Share 0 More ▾ Next Blog» Create Blog Sign In

yourProsoft

virtuality belivers to provide IT solutions
Owner: Wael AlGhool

Friday, 20 January 2012

MFC "encountered an improper argument" message

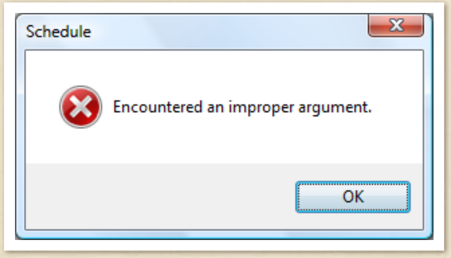
Hi there,

The story:

My MFC application "Schedule" that I work in 2 years ago faced by a very strange bug, the application is built using VC++, VS2008 on vista platform and configured to target vista platform by assigning WINVER=0x0600, _WIN32_WINNT=0x0600, _WIN32_WINDOWS=0x0410.

It runs perfectly on whatever machine running WinXP/Vista, other O.S.s are not available to test on. An overseas customer claimed a bug when she tried to save documents, her machine was HP-dv6000 with WinXP installed. I tolled her to upgrade the machine up to WinXP-SP3, but she upgraded up to Windows7 and the bug still exist.

The Bug is such a message says "encountered an improper argument", it emerged when she tried to save/open a file.



Schedule

Encountered an improper argument.

I googled every where but nobody has solution or a resolution, even in 2009 Microsoft tolled somebody that this issue is solved in VS2010, others talk about resource conflictions according to windows upgrades.

After long time I decided to investigate, where I spent 12 hours of digging into MFC source code and files, and then discovered the problem.

It was the following statement that throws exception because of an invalided resource ID.

```
        ENSURE(title.LoadString(nIDSTitle = bReplace ? AFX_IDS_SAVEFILE :
AFX_IDS_SAVEFILECOPY));
```

"title" is a CString object that has to displayed in title bar of CFileDialog, it then be loaded first from a string resource AFX_IDS_SAVEFILE or AFX_IDS_SAVEFILECOPY which are not exist at the machine of the customer.

I successfully simulated the same bug on my own machine using the following lines of code.

```
CString temp;

ENSURE(temp.LoadString(0xF012));// 0xF012 is invalide
```

What is ENSURE()?

```
#define ENSURE(cond) ENSURE_THROW(cond, ::AfxThrowInvalidArgException
() )
```

yourProSoft Founder

Wael AlGhool

Software Solutions Provider

Blog Archive

- ► 2013 (4)
- ▼ 2012 (2)
  - ► July (1)
  - ▼ January (1)
    - MFC "encountered an improper argument" message
- ► 2011 (4)
- ► 2010 (20)

008738

Bravenet Free Counter

VIEW SITE STATS

Crony Blog

The following paragraph describes what is the macro ENSURE, I cut this paragraph from MSN and past it here.

The purpose of these macros is to improve the validation of parameters. The macros prevent further processing of incorrect parameters in your code. Unlike the ASSERT macros, the ENSURE macros throw an exception in addition to generating an assertion.

The macros behave in two ways, according to the project configuration. The macros call ASSERT and then throw an exception if the assertion fails. Thus, in Debug configurations (that is, where _DEBUG is defined) the macros produce an assertion and exception while in Release configurations, the macros produce only the exception (ASSERT does not evaluate the expression in Release configurations).

The macro ENSURE_ARG acts like the ENSURE macro.

To solve this bug I made two steps, First: I replaced CWinAPP::OnFileOpen using the following code:

```
//----------------------------------------------------------------------

void CScheduleApp::OnMyFileOpen()

{

//manual open using CFileDialog

       CString strDocFileName = _T("");

CFileDialog *pDlg;

pDlg = new CFileDialog (TRUE,_T
("yps"),strDocFileName,OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT,

_T("File (*.yps)|*.yps|All Files (*.*)|*.*||"),NULL);

//

m_strCurFolder = m_strCurFolder.IsEmpty() ? GetParentFolder():m_strCurFolder;

pDlg->m_ofn.lpstrInitialDir = m_strCurFolder.GetBuffer(MAX_PATH);

pDlg->m_ofn.lpstrTitle = _T("فتح");

if(pDlg->DoModal()==IDOK)

{

strDocFileName = pDlg->GetPathName();

//the next line is optained from the following mfc source file

//C:\Program Files\Microsoft Visual Studio 9.0\VC\atlmfc\src\mfc\docmgr.cpp

AfxGetApp()->OpenDocumentFile(strDocFileName);

}

m_strCurFolder.ReleaseBuffer();

delete pDlg;

//Keep track of obtained folder as default for next open operation

m_strCurFolder = strDocFileName.IsEmpty() ? m_strCurFolder:

        (strDocFileName.IsEmpty() ? _T(""):strDocFileName.Left(strDocFileName.ReverseFind
('\\')+1));

//AfxMessageBox(m_strCurFolder);

theApp.WriteString(_T("CurUserFolder"),m_strCurFolder);

}

//----------------------------------------------------------------------
```

Second: I override CDocumment::DoSave() funtion to prevent it from calling AfxGetApp()->DoPromptFileName that uses the buggy resource, here is the code:

```
//---------------------------------------------------------------------

BOOL CScheduleDoc::DoSave(LPCTSTR lpszPathName, BOOL bReplace)

{

CString newName = lpszPathName;

if (newName.IsEmpty())

{

CDocTemplate* pTemplate = GetDocTemplate();

ASSERT(pTemplate != NULL);

newName = m_strPathName;

if (bReplace && newName.IsEmpty())

{

newName = m_strTitle;

// check for dubious filename

int iBad = newName.FindOneOf(_T(":/\\"));

if (iBad != -1)

newName.ReleaseBuffer(iBad);

// append the default suffix if there is one

CString strExt;

if (pTemplate->GetDocString(strExt, CDocTemplate::filterExt) &&

!strExt.IsEmpty())

{

ASSERT(strExt[0] == '.');

int iStart = 0;

newName += strExt.Tokenize(_T(";"), iStart);

}

}

//Replace the bug lines with a new technique

//if (!AfxGetApp()->DoPromptFileName(newName,

// bReplace ? AFX_IDS_SAVEFILE : AFX_IDS_SAVEFILECOPY,

// OFN_HIDEREADONLY | OFN_PATHMUSTEXIST, FALSE, pTemplate))

// return FALSE; // don't even attempt to save

if(!MyDoPromptFileName(newName,bReplace))

return false;

}

CWaitCursor wait;

if (!OnSaveDocument(newName))

{

if (lpszPathName == NULL)

{

// be sure to delete the file
TRY
{
CFile::Remove(newName);
}
CATCH_ALL(e)
{
//the normal place for the following Macro is in mfc\stdafx.h
//I bring it here becuase it is used only here and I have no plan to use it anywhere
#define DELETE_EXCEPTION(e) do { if(e) { e->Delete(); } } while (0)
//
TRACE(traceAppMsg, 0, "Warning: failed to delete file after failed SaveAs.\n");
DELETE_EXCEPTION(e);
}
END_CATCH_ALL
}
return FALSE;
}
// reset the title and change the document name
if (bReplace)
SetPathName(newName);
return TRUE; // success
}
BOOL CScheduleDoc::MyDoPromptFileName(CString& fileName, bool bReplace)
{
CFileDialog *pDlg;
pDlg = new CFileDialog (FALSE,_T("yps"),fileName,OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT,
_T("Timetable File (*.yps)|*.yps|All Files (*.*)|*.*||"),NULL);
CString title = bReplace ? _T("حفظ جدول الحصص"):_T("حفظ جدول الحصص في ملف آخر");
pDlg->m_ofn.lpstrTitle = title;
if(pDlg->DoModal()!=IDOK)
return FALSE;
fileName=pDlg->GetPathName();
return TRUE;
}
//----------------------------------------------------------------------
```

Posted by Wael AlGhool at 13:00

+1 Recommend this on Google

2 comments:

grejam 11 January 2013 08:15

Thanks, I have an MFC app with the same symoptom. Narrowing it down to an ENSURE macro call in an MFC header helps me.

Its naturally only in production environment, which I have limited access to. I can reproduce something like this by adding code with a call to AfxThrowInvalidArgException (which ENSURE calls). The message popup seesm to happen because of no exception handler. Adding a catch all (catch(...)) catches it. Now if I can figure out what code triggers it in production... 8-)

Reply

balaji bala 10 June 2013 22:44

Thanks a lot this is resource problem your correct ....

Reply

Enter your comment...

Comment as: Select profile...

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)