

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# VC++ 2005: 与ISO-C++的集成

李建忠

上海祝成科技 高级讲师

2005年7月4日

# Agenda

- C++/CLI 集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

# 一些有关C++/CLI的问题

- 一个CLI托管类型能够继承一个本地类型吗?
- 一个CLI托管数组能够持有多个本地类型实例吗?
- 一个本地类型能够实现多个CLI接口吗?
- 一个本地类型能够使用CLR的垃圾收集服务吗?
- 一个在ISO-C++中抛出的异常能够在CLI中被捕获吗?

# C++/CLI中的集成技术

- 这些问题的答案实际上就是C++/CLI的集成技术所做的选择。
- C++/CLI对于ISO-C++采取了“集成”而非“替换”的策略, 为基于ISO-C++的静态程序设计和基于CLI的动态程序设计建立了一座互联互通的桥梁。
- C++/CLI的集成既不是让CLI全盘替换ISO-C++, 也不是将CLI和ISO-C++全盘融合, 亦不是让二者完全孤立并存。而是“有所为, 有所不为”地融合集成。

# C++/CLI集成的几个方面

- 代码/函数的集成
  - 变量, 表达式, 操作符, 类型转换, 语句, 异常、函数调用
- 类型的集成
  - 类型继承、类型修饰、类型成员、构造与析构、参数化类型 (模板)
- 对象模型的集成
  - 内存管理、对象包含、对象布局、对象多态、指针与引用

# Agenda

- C++/CLI中集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

# IJW (it just works) 技术

- IJW技术保证了所有的ISO-C++代码经过cl/cclr编译后行为保持不变。
- 在将ISO-C++代码和CLI代码混合时, IJW技术同样保证ISO-C++代码的行为保持不变。
- 绝大多数ISO-C++代码编译后将得到托管代码, 即IL代码。部分不能编译为IL代码的采用P/Invoke调用实现。换言之, IJW使用IL代码来表达ISO-C++语言的行为。

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 代码演示 (1)

## 用CL/CLR编译ISO-C++代码

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 代码演示 (2)

## 用CL/CLR编译混合代码

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 代码演示 (3)

## 异常处理的集成

# Agenda

- C++/CLI中集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

# 本地类型的简单集成

- **IJW**对简单类型采用透明的集成策略, 所有**ISO-C++**中的简单类型在**CLI**中被同样对待, 具有同样的行为。
- **IJW**将**ISO-C++**本地自定义类型编译为**ValueType**的子类型。**IJW**通过全局函数来实现**ISO-C++**本地类型的成员函数。通过对类型实例中的位操作, 来访问**ISO-C++**中本地类型的数据成员。
- **C++/CLI**对于类型集成采取比较保守的策略, **ISO-C++**类型系统和**CLI**类型系统仍然保持各自独立

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 代码演示 (4)

## 用CL/CLR编译本地类型

# 本地类型的混合集成

- **C++/CLI** 不支持本地类型与**CLI** 托管类型之间的继承。
- **C++/CLI** 不支持在本地类型上应用**CLI** 托管类型所使用的修饰符
- **C++/CLI** 不支持在本地类型上应用**CLI Attribute**
- **C++/CLI** 不支持在本地类型内添加**LCI** 托管类型所具有的成员。
- **C++/CLI** 托管类型的构造和析构语义有所改变。
- **C++/CLI** 支持在托管类型上使用**ISO-C++** 的静态模板机制, 但不支持在本地类型上使用**CLI** 的动态泛型机制

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 代码演示 (5)

## 本地类型的混合集成

# Agenda

- C++/CLI中集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

# 对象内部机制的集成

- **C++/CLI** 只支持对托管引用类型进行垃圾收集服务，不支持对**ISO-C++**本地类型的垃圾收集服务。
- **C++/CLI** 中的托管类型的内存布局也和本地类的内存布局有明显的不同。
- **C++/CLI** 中类型的多态机制（虚拟）也不同于本地类的多态机制。

# 对象模型结构的集成

- **C++/CLI** 目前不支持在托管对象中内嵌本地对象。
- **C++/CLI** 目前不支持在本地对象中内嵌托管对象。
- **C++/CLI** 支持直接在托管对象中内嵌本地对象的指针。
- **C++/CLI** 支持“间接地使用 **gcroot** 类”在本地对象内嵌托管对象的指针。

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 代码演示 (6)

## 在托管对象中内嵌本地对象的指针

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 代码演示 (7)

## 在本地对象内嵌托管对象的指针 (gcroot类的使用)

# Agenda

- C++/CLI中集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

# 讲座总结

- **C++/CLI** 对于 **ISO-C++** 采取了“集成”而非“替换”的策略，为基于 **ISO-C++** 的静态程序设计和基于 **CLI** 的动态程序设计建立了一座互联互通的桥梁。
- **C++/CLI** 的集成是涵盖从源代码到对象模型的多层次集成；所谓“有所为，有所不为”。
- 学习掌握 **C++/CLI** 除了要掌握 **ISO-C++** 本地编程模型和 **CLI** 托管编程模型外，还要掌握其中的集成技术。

# Agenda

- C++/CLI中集成技术简介
- 代码/函数的集成
- 类型的集成
- 对象模型的集成
- 讲座总结
- Q&A

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 资源链接

- [msdn.microsoft.com/visualc](http://msdn.microsoft.com/visualc)
- [comp.lang.c++.moderated](http://comp.lang.c++.moderated)
- [blogs.msdn.com/slippman](http://blogs.msdn.com/slippman)
- [pluralsight.com/blogs/hsutter](http://pluralsight.com/blogs/hsutter)
- [blog.joycode.com/lijianzhong](http://blog.joycode.com/lijianzhong)
- [www.chinaitclub.org/forums/](http://www.chinaitclub.org/forums/)

您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

**Microsoft**<sup>®</sup>

msdn  


**MSDN Webcasts**

# Q&A

如需提出问题, 请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后, 请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ✕

在此会议中尚未解答任何问题。

要向演示者提问, 请在此处键入问 