

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 fuctionality:

- 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:

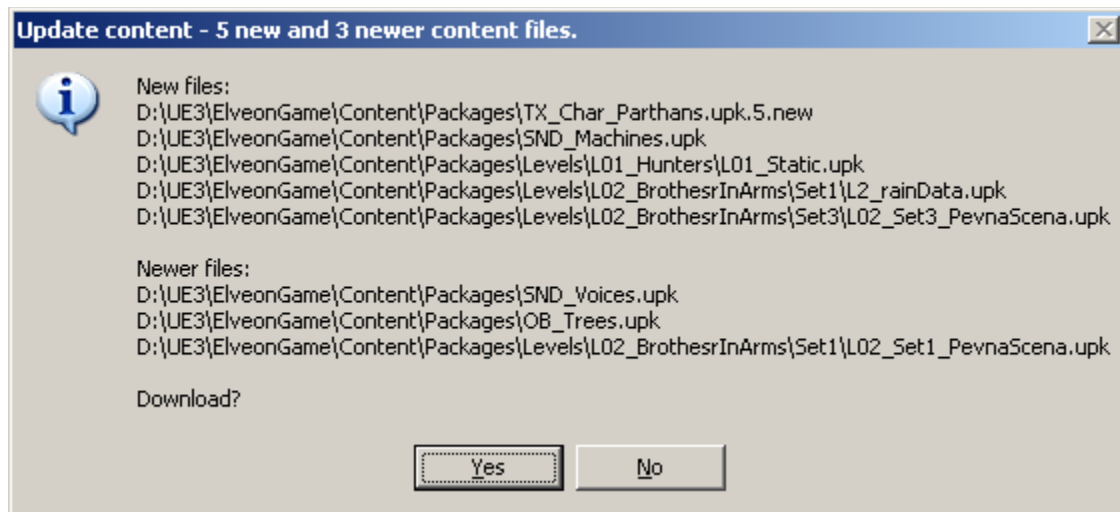


Figure 1 – Update content

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

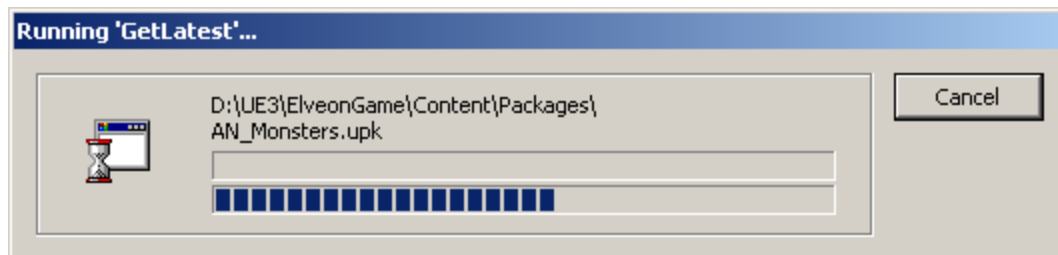


Figure 2 – downloading file

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/packages 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 packages 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:

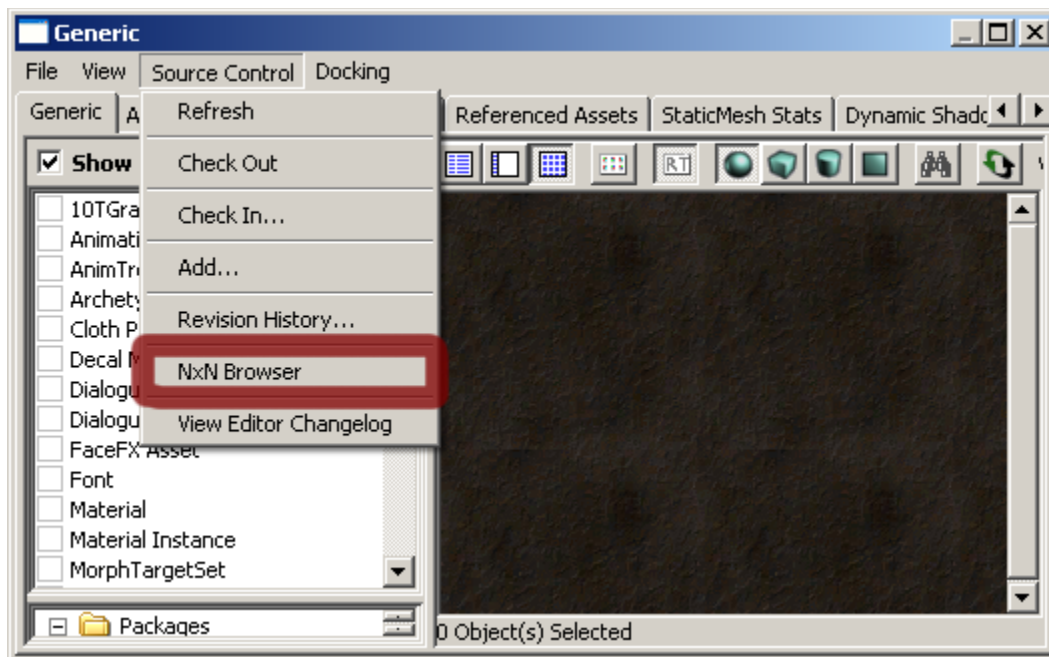


Figure 3 – Generic Browser

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 [SorceControl] in *DefaulEditor.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
#ifdef SCC_NXN_75
#ifndef CONSOLE
#include "..\..\UnrealEd\Inc\SourceControlIntegration.h"
#endif // CONSOLE
#endif // SCC_NXN_75
```

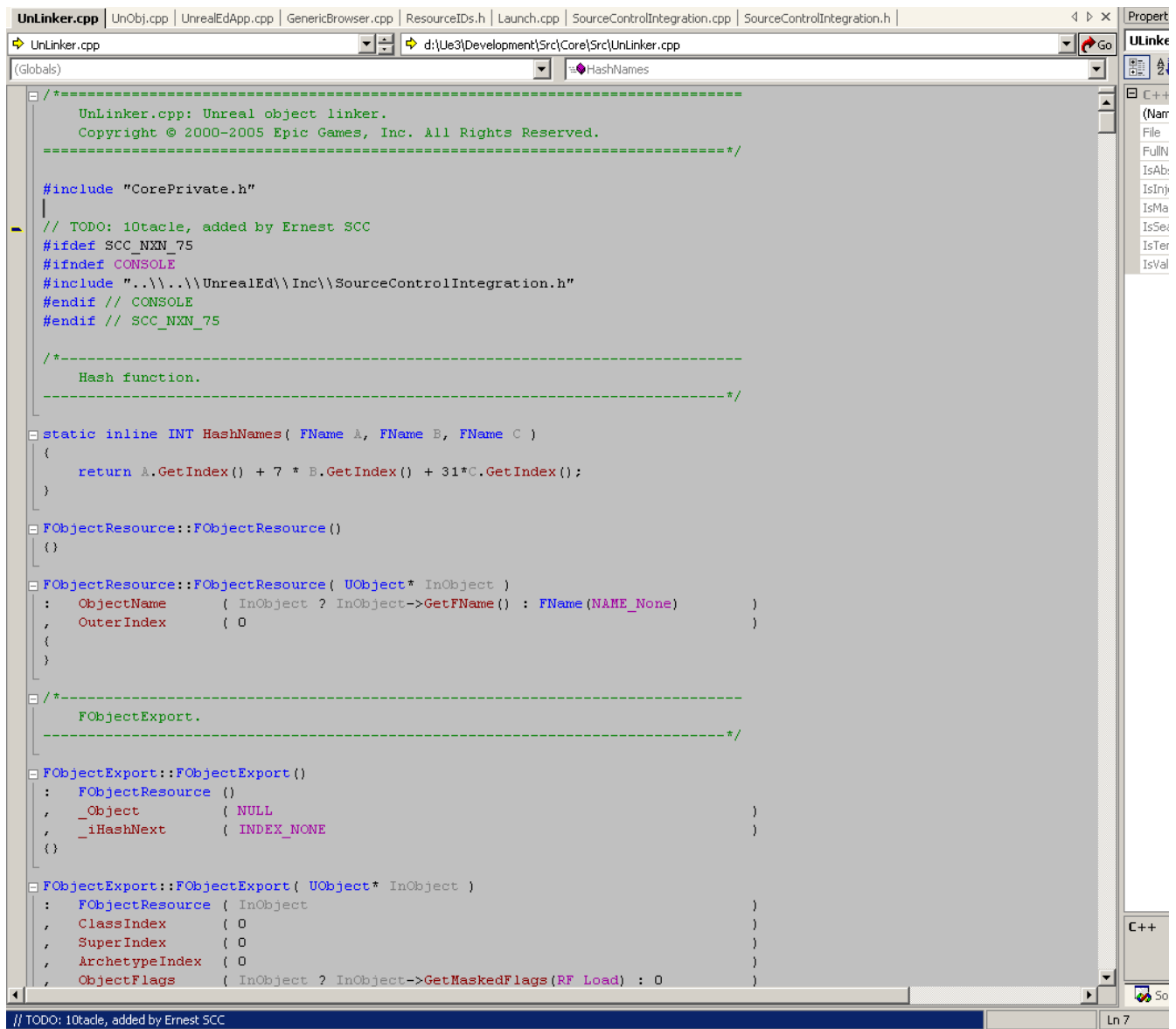


Figure 4 - *UnLinker1*

- call GetLatest (QA builds 108143 – 115119)

```
// 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++, InFilename);
    if (SCCFirst)
        SCCFirst->Get(InFilename);
#endif // CONSOLE
#endif // SCC_NXN_75
```

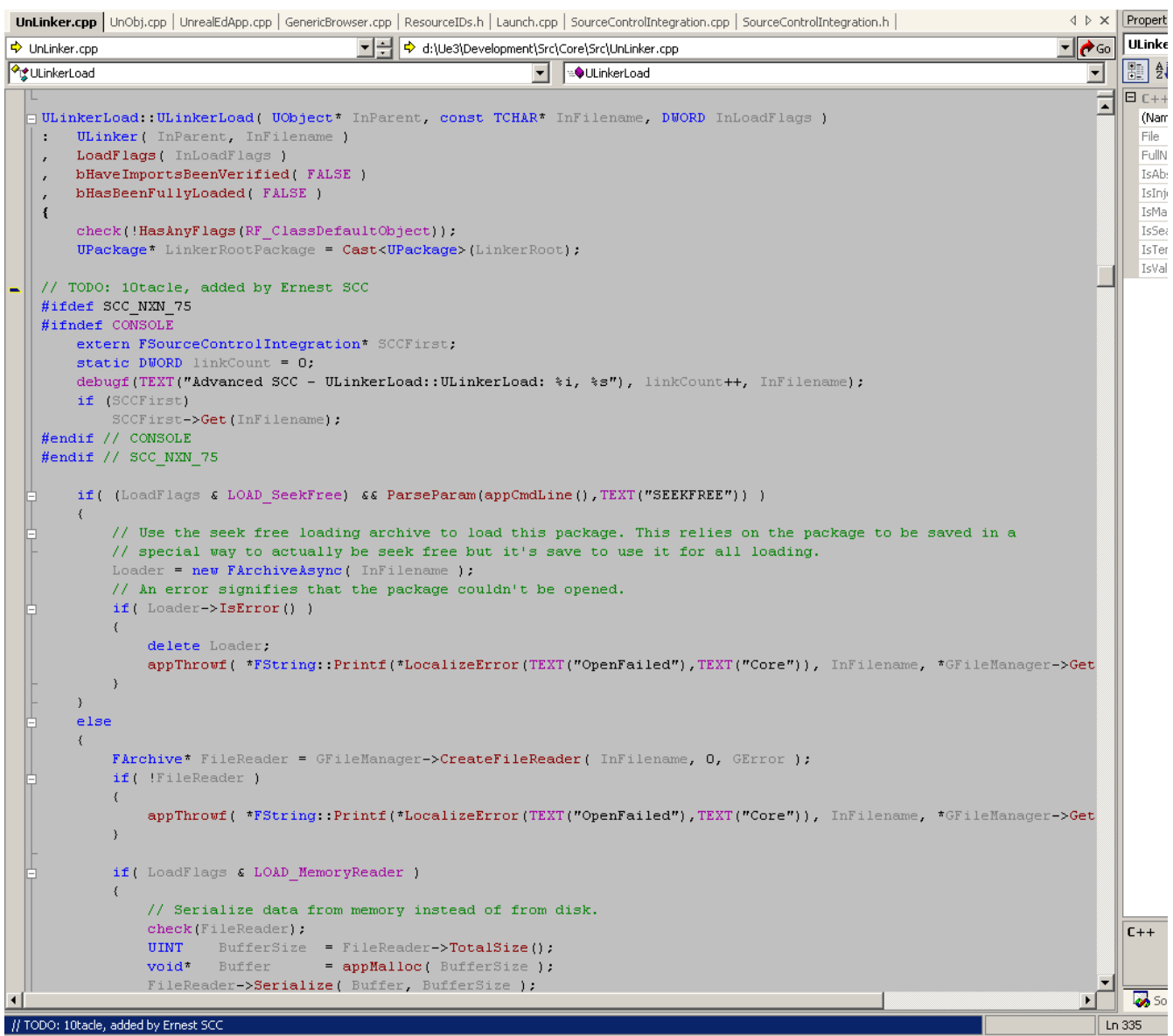


Figure 4.1 - ULinker2

- call GetLatest (QA build 121834)

// 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

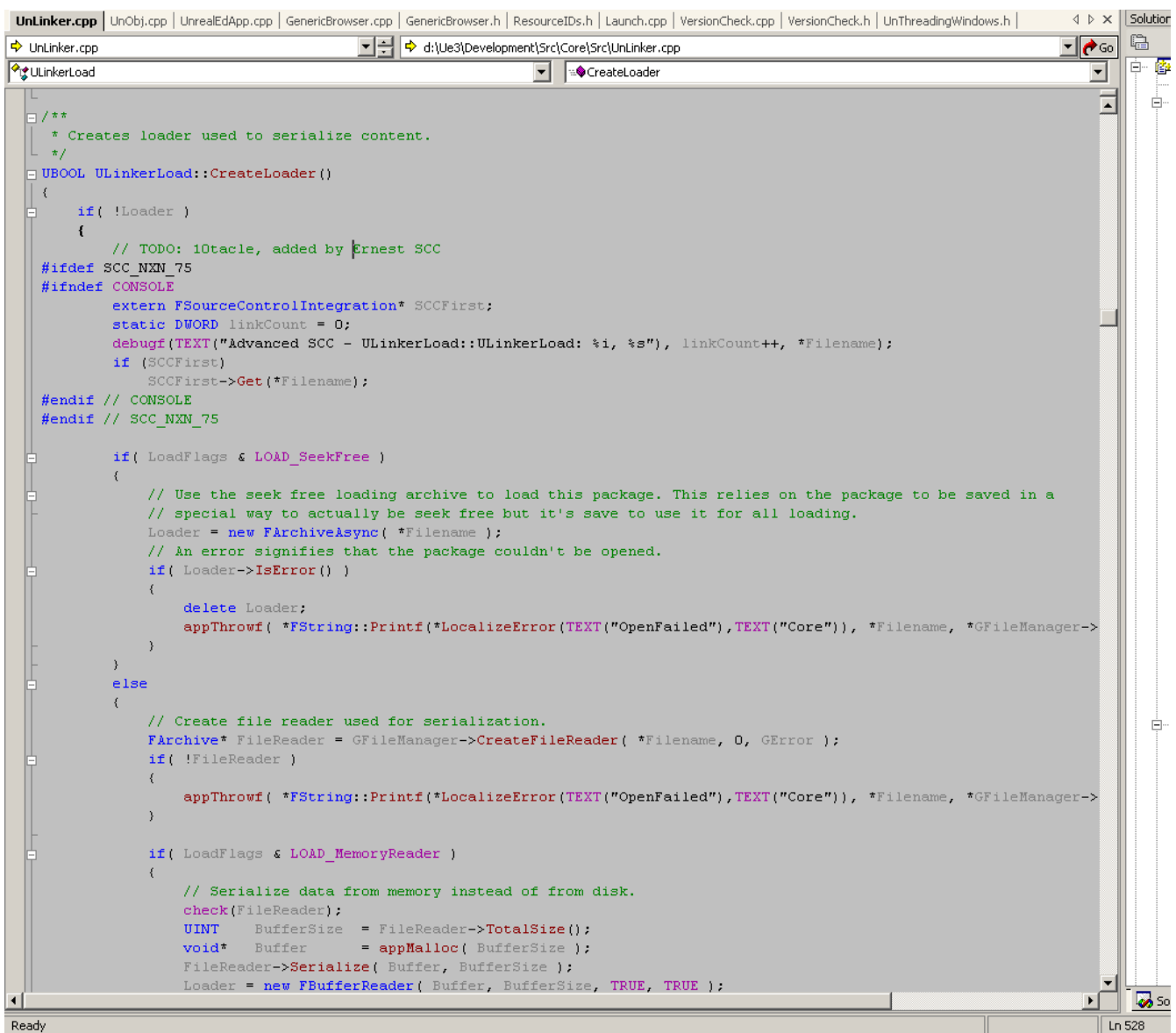


Figure 4.2 – UnLinker2 (QA 121834)

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

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

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

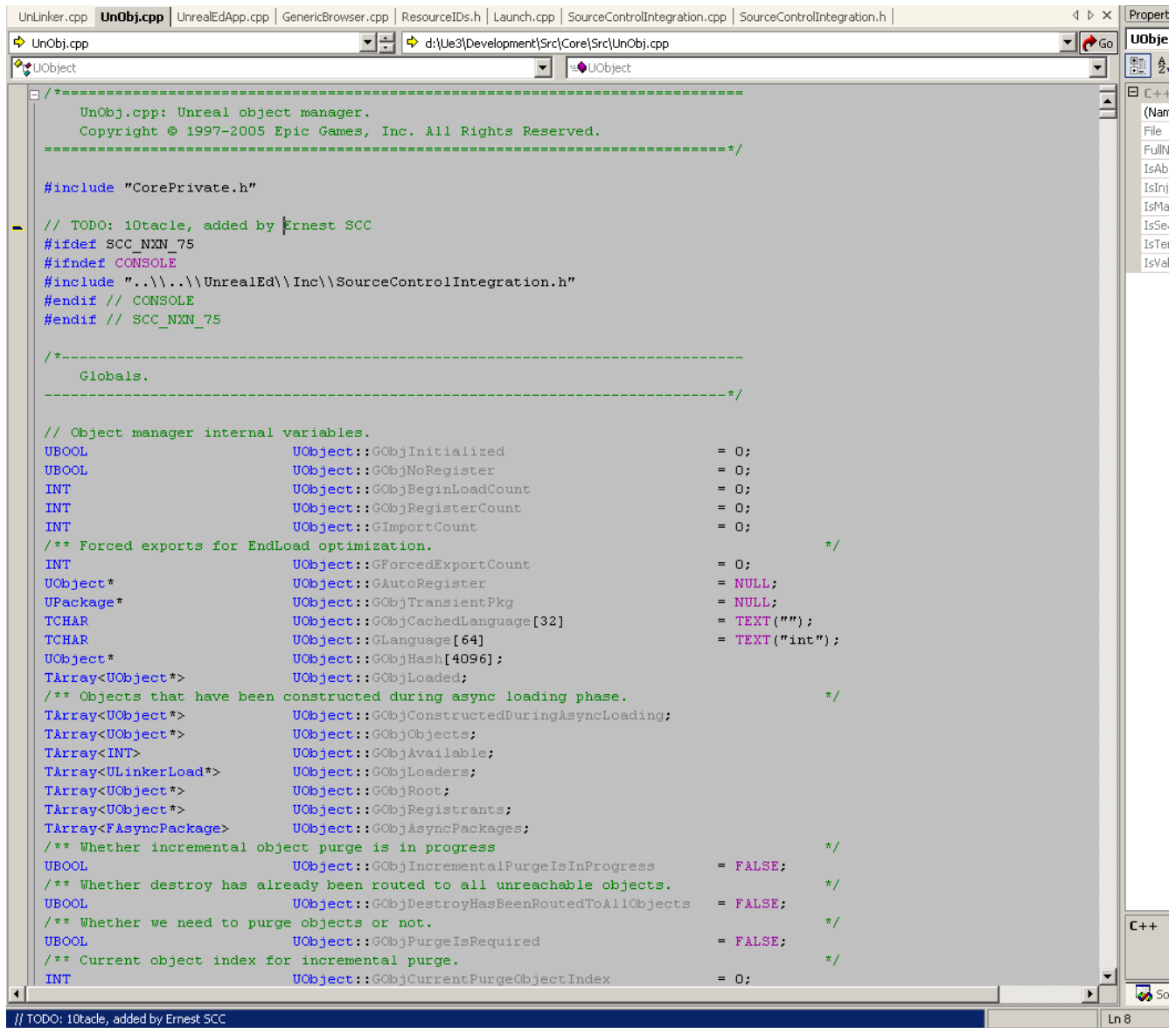


Figure 5 – *UnObj1*

- call GetLatest in case of new file

```
// TODO: 10tacle, added by Ernest SCC
#ifdef SCC_NXN_75
#ifdef 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
```

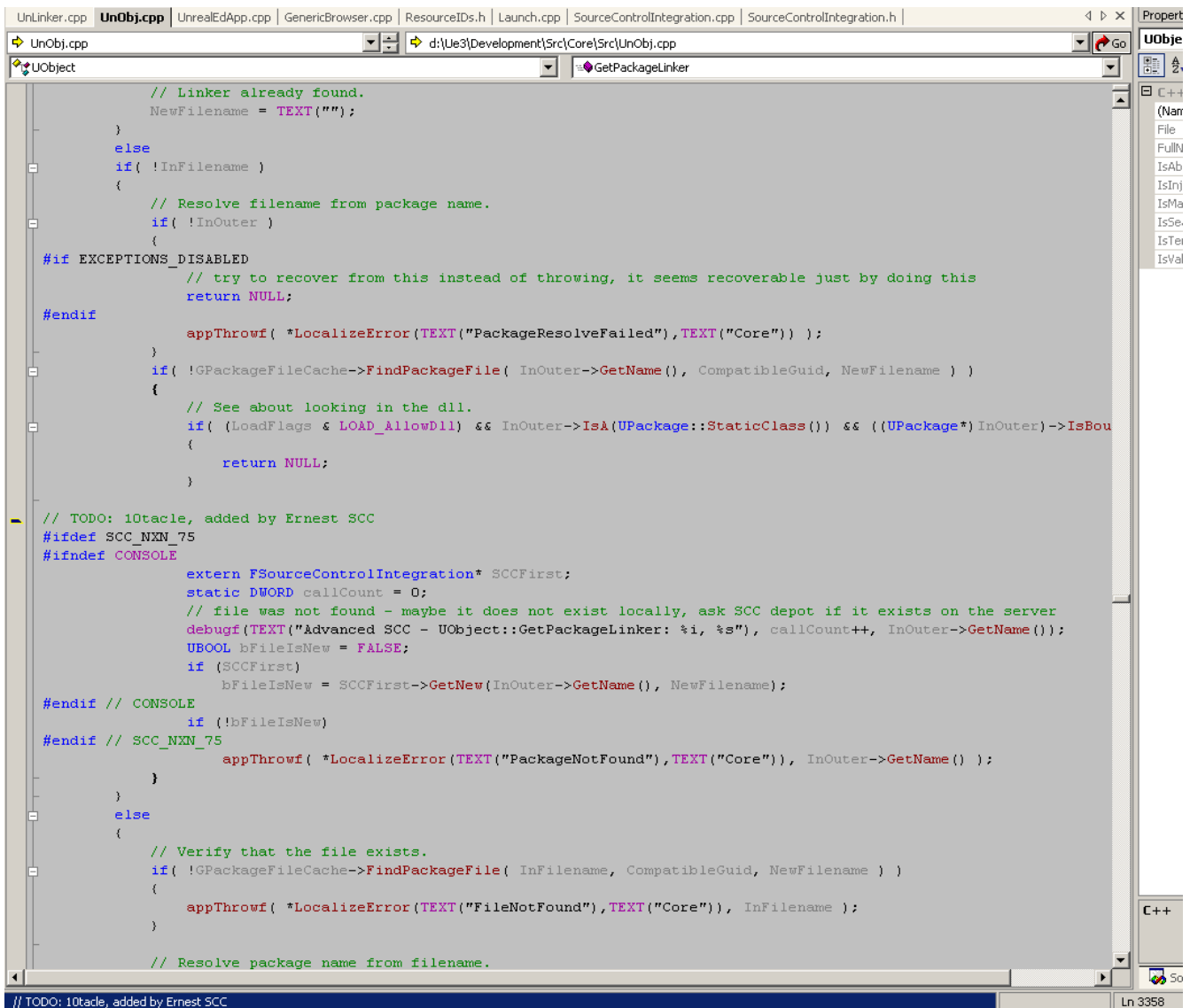


Figure 6 – UnObj2

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

- define pointer to SCC

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

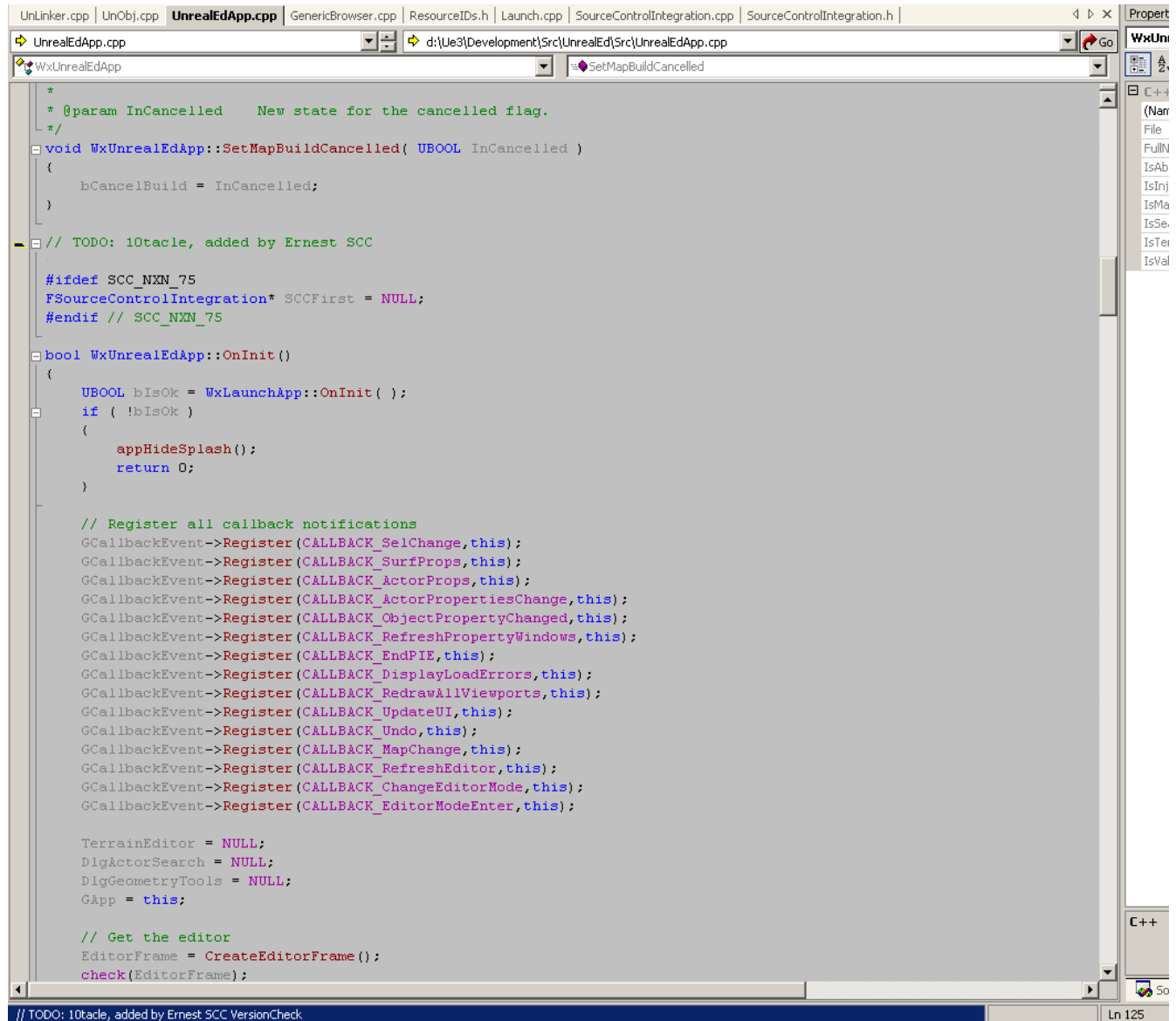


Figure 7 – *UnrealEdApp1*

- init source control

```

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

```

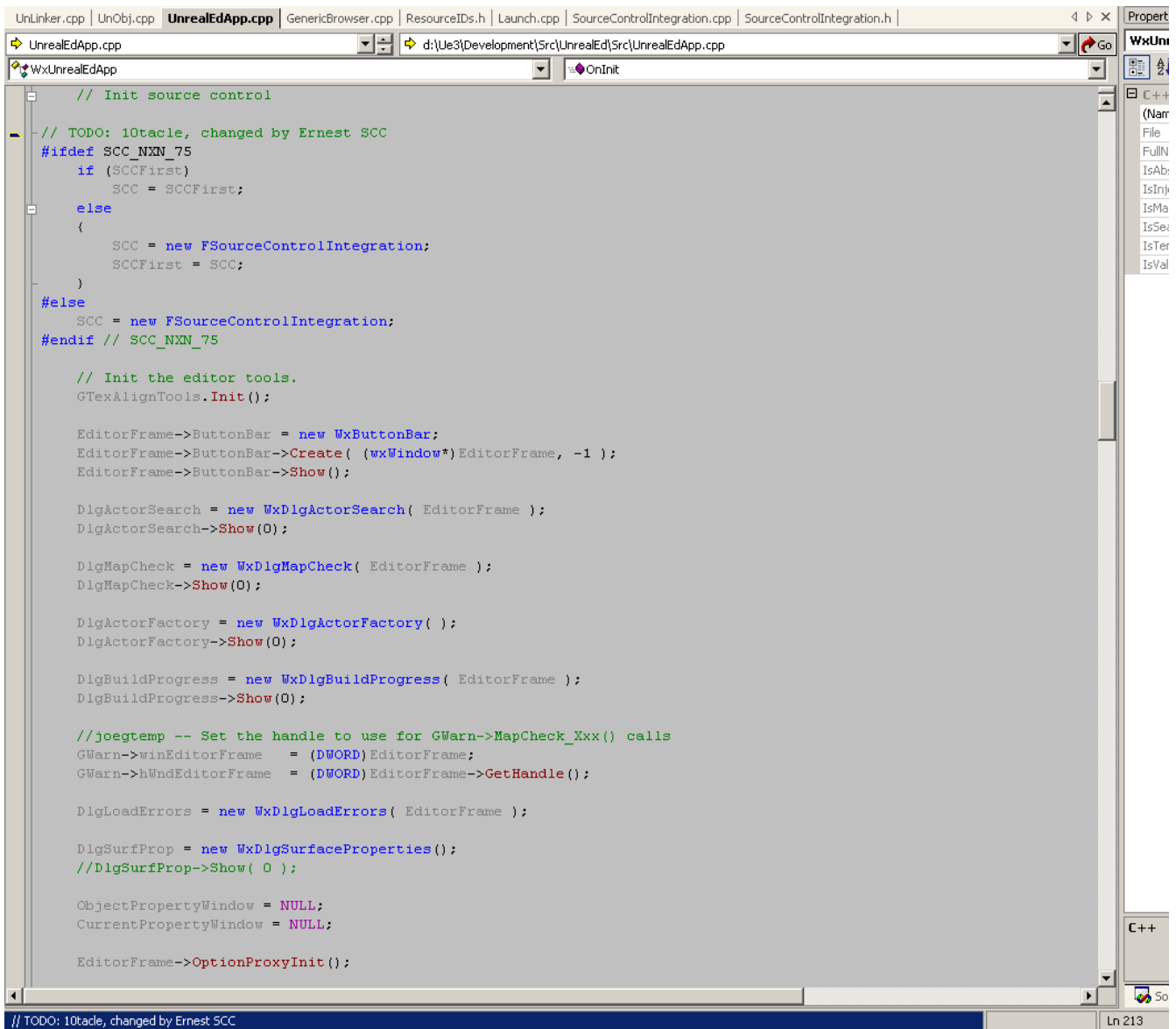


Figure 8 – UnrealEdApp2

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

- add menu item for NxN Browser

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

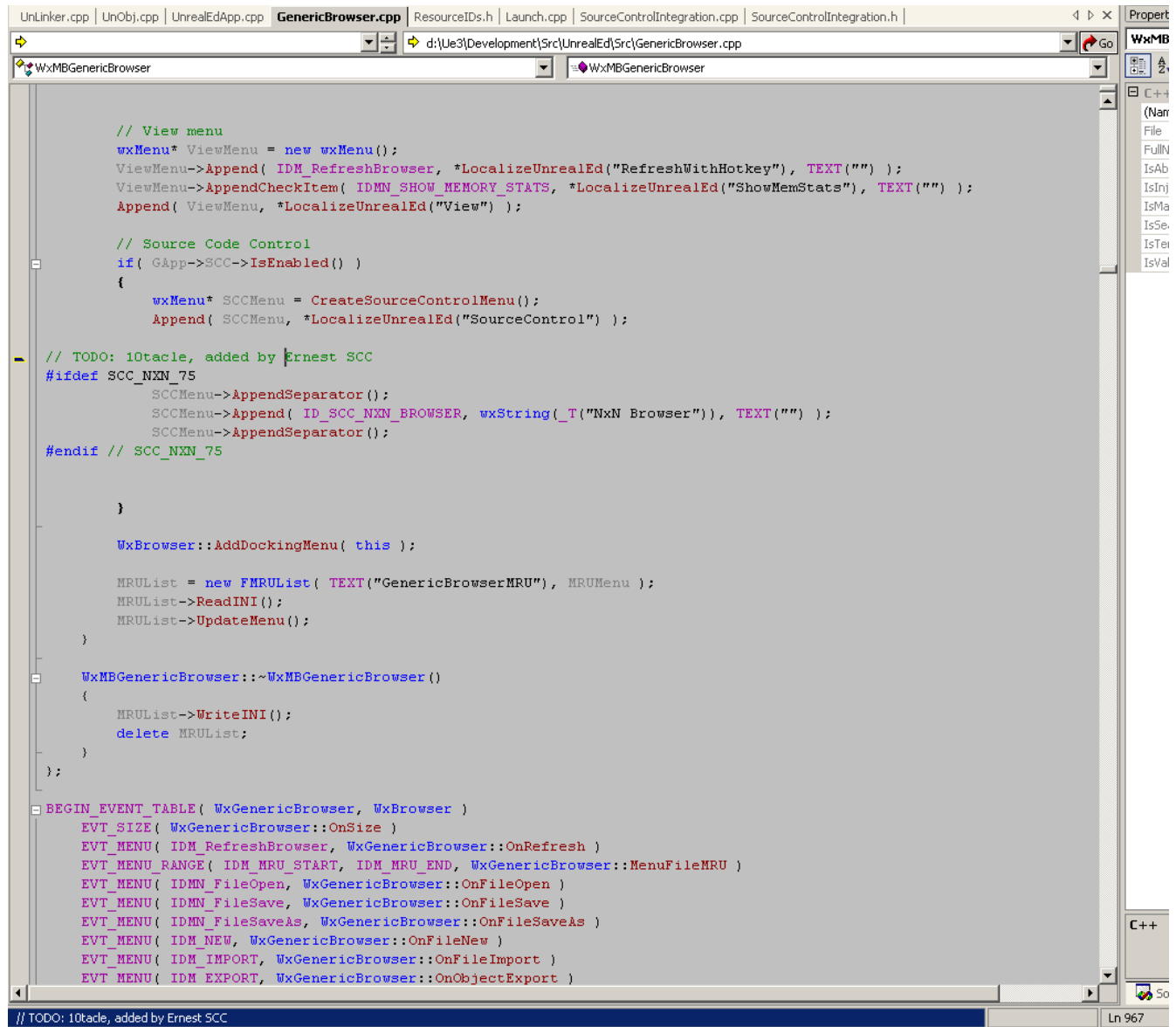


Figure 9 – *GenericBrowserI*

- add event to event table for the item

// TODO: 10tacle, added by Ernest SCC

#ifdef SCC_NXN_75

EVT_MENU(ID_SCC_NXN_BROWSER, WxGenericBrowser::OnSCCNXNBrowse)

#endif // SCC_NXN_75

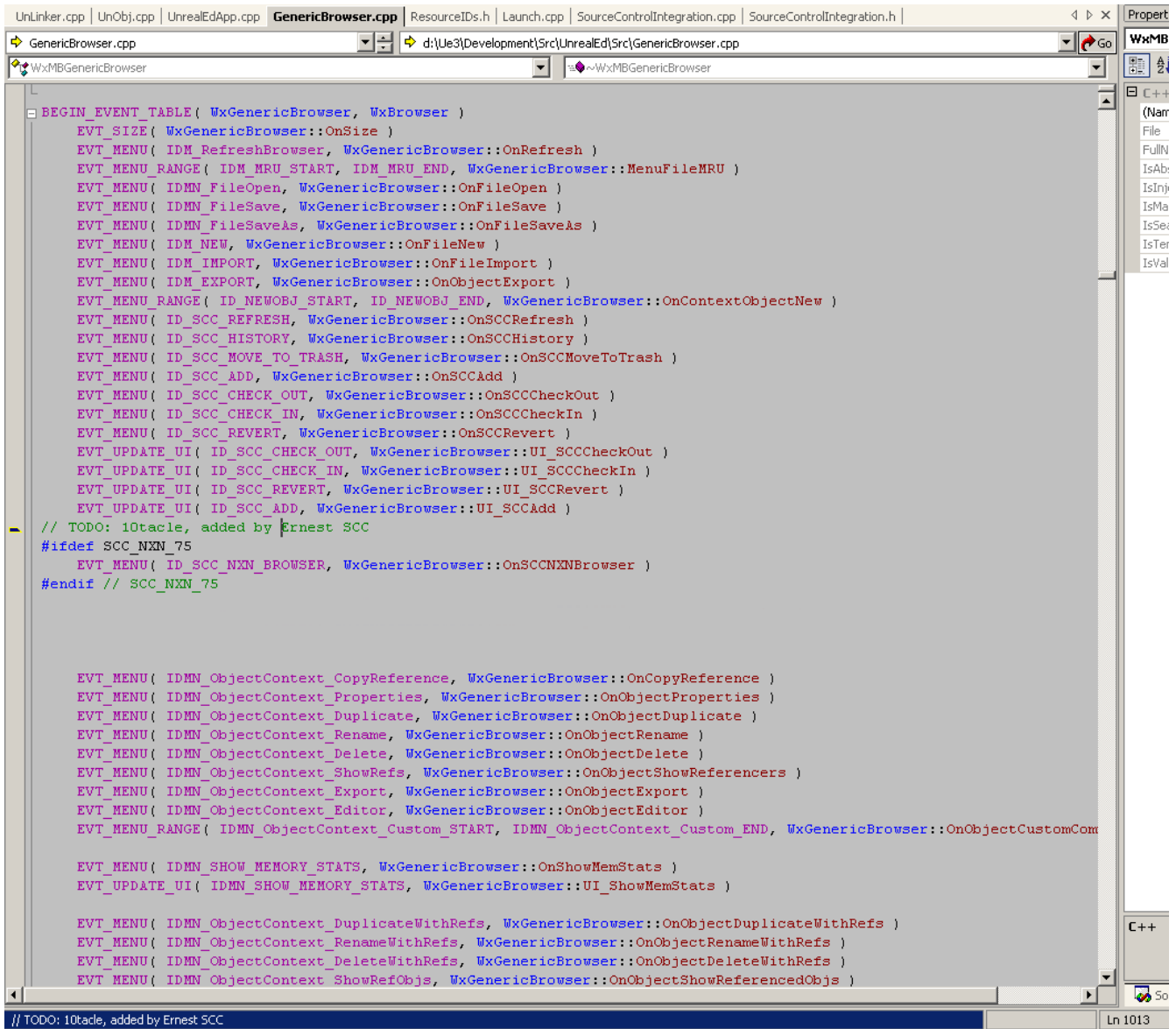


Figure 10 – GenericBrowser2

- definition of the item's processing function

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

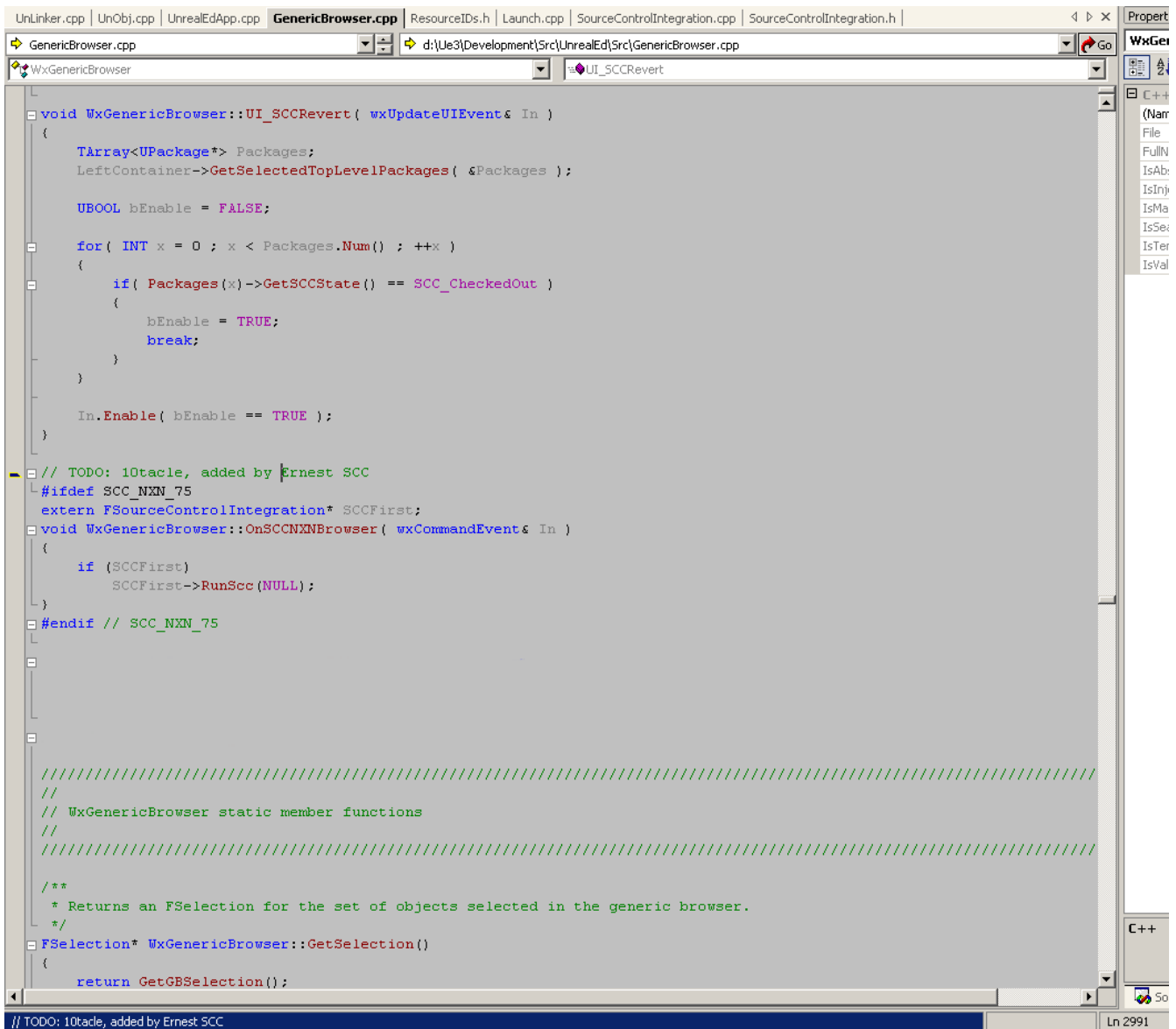


Figure 11 – GenericBrowser3

- declaration of the item's processing function

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

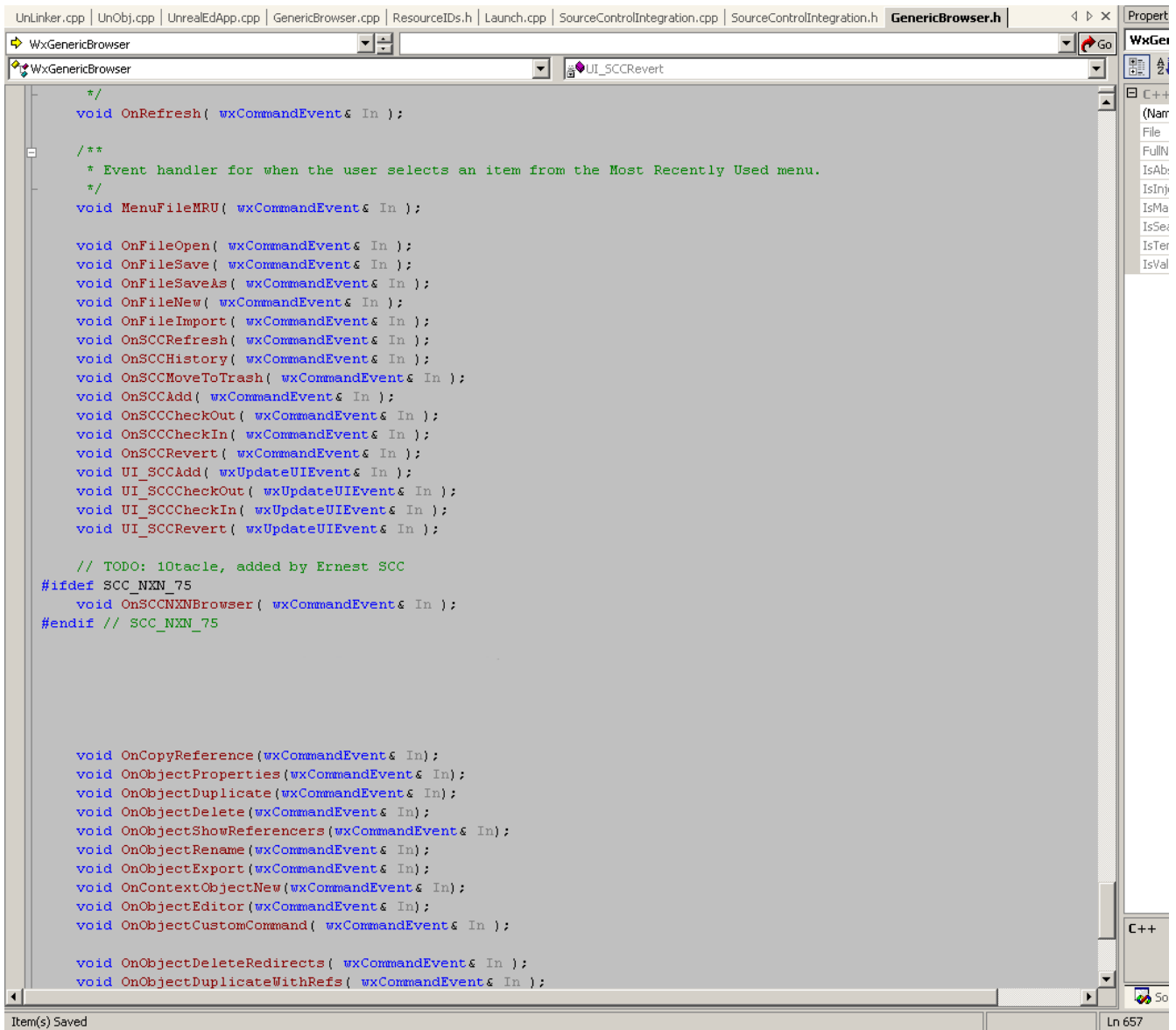


Figure 12 – GenericBrowser4

- add ID for the event

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

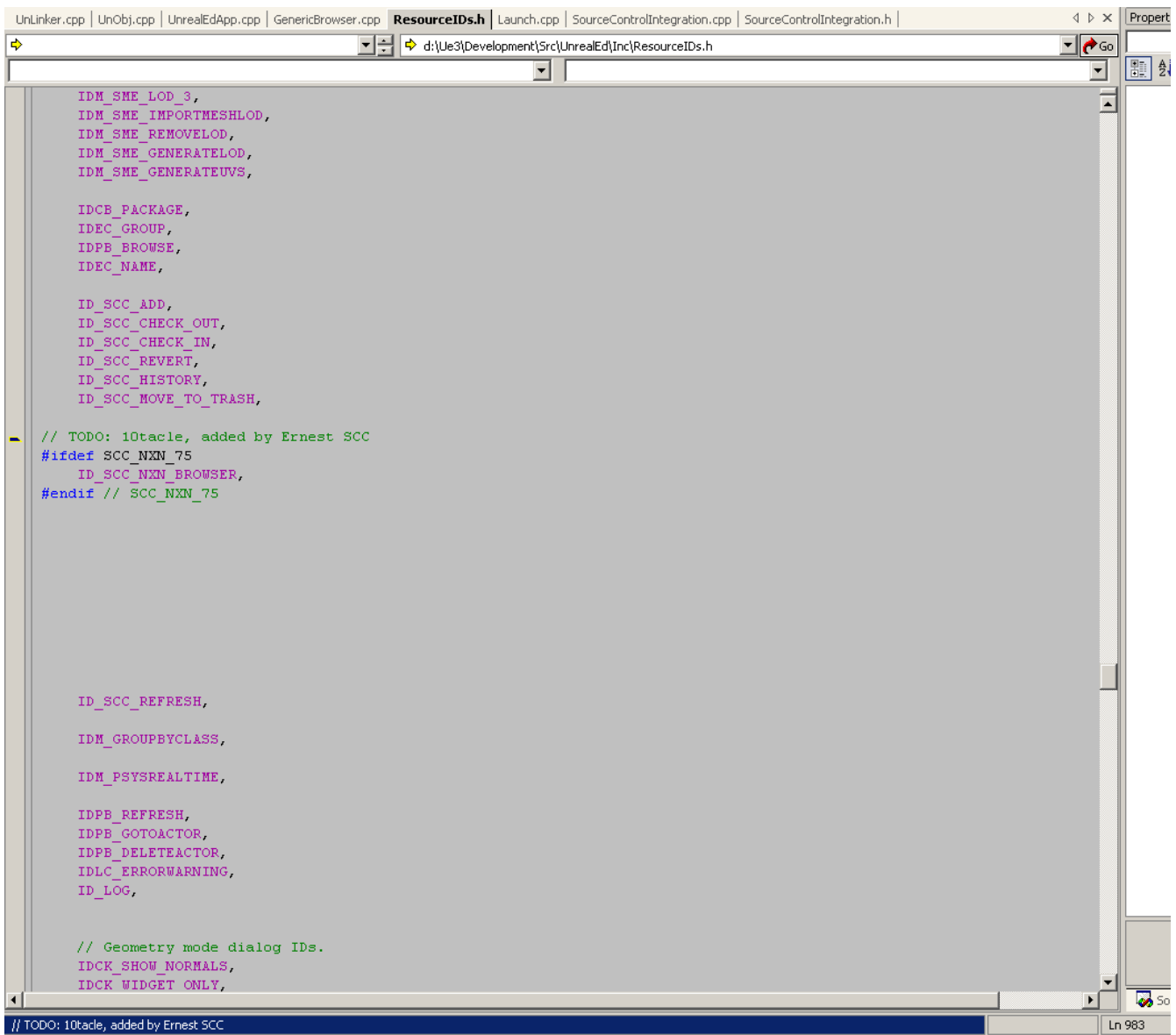


Figure 13 – ResourceIDs

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

- include SCC header and create an instance

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

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

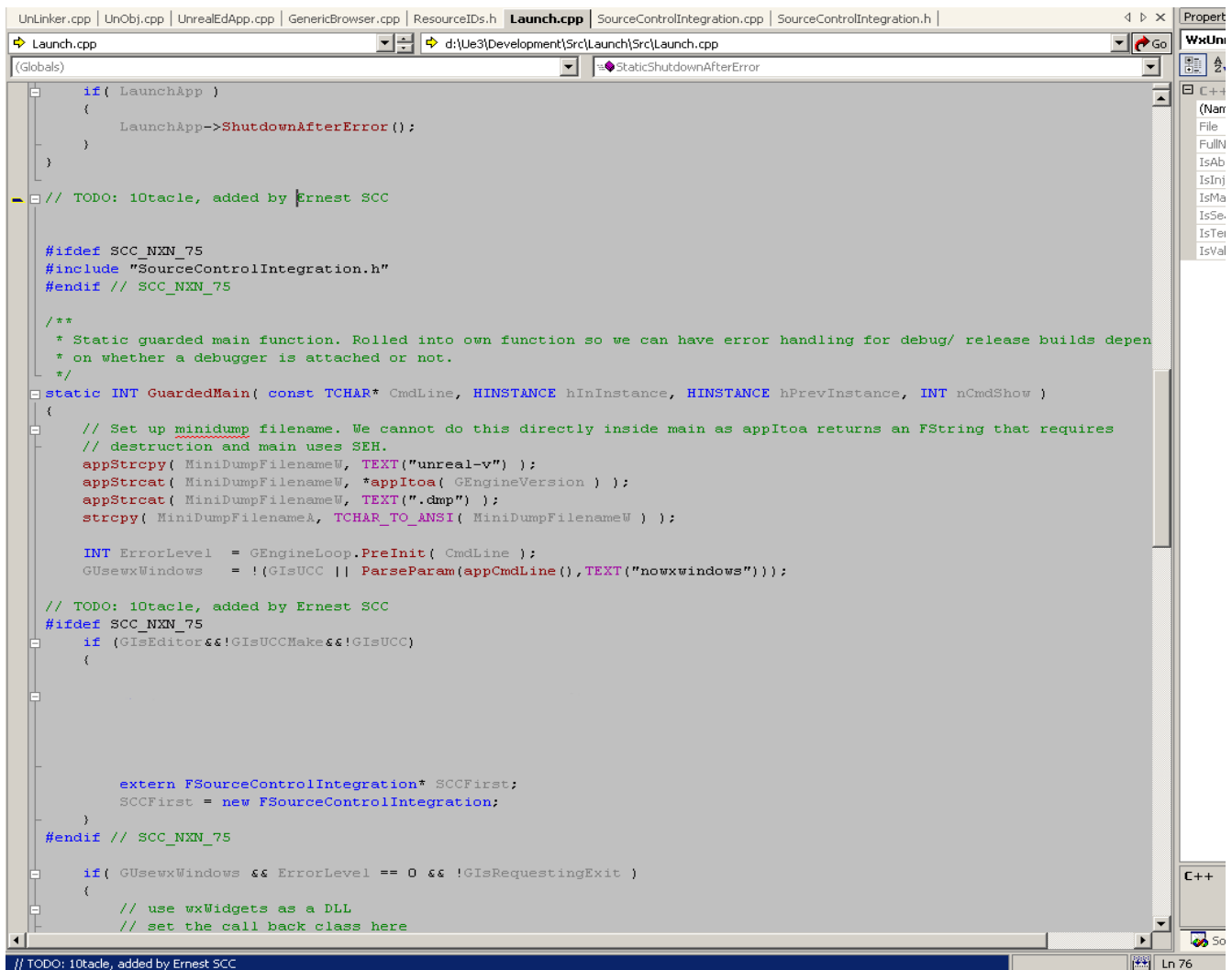


Figure 14 – Launch

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

Ernest Mikita and Filip Fischer, 10TACLE STUDIOS SLOVAKIA

