# 4.5 调试Direct3D应用程序

为了简化代码并突出重点，我们在本书的示例中省略了很多错误处理语句。不过，我们实现了一个宏，用它来检查许多Direct3D函数返回的**HRESULT**值。这个宏定义在d3dUtil.h文件中：

#if defined(DEBUG) | defined(\_DEBUG)

#ifndef HR

#define HR(x) \

{ \

HRESULT hr = (x); \

if(FAILED(hr)) \

{ \

DXTrace(\_\_FILE\_\_, (DWORD)\_\_LINE\_\_, hr, L#x, true); \

} \

}

#endif

#else

#ifndef HR

#define HR(x) (x)

#endif

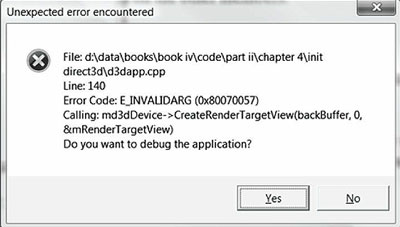
#endif

当函数的返回值表明调用失败时，我们把返回值传递给**DXTrace**函数。请注意，当使用该函数时，我们必须在代码中添加包含语句“#include<dxerr.h>”，并链接dxerr.lib库文件，只有这样程序才能通过编译。

HRESULT WINAPI DXTraceW(const char\* strFile, DWORD dwLine,

HRESULT hr, const WCHAR\* strMsg, BOOL bPopMsgBox);

该函数可以弹出一个消息框，显示出现错误的文件、行号、有关错误的描述信息以及导致错误的函数名。图4.11是它的一个例子。注意，当**DXTrace**函数的最后一个参数设为**false**时，该函数不会显示消息框，而是把调试信息输出到Visual C++的输出窗口。当我们不使用调试模式时，**HR**宏不执行任何代码。另外，**HR**必须是一个宏而不能是一个函数；否则**\_\_FILE\_\_**和**\_\_LINE\_\_**将无法引用调用**HR**宏的函数所在的文件和行。



**图4.11：当Direct3D函数返回一个错误时，通过DXTrace函数显示消息框。**

现在我们使用**HR**宏来包围返回**HRESULT**值的Direct3D函数，下面是一个例子：

HR(D3DX11CreateShaderResourceViewFromFile(md3dDevice,

L"grass.dds", 0, 0, &mGrassTexRV, 0 ));

当我们调试演示程序时，这个宏可以很好地运作，但是对于一个实际的应用程序来说，我们应该使用更完善的错误处理机制。

注意：L#x将**HR**宏的参数转换成一个Unicode字符串。通过这一方式，我们可以把导致错误的函数调用语句输出到消息框上面。