

## How\_to\_use\_the\_CAkWaitDialog

This example will demonstrate the use of CAknWaitDialog in a typical Symbian C++ source code:

### In the mmp file:

```
LIBRARY      avkon.lib
LIBRARY      eikcdlg.lib
LIBRARY      eikctl.lib
```

### In the header file:

```
// System Includes
#include <aknprogressdialog.h> // for MProgressDialogCallback

// FORWARD DECLARATIONS
class CAknWaitDialog;

class CMyClass : public CBase,
                 public MTestActiveObjNotifier,
                 public MProgressDialogCallback
{
    // ...

private: // From MTestActiveObjNotifier

    /**
     * Get the notification about the connection
     * from the registered CTestActiveObj.
     */
    void NotifyConnection(TInt aError);

private:// From MProgressDialogCallback

    /**
     * Get's called when a dialog is dismissed.
     */
    void DialogDismissedL(TInt aButtonId);

    // ...
private: // New functions

    /**
     * Stars the wait dialog.
     * \param aResourceId The dialog resource id.
     */
    void StartWaitDialog(const TDesC& aLabel);

    /**
     * Stops the wait dialog.
     */
    void StopWaitDialog();
```

In the mmp file:

## How\_to\_use\_the\_CAKnWaitDialog

```
// ...
private: // Member data

    /**
     * TODO: This is a samle Active Object class,
     * use your AO here.
     * Owned by CMyClass.
     */
    CTestActiveObj* iTestActiveObj;

    /**
     * The wait dialog.
     * Owned by CMyClass.
     */
    CAKnWaitDialog* iWaitDialog;
```

### In the cpp file:

```
// System includes
#include <aknwaitdialog.h> // for CAKnWaitDialog

// -----
// CMyClass::~~CMyClass()
// Destructor
// -----
//
CMyClass::~~CMyClass()
{
    // ...
    if(iWaitDialog)
    {
        delete iWaitDialog;
        iWaitDialog = NULL;
    }

    if(iTestActiveObj)
    {
        delete iTestActiveObj;
        iTestActiveObj = NULL;
    }
    // ...
}

// -----
// CMyClass::ConstructL()
// Two phase constructor
// -----
//
void CMyClass::ConstructL()
{
    // TODO: Register the Active Object here.
    // In this example it'll call the NotifyConnection()
    // after the connection.
    iTestActiveObj = CTestActiveObj::NewL(*this);
}
```

In the header file:

## How\_to\_use\_the\_CAkWaitDialog

```
// -----  
// CMyClass::DialogDismissedL(TInt aButtonId)  
// Get's called when a dialog is dismissed.  
// -----  
//  
void CMyClass::DialogDismissedL(TInt /*aButtonId*/)  
{  
    // No implementation required.  
}  
  
// -----  
// CMyClass::StartWaitDialog(const TDesC& aLabel)  
// Stars the wait dialog.  
// -----  
//  
void CMyClass::StartWaitDialog(const TDesC& aLabel)  
{  
    if(iWaitDialog)  
    {  
        delete iWaitDialog;  
        iWaitDialog = NULL;  
    }  
  
    // For the wait dialog  
    iWaitDialog = new (ELeave) CAknWaitDialog(  
        REINTERPRET_CAST(CEikDialog**, &iWaitDialog));  
    iWaitDialog->SetCallback(this);  
    iWaitDialog->SetTextL(aLabel);  
    iWaitDialog->ExecuteLD(1); //R_TEST_WAITNOTE);  
}  
  
// -----  
// CMyClass::StopWaitDialog()  
// Stops the wait dialog.  
// -----  
//  
void CMyClass::StopWaitDialog()  
{  
    // For wait dialog  
    iWaitDialog->ProcessFinishedL();  
    iWaitDialog = NULL;  
}  
  
// -----  
// CMyClass::MakeConnection()  
// Create a connection.  
// -----  
//  
void CMyClass::MakeConnection()  
{  
    // TODO: The below function call is a sample here  
    // sending some request to an Active Object  
    iTestActiveObj->Connect();  
  
    // Now start the wait dialog.  
    StartWaitDialog(_L("Connecting..."));  
}  
  
// -----  
// CMyClass::NotifyConnection()  
// This will notify when ever a connection happend.  
// -----
```

In the cpp file:

## How\_to\_use\_the\_CAKnWaitDialog

```
//  
void CMyClass::NotifyConnection(TInt /*aError*/)  
{  
    // This is also a sample implementation.  
  
    // Now stop the wait dialog when ever a connectin  
    // notification has come.  
    StopWaitDialog();  
}
```

**Note:** You should have a basic understanding of Active Objects and the knowledge of the working of Mixin-class.