

This document will very briefly explain how to change the active theme on a [S60 3rd Edition](#) device. The information in this document is provided as is and the code samples are not meant to be of commercial quality.

Please note that you need to download and install the [Extensions Plug-ins](#) for your S60 SDKs to gain access to [APIs](#) you need. Extensions Plug-ins are available for all S60 3rd Edition and S60 5th Edition SDKs. From the Extensions plug-ins you will need ExtendedSkinsAPI and PersonalisationSkinsSettingsAPI.

Contents

- [1 Connect to Skin Server](#)
- [2 Fetch the package ID of the current skin](#)
- [3 Get a list of installed skin packages](#)
- [4 Go through installed packages and switch to the first available new skin](#)
- [5 Save the new skin package ID in central repository](#)
- [6 Free allocated resources](#)
- [7 Whole Source](#)
- [8 Required capabilities](#)
- [9 Related topics](#)

Connect to Skin Server

```
#include <AknSSrvClient.h> // aknskinsrv.lib
?
RAknsSrvSession skinsSession;
User::LeaveIfError( skinsSession.Connect( this ) );
CleanupClosePushL( skinsSession );
```

Fetch the package ID of the current skin

```
#include <centralrepository.h> // centralrepository.lib
#include <AknSkinsInternalCRKeys.h>
?

// iOriginalSkinPid is of type TAknsPkgID
void CMyThemeManager::GetCurrentSkinIdL()
{
    TAknsPkgIDBuf pidBuf;
    CRepository* repository =
        CRepository::NewL( KCRUidPersonalisation );
    TInt retVal = repository->Get( KPslnActiveSkinUid, pidBuf );
    delete repository;
    repository = NULL;
    iOriginalSkinPid.SetFromDesL( pidBuf );
}
```

Get a list of installed skin packages

Please see *aknssrvclient.h* file for more details on how to work with skins residing on memory card.

```
CArrayPtr< CAknsSrvSkinInformationPkg >* skinInfoArray =
    skinsSession.EnumerateSkinPackagesL();
CleanupStack::PushL( skinInfoArray );
```

Go through installed packages and switch to the first available new skin

```
TInt retVal( KErrNone );
TInt i( 0 );
if ( skinInfoArray->Count() > 0 )
{
    for ( i; i < skinInfoArray->Count(); i++ )
    {
        TAKnsPkgID pkgId = skinInfoArray->At( i )->PID();
        if ( pkgId != iOriginalSkinPid )
        {
            //Activate a complete skin package at once.
            retVal = skinsSession.SetAllDefinitionSets( pkgId );
            if ( retVal == KErrNone )
            {
                SetNewSkinIdL( pkgId );
            }
            break;
        }
    }
}
```

Save the new skin package ID in central repository

```
void CMyThemeManager::SetNewSkinIdL( TAKnsPkgID aPkgId )
{
    TAKnsPkgIDBuf pidBuf;
    aPkgId.CopyToDes( pidBuf );
    CRepository* repository =
        CRepository::NewL( 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 mem
    // ...and vice versa

    delete repository;
    repository = NULL;
}
```

Free allocated resources

```
skinInfoArray->ResetAndDestroy();
CleanupStack::PopAndDestroy( 2 ); // skinsSession, skinInfoArray
```

Whole Source

```

#include <AknSSrvClient.h> //aknskinsrv.lib
#include <centralrepository.h> //centralrepository.lib
#include <AknSkinsInternalCRKeys.h>
#include <mmfcontrollerpluginresolver.h> // CleanupResetAndDestroyPushL

class CMyThemeManager : public CBase, public MAknsSkinChangeObserver
{
public:

    void ChangeSkinL();

private:

    //From MAknsSkinChangeObserver
    virtual void SkinContentChanged();
    virtual void SkinConfigurationChanged(const TAKnsSkinStatusConfigurationChangeReason aReason);
    virtual void SkinPackageChanged(const TAKnsSkinStatusPackageChangeReason aReason);
    //
    void GetCurrentSkinIdL();
    void SetNewSkinIdL(TAknsPkgID aPkgId);

private:
    TAKnsPkgID iOriginalSkinPid;
};

/*
 * Set the current skin ID into the Central Repository
 */
void CMyThemeManager::SetNewSkinIdL( TAKnsPkgID aPkgId )
{
    TAKnsPkgIDBuf pidBuf;
    aPkgId.CopyToDes( pidBuf );
    CRepository* repository = CRepository::NewLC(KCRUidPersonalisation);
    User::LeaveIfError(repository->Set(KPslnActiveSkinUid, pidBuf));

    // KPslnActiveSkinLocation has to be changed also
    // if new skin resides on mmc whereas old one resided in phone mem
    // ...and vice versa

    CleanupStack::PopAndDestroy(repository);
}

/*
 * Get the current skin ID from the Central Repository
 */
void CMyThemeManager::GetCurrentSkinIdL()
{
    TAKnsPkgIDBuf pidBuf;
    CRepository* repository = CRepository::NewLC( KCRUidPersonalisation );
    User::LeaveIfError(repository->Get(KPslnActiveSkinUid, pidBuf));
    CleanupStack::PopAndDestroy(repository);
    iOriginalSkinPid.SetFromDesL(pidBuf);
}

/*
 * Change the current skin to the next available skin

```

How_to_change_the_active_theme

```
*/
void CMyThemeManager::ChangeSkinL()
{
//Connect to the skin server
    RAKnsSrvSession skinsSession;
    ::LeaveIfError( skinsSession.Connect( this ) );
    CleanupCloseRushSession();

//Get the current skin package
    GetCurrentSkinIdL();

//Get the list of installed skin packages
    CAknsSrvSkinInformationPkg>* skinInfoArray =skinsSession.EnumerateSkinPackagesL();
    CleanupSkin(skinInfoArray);
    CleanupResetAndDestroy(skinInfoArray);

//Go through installed skin and change it
for (TInt i=0; i < skinInfoArray->Count(); i++ )
{
    TAKnsSkinInfoPkg* pkgId =skinInfoArray->At( i )->PID();
if ( pkgId != iOriginalSkinPid )
{
//Activate a complete skin package at once.
    ::LeaveIfError(skinsSession.SetAllDefinitionSets(pkgId));
    (pkgId); SetNewSkinIdL();
break; //stop the loop
}
}

//Cleanup everything
    CleanupSkinAndDestroy(3,&skinsSession); //skinInfoArray, &skinInfoArray
}

void CMyThemeManager::SkinContentChanged()
{
//TODO:
}

void CMyThemeManager::SkinConfigurationChanged(const TAKnsSkinStatusConfigurationChangeReason /*aReason*/)
{
//TODO:
}

void CMyThemeManager::SkinPackageChanged(const TAKnsSkinStatusPackageChangeReason /*aReason*/)
{
//TODO:
}
```

Required capabilities

WriteDeviceData

Related topics

- [Extended Skins API](#)
- <http://discussion.forum.nokia.com/forum/showthread.php?t=129723>
- <http://discussion.forum.nokia.com/forum/showthread.php?p=555185>
- <http://discussion.forum.nokia.com/forum/showthread.php?t=154995>
- [http://wiki.forum.nokia.com/index.php/TSS001174 - Changing the skin of the status pane](http://wiki.forum.nokia.com/index.php/TSS001174)