

C++ 中 Windows 编程概述

你可以使用 Visual C++ 编写一系列在 Windows PC (x86、x64 或 ARM)、Windows 服务器、云端或在 Xbox 上运行的程序。编写良好的 C++ 程序快速、高效且节约功耗，能够充分利用多核和众核设备、图形处理单元 (GPU) 上的泛型计算，以及其他最新的硬件技术进步。

Windows 应用有好几大类别可以使用 Visual C++ 进行开发。这些类别具有不同的编程模型或应用模型，这意味着它们使用的库和 API 可以提供对平台的访问权限和用户界面。

Windows 通用应用。Windows 8 引入了第三类 Windows 应用，并且该类别的应用在 Windows 10 中继续受支持。这些应用通常被认为只是“Windows 应用”，并且它们包括面向各种设备的桌面和移动应用。通过使用 Windows 运行时库 (WRL)，可以使用 C++/CX 或标准的 C++ 和 COM 编写这些应用，C++/CX 是 C++ 的一种方言，支持 Windows 运行时开发。虽然在 Windows 10 中，用户可以选择在桌面窗口中运行这些应用，但其实它们的最初设计是用于运行全屏幕。这些应用面向触摸设备，但如果用户爱好或触摸屏不可用，也可以使用鼠标进行操作。由于这些应用都是通过 Windows 应用商店发布，因此，又称为“Windows 应用商店应用”。

桌面、服务器以及云应用程序和游戏。此类别包括 Windows 桌面应用程序，由于这些应用程序在 Windows 8 之前使用的是 Win32 API，有时也称为 Win32 应用程序，所有 Windows 应用程序都位于此类别中。此类别中的应用程序可以使用适用于用户界面的 MFC 和 ATL 与 Windows 组件（通常是

COM 对象) 进行交互。

应用程序、组件或使用标准 C++ 编写的库也适用于此类别。

此类别还包括在服务器和云编程的上下文中使用 C++ 编写核心组件和计算代码。有时, 还会使用 C++ 编写核心服务器或云应用程序的性能强化代码以最大化性能。可以将此类代码编译为 DLL, 并通过 C# 或 Visual Basic 使用它。

.NET framework 应用程序。大多数 .NET Framework 应用程序都是使用 C# 或 Visual Basic 编写的, 但也可以使用 C++/CLI (Visual C++ 中的/clr 编译器选项) 编写。我们建议在包含托管和本机代码的较大应用程序中使用 C++/CLI 编写最小互操作层。

Windows 通用应用

使用 Windows 10, 应用程序能够在所有 Windows 10 设备 (如平板电脑和移动电话) 上以及在桌面上运行。在桌面上, 它们能够作为桌面窗口运行而不始终运行全屏幕。这些应用还可以在 Xbox 上以及在将来使用的设备上运行。这两种类型的应用的编程模型不同于 Win32 桌面应用程序。这些 Windows 应用可以在 Windows 运行时上运行, Windows 运行时可以为这些应用提供用户界面元素和重要服务以及为受支持的各种硬件设备提供接口。这些应用编译为本机代码, 具有 XAML 用户界面, 或使用 DirectX。还可以使用可供其他 Windows 应用使用的本机代码来编写 Windows 运行时组件, 这些应用包括使用 C#、Visual Basic 或 JavaScript 编写的应用。有关详细信息, 请参阅使用 C++ 创建你的第一个 Windows 应用商店应用、使用 DirectX 创建你的第一个 Windows 应用商店游戏, 以及使用 C++ 创建 Windows 运行时组件。

有关通用 Windows 平台的示例，请参阅 GitHub 上的 Windows 通用示例

如果具有现有的 Windows 8.1 项目并想要将其移植到 Windows 10，请参阅 迁移到通用 Windows 平台。 如果你想要将现有的经典 Win32 桌面库和代码集成到 UWP 应用，请参阅 如何：在通用 Windows 平台应用中使用现有 C++ 代码。

你也可以不使用 Visual C++ 组件扩展 (C++/CX) 编写通用 Windows 应用、游戏和组件，而是使用 Windows 运行时 C++ 模板库 (WRL)。 有关详细信息，请参阅 Windows 运行时 C++ 模板库 (WRL)。

使用 Visual Studio 2015 中的 Visual C++，你可以开发在 Windows 10 桌面和移动设备上运行的 Windows 通用应用。 你还可以在 Visual Studio 2015 中的 Visual C++ 中开发 Windows 8.1 应用和 Windows Phone 8.1 应用，但若要这样做，首先必须在同一台计算机上安装 Visual Studio 2013，然后配置项目以使用“Visual Studio 2013 (v120)”工具集。 若要在项目中配置此设置，打开项目的属性并在“常规”部分，将“平台工具集”设置为“Visual Studio 2013 (v120)”。

如果在 Visual Studio 安装中安装 Phone 8.0 工具，你还可以将 Windows Phone 8.0 定为目标。

Windows 10 中引入了一种名为 API 约束的新概念，可替换针对特定 Windows 版本的传统做法。 而且，你还可以选择你的应用所需的 API 约定类型，它随后将在支持这些约定的任意 Windows 设备上运行。 API 约定是一组稳定的 API，可提供对平台或设备资源的访问。 API 约定可作为引用包含在项目系统中。 在 Visual Studio 项目中，如果你将引用添加到特定的扩展 SDK，则 Visual Studio 会添加相应的 API 约定。

有关所有这些概念的详细信息，请参阅 Windows 通用应用指南。

桌面、服务器以及云应用和游戏

在云端，你可以使用 C++ 编写 Azure 本机代码程序集，并通过使用 C# 创建的 Web 角色调入它们。 有关详细信息，请参阅 Azure SDK。

若要了解桌面 Windows 客户端应用程序的基础知识，请参阅使用 C++ 开发 Windows 应用程序和使用 C++ 的 Windows 编程介绍。

在 Windows 10 中，可以使用 Visual C++ 创建多种程序：

命令行应用和实用工具。 有关详细信息，请参阅控制台应用程序。

在 PC 或 Xbox 上运行的 DirectX 游戏。 有关详细信息，请参阅 DirectX 开发人员中心。

具有复杂图形用户界面的使用者应用程序。 有关详细信息，请参阅 Hilo：
开发适用于 Windows 的 C++ 应用程序

在 .NET Framework 上运行的企业和业务线应用，或在 .NET Framework 应用与使用本机代码编写的应用或组件之间充当桥梁的企业和业务线应用。 有关详细信息，请参阅使用 C++/CLI 进行 .NET 编程。

在本机代码中运行的 SQL 数据库客户端。 有关详细信息，请参阅 SQL Server Native Client。

Microsoft Office 应用程序的外接程序。 有关详细信息，请参阅为 Outlook 2010 构建 C++ 外接程序

设备驱动程序。 有关详细信息，请参阅 Windows 驱动程序工具包 (WDK)

Windows 服务。 有关详细信息，请参阅 Windows 服务应用程序介绍。

你可以使用 Visual C++ 在 Win32 DLL 或 COM DLL 中打包绝大多数自定义

义高性能功能，这些功能可供 C++ 应用或使用其他语言（例如，C# 或 Visual Basic）编写的应用使用。有关 Win32 DLL 的详细信息，请参阅 Visual C++ 中的 DLL。有关 COM 开发的详细信息，请参阅 Component Object Model (COM)。

SDK 和标头文件

Visual C++ 包含 C 和 C++ 标准库、标准模板库 (STL) 以及其他 Microsoft 特定库。包含这些库的标头文件的包含文件夹位于 \VC\ 文件夹下的 Visual Studio 安装目录，或者 C 运行时 (CRT) 库、Windows SDK 安装文件夹，例如，Windows 10 SDK 的程序文件文件夹中的 Windows Kits\10。Microsoft 库包括：

Microsoft 基础类 (MFC)：一种面向对象的框架，用于创建传统的 Windows 程序（特别是企业级应用程序），这些应用程序具有诸如功能按钮、列表框、树视图和其他控件等丰富的用户界面。有关详细信息，请参阅 MFC 桌面应用程序。

活动模板库 (ATL)：一种功能强大的帮助程序库，用于创建 COM 组件。有关详细信息，请参阅 ATL COM Desktop Components。

C++ AMP (C++ Accelerated Massive Parallelism)：一种可以在 GPU 上实现高性能泛型计算工作的库。有关详细信息，请参阅 C++ AMP (C++ Accelerated Massive Parallelism)。

并发运行时：一种可以简化多核和众核设备编程的并行和异步编程工作的库。有关详细信息，请参阅并发运行时。

许多 Windows 编程方案还需要 Windows SDK，Windows SDK 包括可以实现对 Windows 操作系统组件访问的标头文件。默认情况下，所有 Visual Studio

2015 版本都会安装 Windows SDK, 用于实现通用 Windows 应用的开发。若要开发用于 Windows 10 的通用 Windows 应用, 则需要 Windows SDK 的 Windows 10 版本。有关 Windows 10 SDK 的信息, 请参阅 Windows 10 SDK。

(有关适用于 Windows 早期版本的 Windows SDK 的详细信息, 请参阅 Overview of the Windows SDK)。

其他平台 (例如, Xbox 和 Azure) 有自己 SDK, 你可能需要安装这些 SDK。有关详细信息, 请参阅 DirectX 开发人员中心和 Azure 开发人员中心。

开发工具

Visual Studio 包含一个功能强大的本机代码调试器、静态分析工具、图形调试工具、一个功能齐全的代码编辑器、单元测试支持, 以及许多其他工具和实用程序。有关详细信息, 请参阅 Visual Studio 中的应用程序开发和 IDE 和开发工具。