones, buffer ) );
CEikonEnv::InfoWinL(_L("KProEngBlockedProtectedTones"), buffer);
```

Retrieving the set of MIME types that are not allowed as demo played in the tone selection dialog:

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngBlockedDemoPlayTones, buffer ) );
CEikonEnv::InfoWinL(_L("KProEngBlockedDemoPlayTones"), buffer);
```

Retrieving the set of MIME types that must be excluded from the list of tones in the tone selection dialog:

```
buffer.Zero();
User::LeaveIfError( iRepository->Get( KProEngExcludedTones, buffer ) );
CEikonEnv::InfoWinL(_L("KProEngExcludedTones"), buffer);
```

Example project

File:ProfileCRKey.zip

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.