

**Note!**

This API is not part of the public SDK. It can be found in the [SDK API Plug-in](#).

Extended Skins API provides classes to change active, to get skin package information.

Use cases

To change the active theme.

To receive notifications of change in skin content, configuration.

To get skin package information.

To delete skin, copy skin from one location to another.

To set idle wallpaper.

Example code

Header files

```
#include <AknSSrvClient.h>
#include <AknsPkgID.h>
#include <AknsItemDef.h>
#include <AknsItemID.h>
#include <AknsSrvSkinInformationPkg.h>
#include <EIKENV.H>
#include <centralrepository.h>
#include <AknSkinsInternalCRKeys.h>
```

Libraries Used

```
aknskinsrv.lib centralrepository.lib
```

Capability required

```
CAPABILITY WriteDeviceData
```

First connect to the skin server:

```
void CNotifyEngine::ConstructL()
{
//connect to skin server
```

Use cases

Extended_Skins_API

```
User::LeaveIfError( iSkinSrvSession.Connect( this ) );
// to receive notifications of skin change
iSkinSrvSession.EnableSkinChangeNotify();
}
```

Virtual functions

```
void CNotifyEngine::SkinContentChanged()
{
    CEikonEnv::Static()->InfoWinL(_L("Skin Content Changed"),_L(""));
}
void CNotifyEngine::SkinConfigurationChanged(const TAKnsSkinStatusConfigurationChangeReason aReason)
{
    CEikonEnv::Static()->InfoWinL(_L("Skin Configuration Changed"),_L(""));
}
void CNotifyEngine::SkinPackageChanged(const TAKnsSkinStatusPackageChangeReason aReason)
{
    CEikonEnv::Static()->InfoWinL(_L("Skin Package Changed"),_L(""));
}
```

The following code displays current active theme details:

```
void CNotifyEngine::SkinNameL()
{
    TAKnsPkgIDBuf cidBuf;
    CRepository* repository = CRepository::NewLC( KCRUidPersonalisation );
    TInt retVal = repository->Get( KPslnActiveSkinUid, cidBuf );
    TAKnsPkgID cpkgId;
    cpkgId.SetFromDesL( cidBuf );
    CleanupStack::PopAndDestroy(repository);
    CArrayPtr< CAknsSrvSkinInformationPkg >* skinInfoArray = iSkinSrvSession.EnumerateSkinPackagesL();
    CleanupStack::PushL( skinInfoArray );
    TInt retValue( KErrNone );
    if ( skinInfoArray->Count() > 0 )
    {
        for (TInt i=0; i < skinInfoArray->Count(); i++)
        {
            TAKnsPkgID pkgId = skinInfoArray->At(i)->PID();
            if ( pkgId == cpkgId )
            {
                TBuf<128> lBuf = skinInfoArray->At(i)->Name();
                CEikonEnv::Static()->InfoWinL(_L("Skin Name"),lBuf);
                lBuf.Zero();
                lBuf = skinInfoArray->At(i)->FullName();
                CEikonEnv::Static()->InfoWinL(_L("Skin Path"),lBuf);
                lBuf.Zero();
                lBuf = skinInfoArray->At(i)->Directory();
                CEikonEnv::Static()->InfoWinL(_L("Skin Directory"),lBuf);
                lBuf.Zero();
                lBuf = skinInfoArray->At(i)->IniFileDirectory();
                CEikonEnv::Static()->InfoWinL(_L("Skin IniFile Directory"),lBuf);
                break;
            }
        }
    }
    skinInfoArray->ResetAndDestroy();
    CleanupStack::PopAndDestroy();
}
```

The following code changes the active theme:

```

void CNotifyEngine::ActivateSkinL()
{
StoreCurrentSkinIdL();
CArrayPtr< CAknsSrvSkinInformationPkg >* skinInfoArray = iSkinSrvSession.EnumerateSkinPackagesL();
CleanupStack::PushL( skinInfoArray );
TInt retVal( KErrNone );
if ( skinInfoArray->Count() > 0 )
    {
    for (TInt i=0; i < skinInfoArray->Count(); i++ )
        {
        TAknsPkgID pkgId = skinInfoArray->At(i)->PID();
        if ( pkgId != iOriginalSkinPid )
            {
            TBuf<100> lName = skinInfoArray->At(i)->Name();
            CEikonEnv::Static()->InfoWinL(_L("New Skin Name"),lName);
            //Activate a complete skin package at once.
            retVal = iSkinSrvSession.SetAllDefinitionSets( pkgId );
            if ( retVal == KErrNone )
                {
                SetNewSkinIdL( pkgId );
                }
            break;
            }
        }
    }
skinInfoArray->ResetAndDestroy();
CleanupStack::PopAndDestroy();
}

void CNotifyEngine::StoreCurrentSkinIdL()
{
TAknsPkgIDBuf pidBuf;
CRepository* repository = CRepository::NewLC( KCRUidPersonalisation );
TInt retVal = repository->Get( KPslnActiveSkinUid, pidBuf );
CleanupStack::PopAndDestroy(repository);
// iOriginalSkinPid is of type TAknsPkgID
iOriginalSkinPid.SetFromDesL( pidBuf );
}

void CNotifyEngine::SetNewSkinIdL( TAknsPkgID aPkgId )
{
TAknsPkgIDBuf pidBuf;
aPkgId.CopyToDes( pidBuf );
CRepository* repository = CRepository::NewLC( KCRUidPersonalisation );
TInt retVal = repository->Set( KPslnActiveSkinUid, pidBuf );
// KPslnActiveSkinLocation has to be changed also
// if new skin resides on mmc whereas old one resided in phone memory
CleanupStack::PopAndDestroy(repository);
}

```

Sample application

<http://wiki.forum.nokia.com/index.php/Image:ExtSkins.zip>

Related topics

- [How to change the active theme](#)