

## **Advanced NXN Alienbrain 7.5 integration to UnrealEd3**

### **by 10TACLE STUDIOS**

(for QA builds 108143 to 121834)

#### **Preface**

This document describes how to implement the advanced integration of NXN Alienbrain 7.5 to Unreal Editor 3. UnrealEd itself provides a default integration functionality by using the standard Source Code Control (SCC) provider as a asset manager client.

However, the default integration of SCC provider in UnrealEd3 does not update the game content assets automatically, meaning that when a new/newer version of an asset is available on the server (depot), it does not use the „get latest version“ function to update the user's local data/assets. The result is, that user works with obsolete data assets and can even result in loss of data (commit/check in of a map with old references to objects in packages).

To avoid this unwanted behavior, we decided to add some advanced asset management functionality to UnrealEd3.

This document describes how to add the mentioned advanced functionality for NXN Alienbrain 7.5 and newer, however this integration solution could be useful as a guide for users which are using other asset management solutions as well.

As mentioned, UnrealEd3 is using SCC provider as an asset manager client. Therefore, install Alienbrain integration for Visual Studio on all computers where you want to use it.

Advanced functionality:

- Update of maps and packages on editor start up
- Download of maps/packages when new/newer is available while map is opening
- NXN Alienbrain browser

## User's guide

- Update of maps and packages on editor start up

When editor is started and any updates are available (new/newer maps/packages) a message box appears:

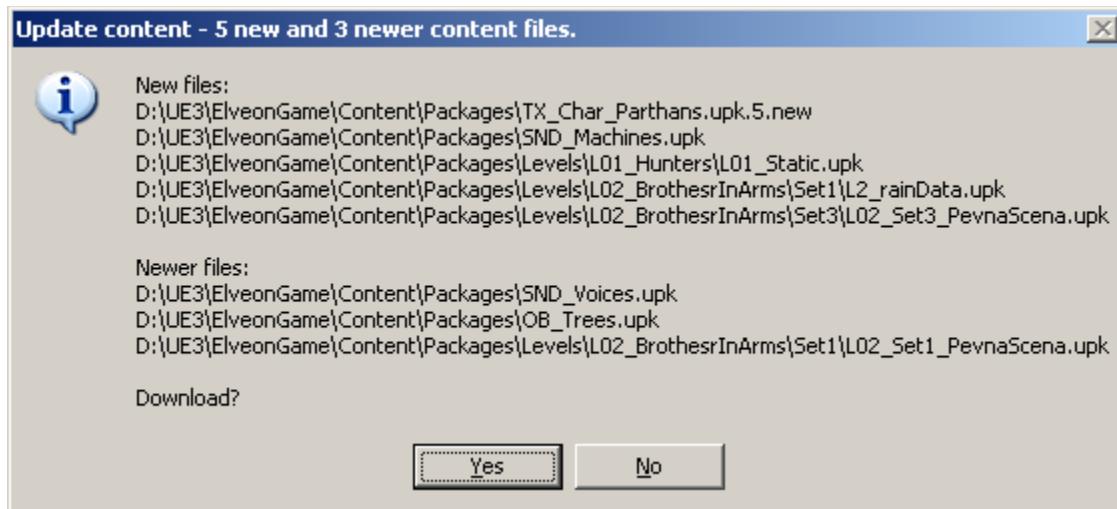


Figure 1 – Update content

If you choose Yes, all listed files will be downloaded file by file:

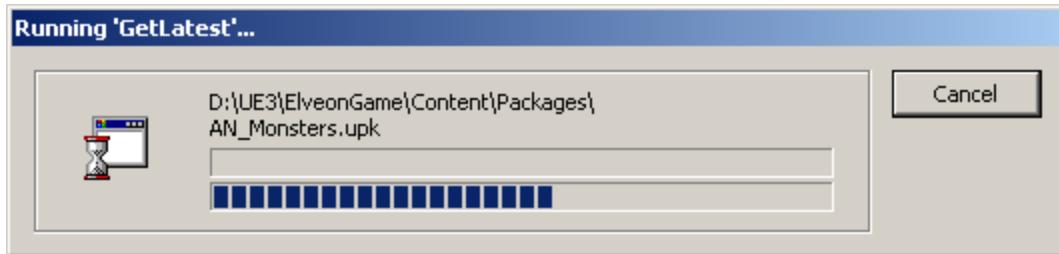


Figure 2 – downloading file

NOTE: don't press Cancel button to terminate downloading, hold Ctrl key between files instead.

NOTE: all folders which contain map/packages must exist locally, otherwise your new/newer files will not be downloaded

- Downloading maps/pakages when new/newer is available whenever map is opened

When you open a map from editor and newer version is available in depot, the newer map is automatically downloaded. While opening map all new/newer pakages which are referenced by the map are downloaded, too.

NOTE: all folders which contain maps/packages must exist locally, otherwise your new/newer files will not be downloaded

- NXN Alienbrain browser

New item 'NxN Browser' was added to Generic Browser Source Control menu:

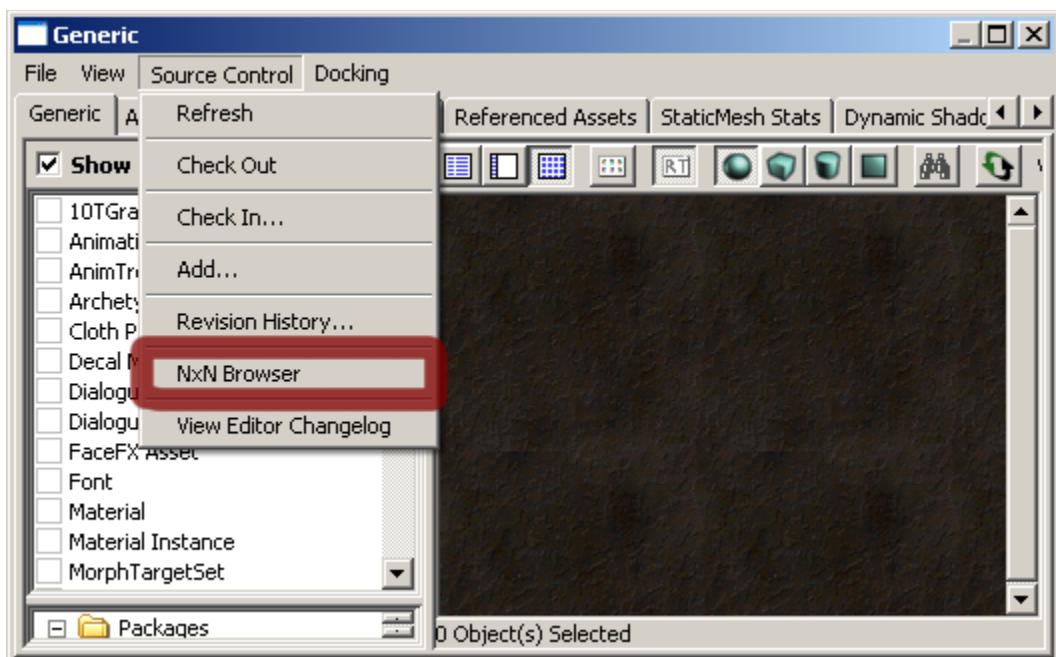


Figure 3 – Generic Browser

By choosing this item you can open Alienbrain Database Explorer to perform any operation provided by NXN Alienbrain directly from UnrealEd3.

## Programmer's guide

Follow these steps to make the integration running:

1) Copy *SourceControlIntegration.cpp* and *SourceControlIntegration.h* over your files in UnrealEd project (this files have not changed through QA builds 112837 to 121834, if you use other QA build you have to merge them).

2) Update section [SourceControl] in *DefaultEditor.ini*:

- add two new keys *DisableSccGet* and *DisableUpdateMapsAndPackages*
- set *ProjectPath* key (be sure your <alienbrain project name> working path points to Content folder of your game)

```
[SourceControl]
Disabled=False
DisableSccGet=False
DisableUpdateMapsAndPackages=False
LocalPath=
ProjectPath=\Workspace\<alienbrain project name>\
AuxProjectPath=
```

### 3) Update *UnLinker.cpp* (Core project):

- we're going to use SCC in this file, include it

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
#include "..\\..\\UnrealEd\\Inc\\SourceControlIntegration.h"
#endif // CONSOLE
#endif // SCC_NXN_75
```

The screenshot shows the Unreal Engine 3 development environment with the *UnLinker.cpp* file open in the code editor. The code editor window has tabs for *UnLinker.cpp*, *UnObj.cpp*, *UnrealEdApp.cpp*, *GenericBrowser.cpp*, *ResourceIDs.h*, *Launch.cpp*, *SourceControlIntegration.cpp*, and *SourceControlIntegration.h*. The current file is *UnLinker.cpp*. The code itself contains C++ code with various preprocessor directives, including `#ifndef SCC_NXN_75`, `#ifndef CONSOLE`, and `#include "..\\..\\UnrealEd\\Inc\\SourceControlIntegration.h"`. There are also several `/* TODO: ... */` comments throughout the code. The right side of the interface features a properties panel titled "UnLinker" with numerous properties listed, such as `IsAb`, `IsInj`, `IsMa`, `IsSel`, `IsTer`, and `IsVal`. The bottom status bar indicates "Ln 7".

Figure 4 - *UnLinker1*

- call GetLatest (QA builds 108143 – 115119)

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
    extern FSourceControlIntegration* SCCFirst;
    static DWORD linkCount = 0;
    debugf(TEXT("Advanced SCC - ULinkerLoad::ULinkerLoad: %i, %s"), linkCount++, InFilename);
    if (SCCFirst)
        SCCFirst->Get(InFilename);
#endif // CONSOLE
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h
UnLinker.cpp
d:\Ue3\Development\Src\Core\Src\UnLinker.cpp
ULinkerLoad
Property
ULink
C++ (Name
File
FullN
IsAb:
IsInj:
IsMa:
IsSel:
IsTer:
IsVal:
```

```
ULinkerLoad::ULinkerLoad( UObject* InParent, const TCHAR* InFilename, DWORD InLoadFlags )
: ULinker( InParent, InFilename )
, LoadFlags( InLoadFlags )
, bHaveImportsBeenVerified( FALSE )
, bHasBeenFullyLoaded( FALSE )
{
    check(!HasAnyFlags(RF_ClassDefaultObject));
    UPackage* LinkerRootPackage = Cast<UPackage>(LinkerRoot);

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
    extern FSourceControlIntegration* SCCFirst;
    static DWORD linkCount = 0;
    debugf(TEXT("Advanced SCC - ULinkerLoad::ULinkerLoad: %i, %s"), linkCount++, InFilename);
    if (SCCFirst)
        SCCFirst->Get(InFilename);
#endif // CONSOLE
#endif // SCC_NXN_75

    if( (LoadFlags & LOAD_SeekFree) && ParseParam(appCmdLine(), TEXT("SEEKFREE")) )
    {
        // Use the seek free loading archive to load this package. This relies on the package to be saved in a
        // special way to actually be seek free but it's save to use it for all loading.
        Loader = new FArchiveAsync( InFilename );
        // An error signifies that the package couldn't be opened.
        if( Loader->IsError() )
        {
            delete Loader;
            appThrowf( *FString::Printf(*FLocalizeError(TEXT("OpenFailed")), TEXT("Core")), InFilename, *GFileManager->Get
        }
    }
    else
    {
        FArchive* FileReader = GFileManager->CreateFileReader( InFilename, 0, GError );
        if( !FileReader )
        {
            appThrowf( *FString::Printf(*FLocalizeError(TEXT("OpenFailed")), TEXT("Core")), InFilename, *GFileManager->Get
        }

        if( LoadFlags & LOAD_MemoryReader )
        {
            // Serialize data from memory instead of from disk.
            check(FileReader);
            UINT BufferSize = FileReader->TotalSize();
            void* Buffer = appMalloc( BufferSize );
            FileReader->Serialize( Buffer, BufferSize );
        }
    }
}

// TODO: 10tacle, added by Ernest SCC
```

Figure 4.1 - UnLinker2

- call GetLatest (QA build 121834)

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
    extern FSourceControlIntegration* SCCFirst;
    static DWORD linkCount = 0;
    debugf(TEXT("Advanced SCC - ULinkerLoad::ULinkerLoad: %i, %s"), linkCount++, *Filename);
    if (SCCFirst)
        SCCFirst->Get(*Filename);
#endif // CONSOLE
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | GenericBrowser.h | ResourceIDs.h | Launch.cpp | VersionCheck.cpp | VersionCheck.h | UnThreadingWindows.h | Solution
UnLinker.cpp
d:\Ue3\Development\Src\Core\Src\UnLinker.cpp
CreateLoader


    /**
     * Creates loader used to serialize content.
     */
    UBOOL ULinkerLoad::CreateLoader()
    {
        if( !Loader )
        {
            // TODO: 10tacle, added by Ernest SCC
            #ifdef SCC_NXN_75
            #ifndef CONSOLE
                extern FSourceControlIntegration* SCCFirst;
                static DWORD linkCount = 0;
                debugf(TEXT("Advanced SCC - ULinkerLoad::ULinkerLoad: %i, %s"), linkCount++, *Filename);
                if (SCCFirst)
                    SCCFirst->Get(*Filename);
            #endif // CONSOLE
            #endif // SCC_NXN_75

            if( LoadFlags & LOAD_SeekFree )
            {
                // Use the seek free loading archive to load this package. This relies on the package to be saved in a
                // special way to actually be seek free but it's save to use it for all loading.
                Loader = new FArchiveAsync( *Filename );
                // An error signifies that the package couldn't be opened.
                if( Loader->IsError() )
                {
                    delete Loader;
                    appThrowf( *FString::Printf(*FString::LocalizeError(TEXT("OpenFailed")), TEXT("Core")), *Filename, *GFileManager->
                }
            }
            else
            {
                // Create file reader used for serialization.
                FileReader = GFileManager->CreateFileReader( *Filename, 0, GError );
                if( !FileReader )
                {
                    appThrowf( *FString::Printf(*FString::LocalizeError(TEXT("OpenFailed")), TEXT("Core")), *Filename, *GFileManager->
                }

                if( LoadFlags & LOAD_MemoryReader )
                {
                    // Serialize data from memory instead of from disk.
                    check(FileReader);
                    UINT BufferSize = FileReader->TotalSize();
                    void* Buffer = appMalloc(BufferSize);
                    FileReader->Serialize( Buffer, BufferSize );
                    Loader = new FBufferedReader( Buffer, BufferSize, TRUE, TRUE );
                }
            }
        }
    }


```

Figure 4.2 – UnLinker2 (QA 121834)

4) Update *UnObj.cpp* (Core project):

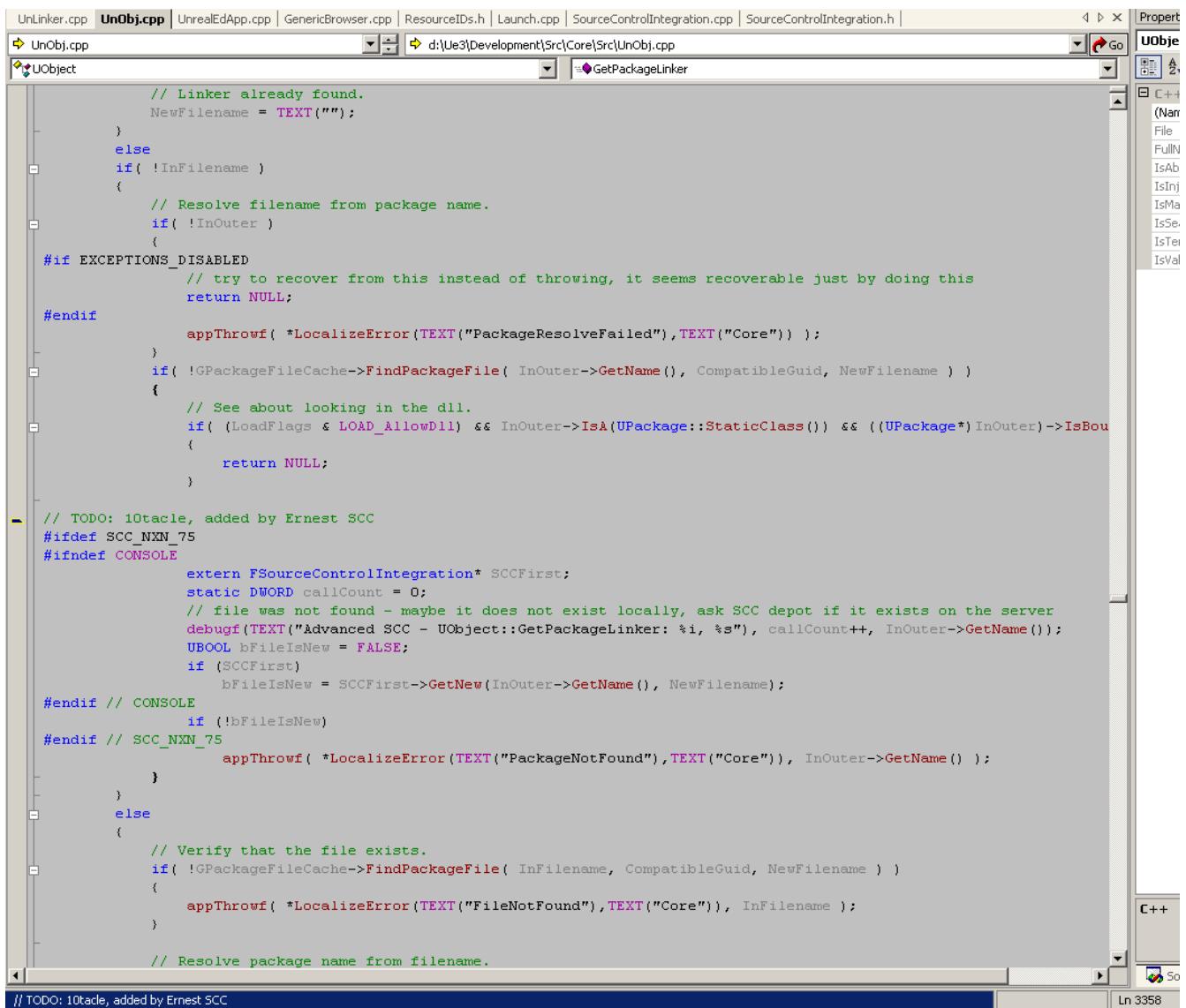
- we're going to use SCC in this file, include it

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
#include "..\\..\\UnrealEd\\Inc\\SourceControlIntegration.h"
#endif // CONSOLE
#endif // SCC_NXN_75
```

*Figure 5 – UnObj1*

- call GetLatest in case of new file

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#ifndef CONSOLE
extern FSourceControlIntegration* SCCFirst;
static DWORD callCount = 0;
// file was not found - maybe it does not exist locally, ask SCC depot if it exists on the server
debugf(TEXT("Advanced SCC - UObject::GetPackageLinker: %i, %s"), callCount++, InOuter->GetName());
UBOOL bFileIsNew = FALSE;
if (SCCFirst)
    bFileIsNew = SCCFirst->GetNew(InOuter->GetName(), NewFilename);
#endif // CONSOLE
    if (!bFileIsNew)
#endif // SCC_NXN_75
```

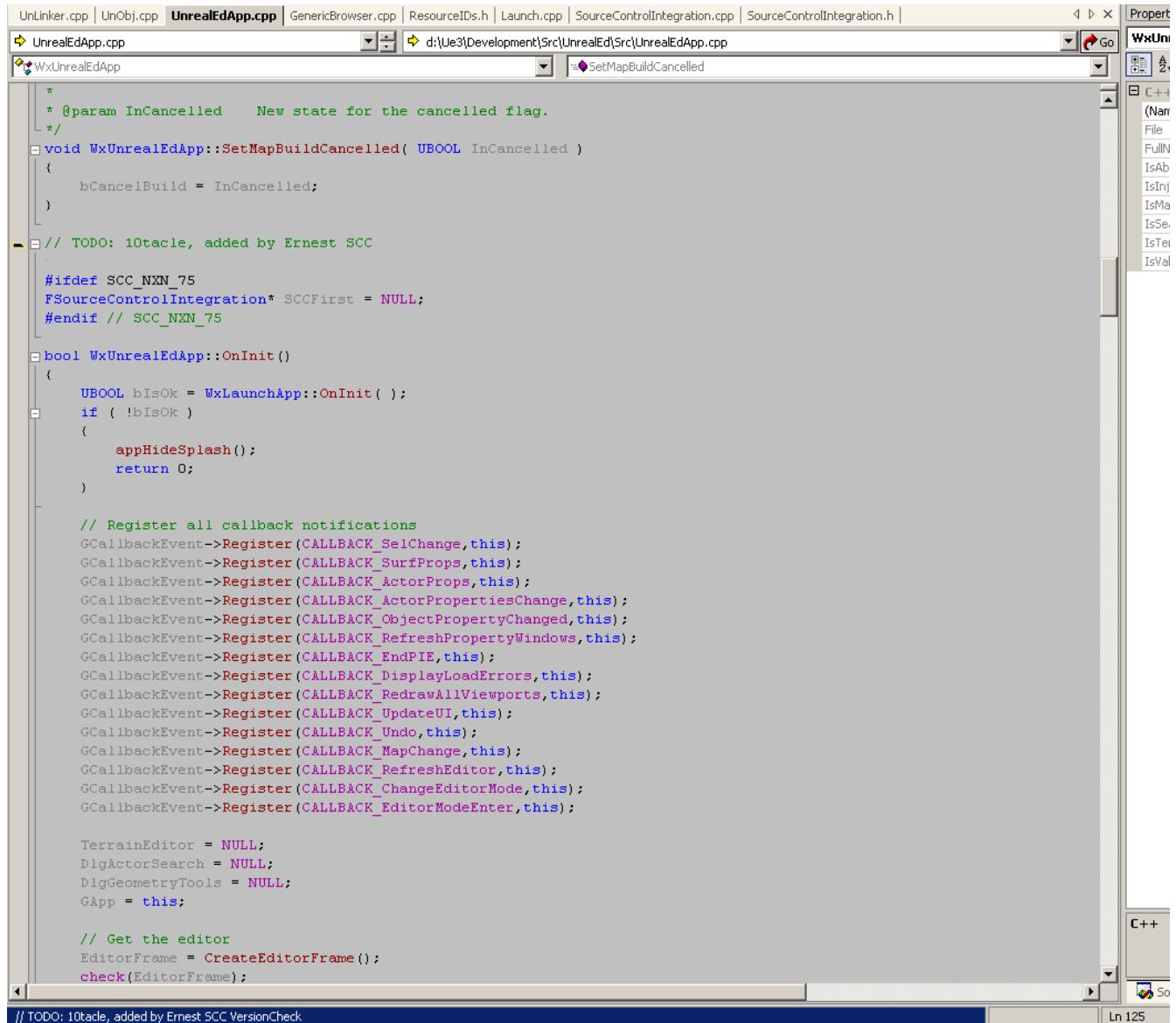


*Figure 6 – UnObj2*

5) Update *UnrealEdApp.cpp* (UnrealEd project):

- define pointer to SCC

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
FSourceControlIntegration* SCCFirst = NULL;
#endif // SCC_NXN_75
```



```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h |
UnrealEdApp.cpp | d:\Ue3\Development\Src\|UnrealEd\Src\|UnrealEdApp.cpp | Go
WxUnrealEdApp | SetMapBuildCancelled
Properties
WxUnrealEdApp
C++ (Narr File FullN IsAb IsInj IsMa IsSe IsTe IsVal
* @param InCancelled New state for the cancelled flag.
*/
void WxUnrealEdApp::SetMapBuildCancelled( UBOOL InCancelled )
{
    bCancelBuild = InCancelled;
}

// TODO: 10tacle, added by Ernest SCC

#ifndef SCC_NXN_75
FSourceControlIntegration* SCCFirst = NULL;
#endif // SCC_NXN_75

bool WxUnrealEdApp::OnInit()
{
    UBOOL bisOk = WxLaunchApp::OnInit();
    if ( !bisOk )
    {
        appHideSplash();
        return 0;
    }

    // Register all callback notifications
    GCallbackEvent->Register(CALLBACK_SelChange,this);
    GCallbackEvent->Register(CALLBACK_SurfProps,this);
    GCallbackEvent->Register(CALLBACK_ActorProps,this);
    GCallbackEvent->Register(CALLBACK_ActorPropertiesChange,this);
    GCallbackEvent->Register(CALLBACK_ObjectPropertyChanged,this);
    GCallbackEvent->Register(CALLBACK_RefreshPropertyWindows,this);
    GCallbackEvent->Register(CALLBACK_EndPIE,this);
    GCallbackEvent->Register(CALLBACK_DisplayLoadErrors,this);
    GCallbackEvent->Register(CALLBACK_RedrawAllViewports,this);
    GCallbackEvent->Register(CALLBACK_UpdateUI,this);
    GCallbackEvent->Register(CALLBACK_Undo,this);
    GCallbackEvent->Register(CALLBACK_MapChange,this);
    GCallbackEvent->Register(CALLBACK_RefreshEditor,this);
    GCallbackEvent->Register(CALLBACK_ChangeEditorMode,this);
    GCallbackEvent->Register(CALLBACK_EditorModeEnter,this);

    TerrainEditor = NULL;
    DlgActorSearch = NULL;
    DlgGeometryTools = NULL;
    Gapp = this;

    // Get the editor
    EditorFrame = CreateEditorFrame();
    check(EditorFrame);

// TODO: 10tacle, added by Ernest SCC VersionCheck
```

Figure 7 – UnrealEdAppI

- init source control

```
// TODO: 10tacle, changed by Ernest SCC
#ifndef SCC_NXN_75
    if(SCCFirst)
        SCC = SCCFirst;
    else
    {
        SCC = new FSourceControlIntegration;
        SCCFirst = SCC;
    }
#else
    SCC = new FSourceControlIntegration;
#endif // SCC_NXN_75
```

The screenshot shows the UnrealEdApp.cpp file open in a code editor. The code implements a conditional block to initialize a source control object based on the SCC\_NXN\_75 macro. If defined, it uses SCCFirst; otherwise, it creates a new FSourceControlIntegration object and sets SCCFirst to it. The code is annotated with several TODO comments and includes declarations for various UI components like WxButtonBar and WxDlgActorSearch.

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h | d:\Ue3\Development\Src\UnrealEd\Src\UnrealEdApp.cpp
UnrealEdApp.cpp
WxUnrealEdApp
OnInit

// Init source control

// TODO: 10tacle, changed by Ernest SCC
#ifndef SCC_NXN_75
    if (SCCFirst)
        SCC = SCCFirst;
    else
    {
        SCC = new FSourceControlIntegration;
        SCCFirst = SCC;
    }
#else
    SCC = new FSourceControlIntegration;
#endif // SCC_NXN_75

// Init the editor tools.
GTexAlignTools::Init();

EditorFrame->ButtonBar = new WxButtonBar;
EditorFrame->ButtonBar->Create( (wxWindow*)EditorFrame, -1 );
EditorFrame->ButtonBar->Show();

DlgActorSearch = new WxDlgActorSearch( EditorFrame );
DlgActorSearch->Show(0);

DlgMapCheck = new WxDlgMapCheck( EditorFrame );
DlgMapCheck->Show(0);

DlgActorFactory = new WxDlgActorFactory();
DlgActorFactory->Show(0);

DlgBuildProgress = new WxDlgBuildProgress( EditorFrame );
DlgBuildProgress->Show(0);

//joegtemp -- Set the handle to use for GWarn->MapCheck_Xxx() calls
GWarn->winEditorFrame = (DWORD)EditorFrame;
GWarn->hWndEditorFrame = (DWORD)EditorFrame->GetHandle();

DlgLoadErrors = new WxDlgLoadErrors( EditorFrame );

DlgSurfProp = new WxDlgSurfaceProperties();
//DlgSurfProp->Show( 0 );

ObjectPropertyWindow = NULL;
CurrentPropertyWindow = NULL;

EditorFrame->OptionProxyInit();

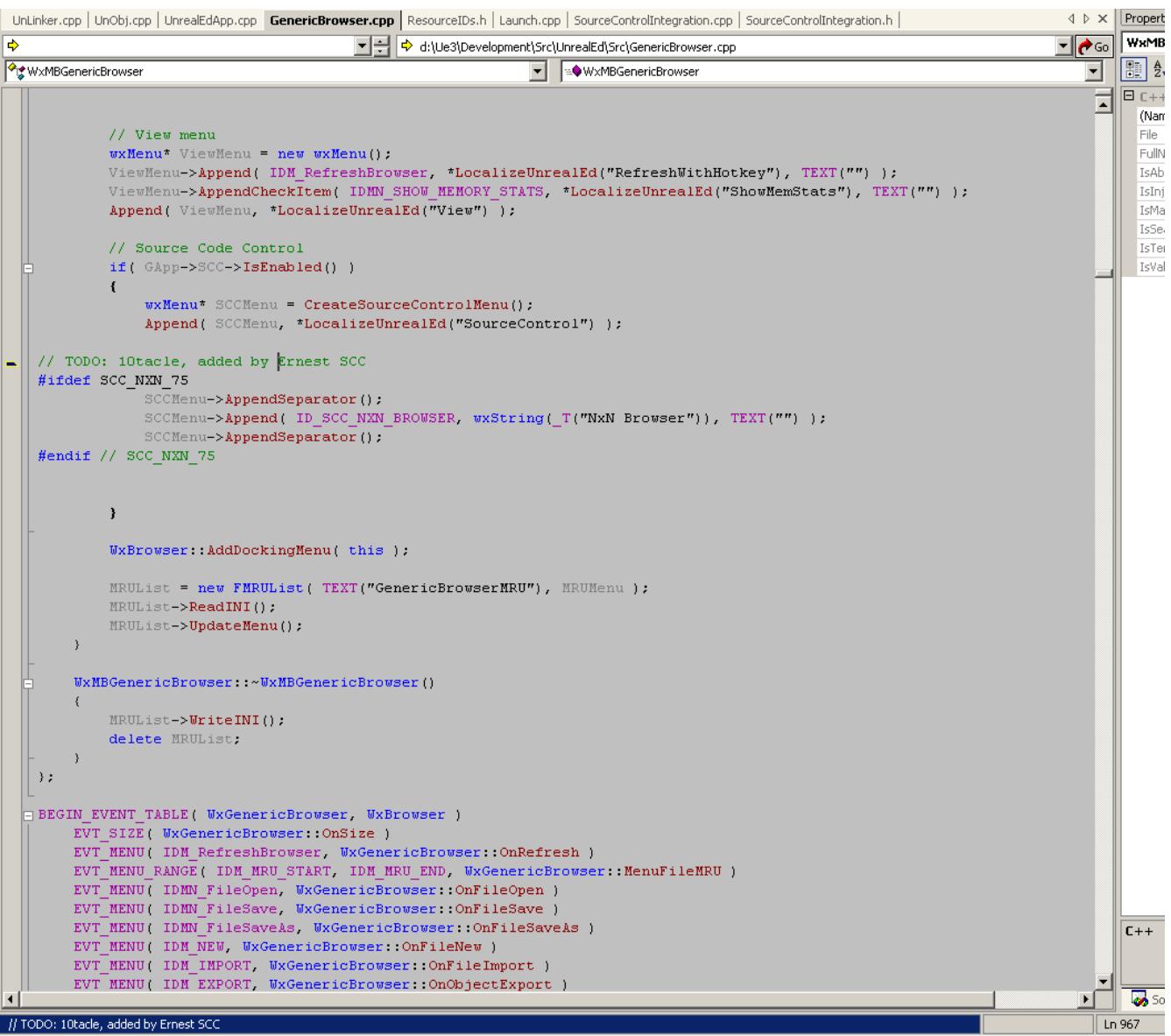
// TODO: 10tacle, changed by Ernest SCC
```

Figure 8 – UnrealEdApp2

6) Update *GenericBrowser.cpp*, *GenericBrowser.h* and *ResourceIDs.h* (UnrealEd project):

- add menu item for NxN Browser

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    SCCMenu->AppendSeparator();
    SCCMenu->Append( ID_SCC_NXN_BROWSER, wxString(_T("NxN Browser")), TEXT("") );
    SCCMenu->AppendSeparator();
#endif // SCC_NXN_75
```



```
// View menu
wxMenu* ViewMenu = new wxMenu();
ViewMenu->Append( IDM_RefreshBrowser, *LocalizeUnrealEd("RefreshWithHotkey"), TEXT("") );
ViewMenu->AppendCheckItem( IDMN_SHOW_MEMORY_STATS, *LocalizeUnrealEd("ShowMemStats"), TEXT("") );
Append( ViewMenu, *LocalizeUnrealEd("View") );

// Source Code Control
if( GApp->SCC->IsEnabled() )
{
    wxMenu* SCCMenu = CreateSourceControlMenu();
    Append( SCCMenu, *LocalizeUnrealEd("SourceControl") );
}

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    SCCMenu->AppendSeparator();
    SCCMenu->Append( ID_SCC_NXN_BROWSER, wxString(_T("NxN Browser")), TEXT("") );
    SCCMenu->AppendSeparator();
#endif // SCC_NXN_75

}

WxBrowser::AddDockingMenu( this );

MRUList = new FMRLIST( TEXT("GenericBrowserMRU"), MRUMenu );
MRUList->ReadINI();
MRUList->UpdateMenu();

WxMBGenericBrowser::~WxMBGenericBrowser()
{
    MRUList->WriteINI();
    delete MRUList;
}

BEGIN_EVENT_TABLE( WxGenericBrowser, WxBrowser )
    EVT_SIZE( WxGenericBrowser::OnSize )
    EVT_MENU( IDM_RefreshBrowser, WxGenericBrowser::OnRefresh )
    EVT_MENU_RANGE( IDM_MRU_START, IDM_MRU_END, WxGenericBrowser::MenuFileMRU )
    EVT_MENU( IDMN_FileOpen, WxGenericBrowser::OnFileOpen )
    EVT_MENU( IDMN_FileSave, WxGenericBrowser::OnFileSave )
    EVT_MENU( IDMN_FileSaveAs, WxGenericBrowser::OnFileSaveAs )
    EVT_MENU( IDM_NEW, WxGenericBrowser::OnFileNew )
    EVT_MENU( IDM_IMPORT, WxGenericBrowser::OnFileImport )
    EVT_MENU( IDM_EXPORT, WxGenericBrowser::OnObjectExport )
}

// TODO: 10tacle, added by Ernest SCC
```

Figure 9 – GenericBrowser1

- add event to event table for the item

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    EVT_MENU( ID_SCC_NXN_BROWSER, WxGenericBrowser::OnSCCNXBrower )
#endif // SCC_NXN_75
```

```
BEGIN_EVENT_TABLE( WxGenericBrowser, WxBrowser )
    EVT_SIZE( WxGenericBrowser::OnSize )
    EVT_MENU( IDM_RefreshBrowser, WxGenericBrowser::OnRefresh )
    EVT_MENU_RANGE( IDM_HRU_START, IDM_HRU_END, WxGenericBrowser::MenuFileMRU )
    EVT_MENU( IDMN_FileOpen, WxGenericBrowser::OnFileOpen )
    EVT_MENU( IDMN_FileSave, WxGenericBrowser::OnFileSave )
    EVT_MENU( IDMN_FileSaveAs, WxGenericBrowser::OnFileSaveAs )
    EVT_MENU( IDM_NEW, WxGenericBrowser::OnFileNew )
    EVT_MENU( IDM_IMPORT, WxGenericBrowser::OnFileImport )
    EVT_MENU( IDM_EXPORT, WxGenericBrowser::OnObjectExport )
    EVT_MENU_RANGE( ID_NEWOBJ_START, ID_NEWOBJ_END, WxGenericBrowser::OnContextObjectNew )
    EVT_MENU( ID_SCC_REFRESH, WxGenericBrowser::OnSCCRefresh )
    EVT_MENU( ID_SCC_HISTORY, WxGenericBrowser::OnSCCHistory )
    EVT_MENU( ID_SCC_MOVE_TO_TRASH, WxGenericBrowser::OnSCCMoveToTrash )
    EVT_MENU( ID_SCC_ADD, WxGenericBrowser::OnSCCAAdd )
    EVT_MENU( ID_SCC_CHECK_OUT, WxGenericBrowser::OnSCCCheckOut )
    EVT_MENU( ID_SCC_CHECK_IN, WxGenericBrowser::OnSCCCheckIn )
    EVT_MENU( ID_SCC_REVERT, WxGenericBrowser::OnSCCRevert )
    EVT_UPDATE_UI( ID_SCC_CHECK_OUT, WxGenericBrowser::UI_SCCCheckOut )
    EVT_UPDATE_UI( ID_SCC_CHECK_IN, WxGenericBrowser::UI_SCCCheckIn )
    EVT_UPDATE_UI( ID_SCC_REVERT, WxGenericBrowser::UI_SCCRevert )
    EVT_UPDATE_UI( ID_SCC_ADD, WxGenericBrowser::UI_SCCAdd )

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    EVT_MENU( ID_SCC_NXN_BROWSER, WxGenericBrowser::OnSCCNXBrower )
#endif // SCC_NXN_75

EVT_MENU( IDMN_ObjectContext_CopyReference, WxGenericBrowser::OnCopyReference )
EVT_MENU( IDMN_ObjectContext_Properties, WxGenericBrowser::OnObjectProperties )
EVT_MENU( IDMN_ObjectContext_Duplicate, WxGenericBrowser::OnObjectDuplicate )
EVT_MENU( IDMN_ObjectContext_Rename, WxGenericBrowser::OnObjectRename )
EVT_MENU( IDMN_ObjectContext_Delete, WxGenericBrowser::OnObjectDelete )
EVT_MENU( IDMN_ObjectContext_ShowRefs, WxGenericBrowser::OnObjectShowReferencers )
EVT_MENU( IDMN_ObjectContext_Export, WxGenericBrowser::OnObjectExport )
EVT_MENU( IDMN_ObjectContext_Editor, WxGenericBrowser::OnObjectEditor )
EVT_MENU_RANGE( IDMN_ObjectContext_Custom_START, IDMN_ObjectContext_Custom_END, WxGenericBrowser::OnObjectCustomCom

EVT_MENU( IDMN_SHOW_MEMORY_STATS, WxGenericBrowser::OnShowMemStats )
EVT_UPDATE_UI( IDMN_SHOW_MEMORY_STATS, WxGenericBrowser::UI_ShowMemStats )

EVT_MENU( IDMN_ObjectContext_DuplicateWithRefs, WxGenericBrowser::OnObjectDuplicateWithRefs )
EVT_MENU( IDMN_ObjectContext_RenameWithRefs, WxGenericBrowser::OnObjectRenameWithRefs )
EVT_MENU( IDMN_ObjectContext_DeleteWithRefs, WxGenericBrowser::OnObjectDeleteWithRefs )
EVT_MENU( IDMN_ObjectContext_ShowRefObjs, WxGenericBrowser::OnObjectShowReferencedObjs )
```

Figure 10 – GenericBrowser2

- definition of the item's processing function

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
extern FSourceControlIntegration* SCCFirst;
void WxGenericBrowser::OnSCCNXNBrowser( wxCommandEvent& In )
{
    if(SCCFirst)
        SCCFirst->RunScc(NULL);
}
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h | D X
GenericBrowser.cpp
d:\Ue3\Development\Src\UnrealEd\Src\GenericBrowser.cpp
WxGenericBrowser
UI_SCCRevert

void WxGenericBrowser::UI_SCCRevert( wxUpdateUIEvent& In )
{
    TArray<UPackage*> Packages;
    LeftContainer->GetSelectedTopLevelPackages( &Packages );

    UBOOL bEnable = FALSE;

    for( INT x = 0 ; x < Packages.Num() ; ++x )
    {
        if( Packages(x)->GetSCCState() == SCC_CheckedOut )
        {
            bEnable = TRUE;
            break;
        }
    }

    In.Enable( bEnable == TRUE );
}

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
extern FSourceControlIntegration* SCCFirst;
void WxGenericBrowser::OnSCCNXNBrowser( wxCommandEvent& In )
{
    if (SCCFirst)
        SCCFirst->RunScc(NULL);
}
#endif // SCC_NXN_75

// WxGenericBrowser static member functions
/***
 * Returns an FSelection for the set of objects selected in the generic browser.
 */
FSelection* WxGenericBrowser::GetSelection()
{
    return GetGSElection();
}

// TODO: 10tacle, added by Ernest SCC
```

Figure 11 – GenericBrowser3

- declaration of the item's processing function

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    void OnSCCNXNBrowser( wxCommandEvent& In );
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h | GenericBrowser.h | < Go
WxGenericBrowser
WxGenericBrowser | UI_SCCRevert
/*
void OnRefresh( wxCommandEvent& In );

/**
 * Event handler for when the user selects an item from the Most Recently Used menu.
 */
void MenuFileMRU( wxCommandEvent& In );

void OnFileOpen( wxCommandEvent& In );
void OnFileSave( wxCommandEvent& In );
void OnFileSaveAs( wxCommandEvent& In );
void OnFileNew( wxCommandEvent& In );
void OnFileImport( wxCommandEvent& In );
void OnSCCRefresh( wxCommandEvent& In );
void OnSCCHistory( wxCommandEvent& In );
void OnSCCMoveToTrash( wxCommandEvent& In );
void OnSCCAAdd( wxCommandEvent& In );
void OnSCCCheckOut( wxCommandEvent& In );
void OnSCCCheckIn( wxCommandEvent& In );
void OnSCCRevert( wxCommandEvent& In );
void UI_SCCAAdd( wxUpdateUIEvent& In );
void UI_SCCCheckOut( wxUpdateUIEvent& In );
void UI_SCCCheckIn( wxUpdateUIEvent& In );
void UI_SCCRevert( wxUpdateUIEvent& In );

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    void OnSCCNXNBrowser( wxCommandEvent& In );
#endif // SCC_NXN_75

void OnCopyReference(wxCommandEvent& In);
void OnObjectProperties(wxCommandEvent& In);
void OnObjectDuplicate(wxCommandEvent& In);
void OnObjectDelete(wxCommandEvent& In);
void OnObjectShowReferencers(wxCommandEvent& In);
void OnObjectRename(wxCommandEvent& In);
void OnObjectExport(wxCommandEvent& In);
void OnContextObjectNew(wxCommandEvent& In);
void OnObjectEditor(wxCommandEvent& In);
void OnObjectCustomCommand( wxCommandEvent& In );

void OnObjectDeleteRedirects( wxCommandEvent& In );
void OnObjectDuplicateWithRefs( wxCommandEvent& In );
```

Figure 12 – GenericBrowser4

- add ID for the event

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    ID_SCC_NXN_BROWSER,
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h |
d:\Ue3\Development\Src\UnrealEd\Inc\ResourceIDs.h
Properties

IDM_SME_LOD_3,
IDM_SME_IMPORTMESHLOD,
IDM_SME_REMOVELOD,
IDM_SME_GENERATELOD,
IDM_SME_GENERATEUVS,

IDCB_PACKAGE,
IDEC_GROUP,
IDPB_BROWSER,
IDEC_NAME,

ID_SCC_ADD,
ID_SCC_CHECK_OUT,
ID_SCC_CHECK_IN,
ID_SCC_REVERT,
ID_SCC_HISTORY,
ID_SCC_MOVE_TO_TRASH,

// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    ID_SCC_NXN_BROWSER,
#endif // SCC_NXN_75

ID_SCC_REFRESH,
IDM_GROUPBYCLASS,
IDM_PSYSREALTIME,

IDPB_REFRESH,
IDPB_GOTOACTOR,
IDPB_DELETEACTOR,
IDL_C_ERRORWARNING,
ID_LOG,

// Geometry mode dialog IDs.
IDCK_SHOW_NORMALS,
IDCK_WIDGET_ONLY,
```

Figure 13 – ResourceIDs

## 7) Update *Launch.cpp* (PCLaunch project):

- include SCC header and create an instance

```
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
#include "SourceControlIntegration.h"
#endif // SCC_NXN_75

...
// TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
if(GIsEditor&&!GIsUCCMake&&!GIsUCC)
{
    extern FSourceControlIntegration* SCCFirst;
    SCCFirst = new FSourceControlIntegration;
}
#endif // SCC_NXN_75
```

```
UnLinker.cpp | UnObj.cpp | UnrealEdApp.cpp | GenericBrowser.cpp | ResourceIDs.h | Launch.cpp | SourceControlIntegration.cpp | SourceControlIntegration.h |
Launch.cpp | d:\Ue3\Development\Src\Launch\Src\Launch.cpp | StaticShutdownAfterError
(Globals) | -> StaticShutdownAfterError

if( LaunchApp )
{
    LaunchApp->ShutdownAfterError();
}

// TODO: 10tacle, added by Ernest SCC

#ifndef SCC_NXN_75
#include "SourceControlIntegration.h"
#endif // SCC_NXN_75

/**
 * Static guarded main function. Rolled into own function so we can have error handling for debug/ release builds dependent
 * on whether a debugger is attached or not.
 */
static INT GuardedMain( const TCHAR* CmdLine, HINSTANCE hInInstance, HINSTANCE hPrevInstance, INT nCmdShow )
{
    // Set up minidump filename. We cannot do this directly inside main as appItoa returns an FString that requires
    // destruction and main uses SEH.
    appStrcpy( MiniDumpFilenameW, TEXT("unreal-v") );
    appStrcat( MiniDumpFilenameW, *appItoa( GEngineVersion ) );
    appStrcat( MiniDumpFilenameW, TEXT(".dmp") );
    strcpy( MiniDumpFilenameA, TCHAR_TO_ANSI( MiniDumpFilenameW ) );

    INT ErrorLevel = GEngineLoop.PreInit( CmdLine );
    GUseuxWindows = !(GIsUCC || ParseParam(appCmdLine(), TEXT("nowxwindows")));

    // TODO: 10tacle, added by Ernest SCC
#ifndef SCC_NXN_75
    if (GIsEditor&&!GIsUCCMake&&!GIsUCC)
    {
        ...
        ...
        ...
        extern FSourceControlIntegration* SCCFirst;
        SCCFirst = new FSourceControlIntegration;
    }
#endif // SCC_NXN_75

    if( GUseuxWindows && ErrorLevel == 0 && !GIsRequestingExit )
    {
        // use wxWidgets as a DLL
        // set the call back class here
    }
}

// TODO: 10tacle, added by Ernest SCC
```

Figure 14 – Launch

- 8) Add preprocessor definition SCC\_NXN\_75 to Core, UnrealEd and PCLaunch projects (Configuration Properties -> C++ -> Preprocessor). You can remove the definition later to use default SCC integration.
- 9) Rebuild your solution.

Alienbrain and the Alienbrain logo are trademarks of NXN Software GmbH. All other trademarks contained herein are the property of their respective owners.

Microsoft Visual Studio® is registered trademarks of Microsoft Corporation.

THIS SOFTWARE IS PROVIDED „AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,  
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE 10TACLE STUDIOS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

In case of problems, troubles, bugs please contact: [scc\\_support@10tacle.com](mailto:scc_support@10tacle.com)

Ernest Mikita and Filip Fischer, 10TACLE STUDIOS SLOVAKIA

