Create views of OpenCASCADE objects in the Debugger

eryar@163.com

Abstract. The Visual Studio Natvis framework lets you customize the way Visual Studio displays native types in debugger variable windows such as the Watch, Locals and Data Tips windows. It supersedes the autoexp.dat file that has been used in earlier versions of Visual Studio and offers XML syntax, better diagnostics, versioning, and multiple file support. The container in OpenCASCADE is difficult for debugging, such as TColStd_Array1OfInteger in the TColStd package, .etc. Use the natvis framework to create views of these objects will make it easy for developers to inspect them during debugging and so accelerate the debug process.

Key Words. Visual Studio Natvis, OpenCASCADE

1. Introduction

因为 OpenCASCADE 早期使用 C 开发，所以自定义了一些容器类，如包 TColStd 中的类，TColGeom 包及包 TopTools 中的类等，这些类在 C++中都可以使用 STL 来替代了。这些类在 Debug 过程中，很难查看其中的值，如 TColStd_Array1OfInteger，这个类相当于 std::vector<int>，但是 Debug 时只能看到数据的指针，不容易查看容器中每个元素的值，如下图 1.1 所示：

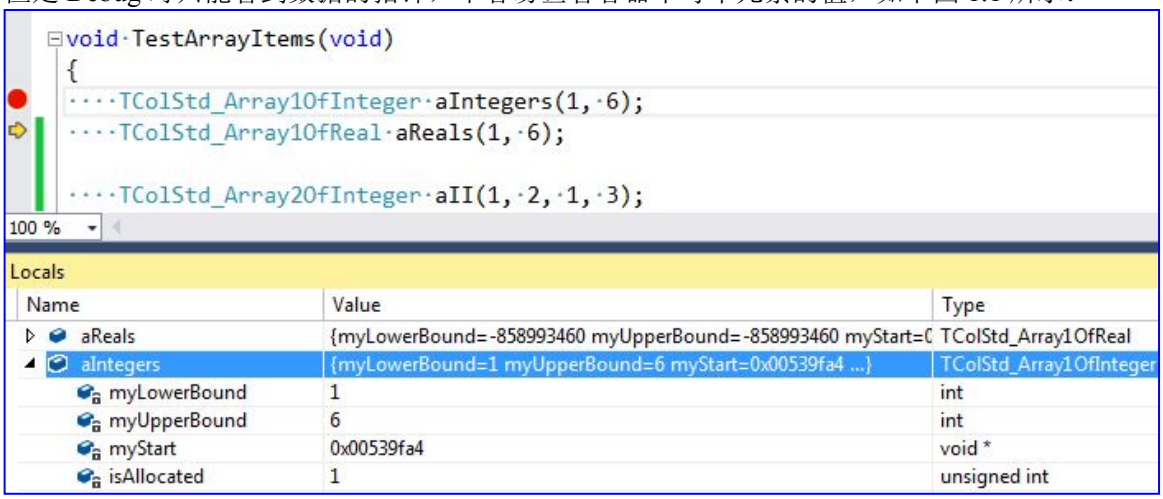

Figure 1.1 View of TColStd_Array1OfInteger in Locals Window

由上图 1.1 可知，对于这个类的对象，Debug 时只能看到数据的起始指针。为了方便自定义类型调试，Visual Studio 在 2012 版本以后，引入了 Natvis 框架，用来替代原来的 autoexp.dat 来为自定义类型定义调试时的数据显示。Natvis 使用了 XML 文件，可读性更好，易于实现。

本文使用 Visual Studio 的 Natvis 框架，来对 OpenCASCADE 中的一些容器类数据进行可视化，方便开发者对 OpenCASCADE 的调试。

2. For Array Container

对于 OpenCASCADE 的包 TColStd 中的数组类，定义其 natvis 如下所示：

```
<Type Name="TColStd_Array1OfInteger">
    <DisplayString Condition="isAllocated != 1">empty</DisplayString>
    <DisplayString>{{size = {myUpperBound - myLowerBound + 1}}}</DisplayString>
    <Expand>
        <Item Condition="isAllocated == 1" Name="[size]">myUpperBound - myLowerBound
+ 1</Item>
        <ArrayItems Condition="isAllocated == 1">
            <Size>myUpperBound - myLowerBound + 1</Size>
            <ValuePointer>(Standard_Integer*)(myStart)                    +
myLowerBound</ValuePointer>
        </ArrayItems>
    </Expand>
</Type>
```

调试时数据显示如下图 2.1 所示：

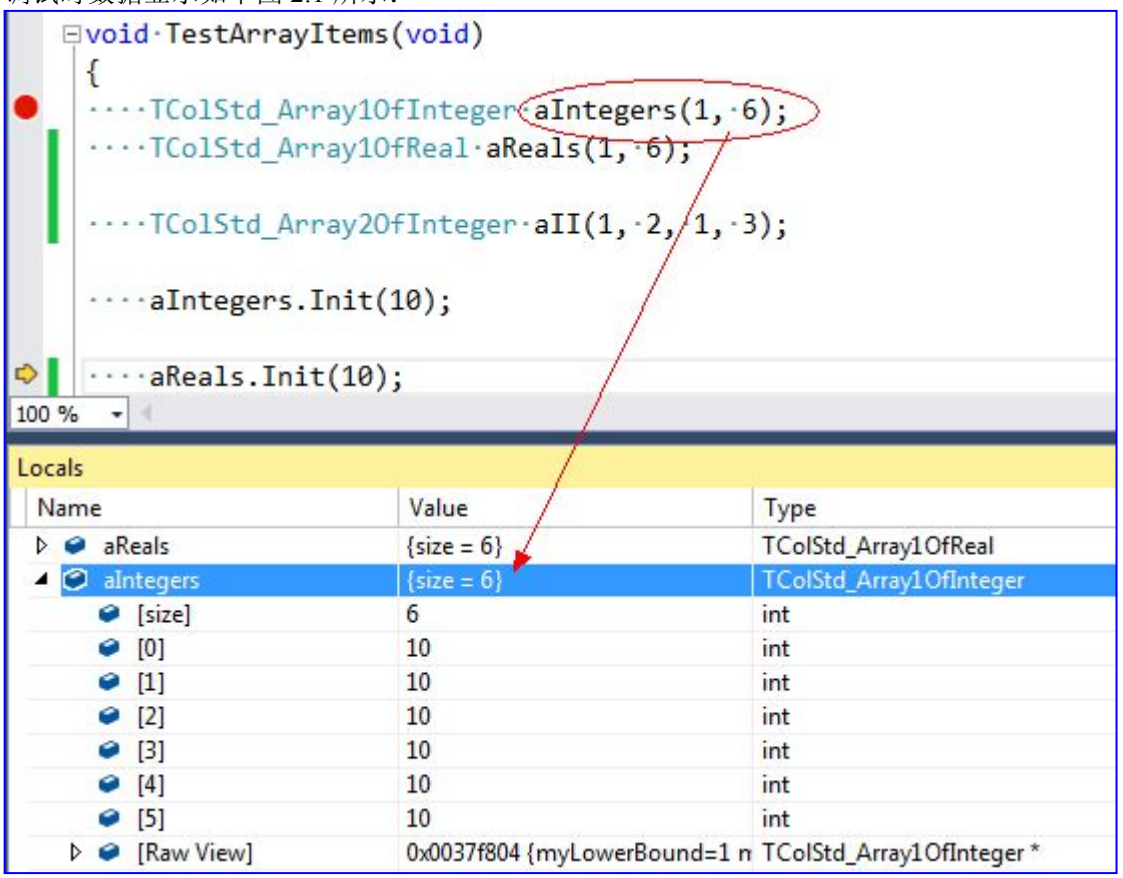

Figure 2.1 OpenCASCADE array in Locals Windows

同理，可对此包中其他一维数组使用同样的规则，即可对其中的数据可视化，与 std::vector 显示的效果一样，方便调试。

3. For List Container

对于 OpenCASCADE 的包 TColStd 中的链表类，定义其 natvis 如下所示：

```
<Type Name="TColStd_ListNodeOfListOfInteger">
    <DisplayString>{{current = {myValue}}}</DisplayString>
    <Expand>
        <LinkedListItems>
            <HeadPointer>this</HeadPointer>
            <NextPointer>(TColStd_ListNodeOfListOfInteger*)myNext</NextPointer>
            <ValueNode>this-&gt;myValue</ValueNode>
        </LinkedListItems>
    </Expand>
</Type>

<Type Name="TColStd_ListOfInteger">
    <DisplayString Condition="myFirst == 0">empty</DisplayString>
    <Expand>
        <Item Name="first">(TColStd_ListNodeOfListOfInteger*)myFirst</Item>
    </Expand>
</Type>
```

调试时对于类 TColStd_ListOfInteger，natvis 诊断说找不到类 TColStd_ListNodeOfListOfInteger 定义，当跟踪到此类一个具体函数时，就可以看到具体的值了：

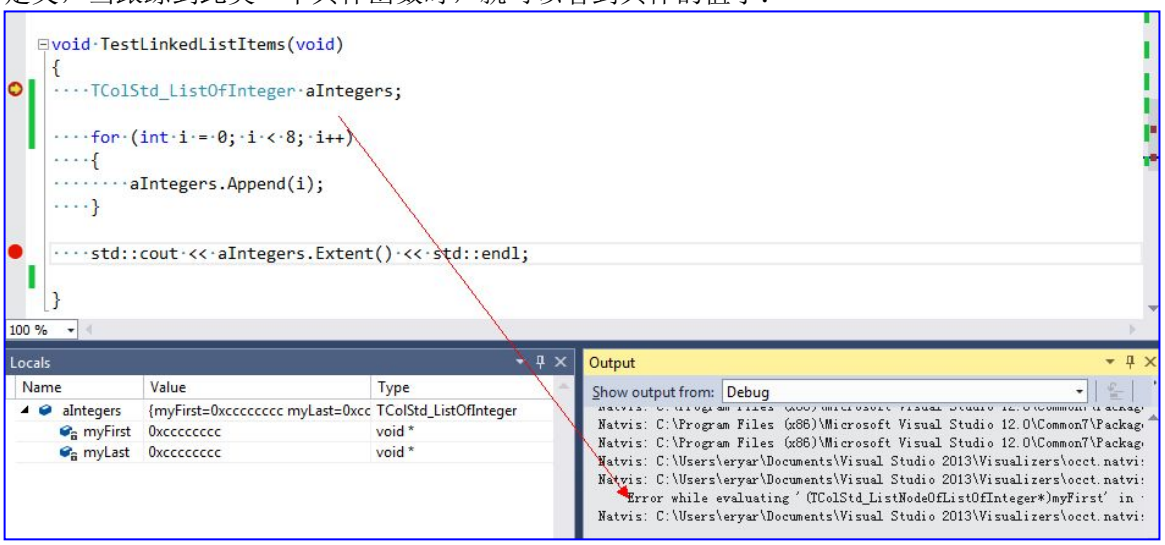

Figure 3.1 Natvis gives a Error info

跟踪到 TColStd_ListOfInteger 内部后，就可以看到类 TColStd_ListNodeOfListOfInteger 中的数据了，但是从 TColStd_ListOfInteger 的函数中出来后，就看不到了。

如果这个问题解决了，对于类 TopoDS_ListOfShape 中的数据也可以用同样的方式来显示，极大地方便了开发者对其调试。如果你对此有解决办法，欢迎不吝赐教。

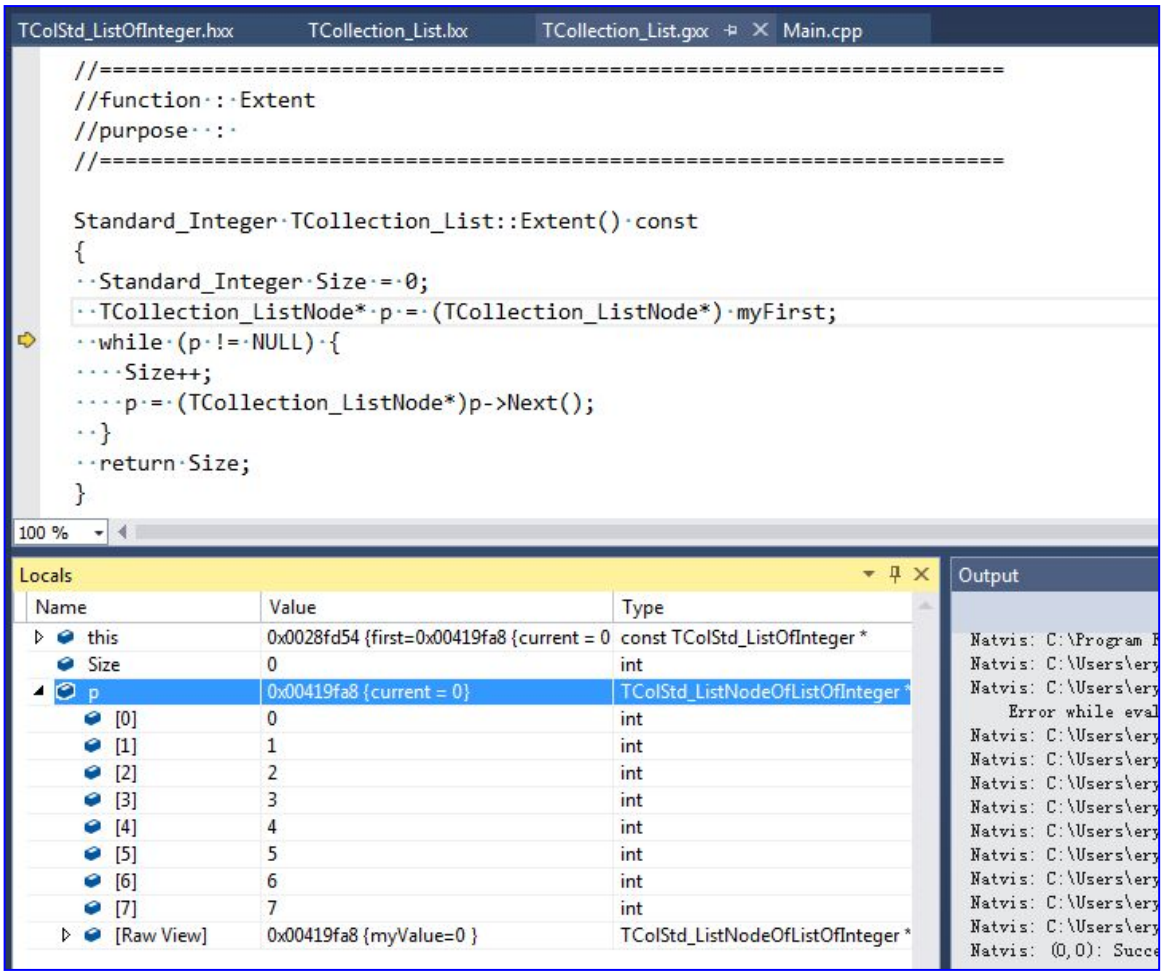

Figure 3.2 Data view for TColStd_ListNodeOfListOfInteger

先从简单的容器类着手，解决后可将 TopoDS_ListOfShape 中的数据显示出来，为 OpenCASCADE 程序的调试提供方便。

4. Conclusion

Visual Studio 2012 版本以后引入了 Natvis 框架来对自定义的类进行可视化，方便调试。OpenCASCADE 中有很多容器类直接使用了指针，调试程序时数据很不直观。应用 Natvis 来对一些类在调试时的视图进行配置，可以方便查看其中数据，使 OpenCASCADE 的调试更轻松。

对于一维数组的 natvis 定义还是很简单的，但是对于 List 出现了问题。如果这个问题解决了，对 TopoDS_ListOfShape 的可视化也可做同样的处理，方便造型算法调试。若您有解决方案，望不吝赐教。

5. References

1. Create custom views of native objects in the debugger.
   https://msdn.microsoft.com/en-us/library/vstudio/jj620914.aspx
2. Writing debugger type visualizers for C++ using .natvis files
   https://code.msdn.microsoft.com/Writing-type-visualizers-2eae77a2#content
3. vczh. C++实用技巧之配置 Visual C++的调试器显示数据结构的格式.
   http://www.cppblog.com/vczh/archive/2013/03/21/198665.html
4. stl.natvis in %VSINSTALLDIR%\Common7\Packages\Debugger\Visualizers
5. qt5.natvis in %VSINSTALLDIR%\Common7\Packages\Debugger\Visualizers