

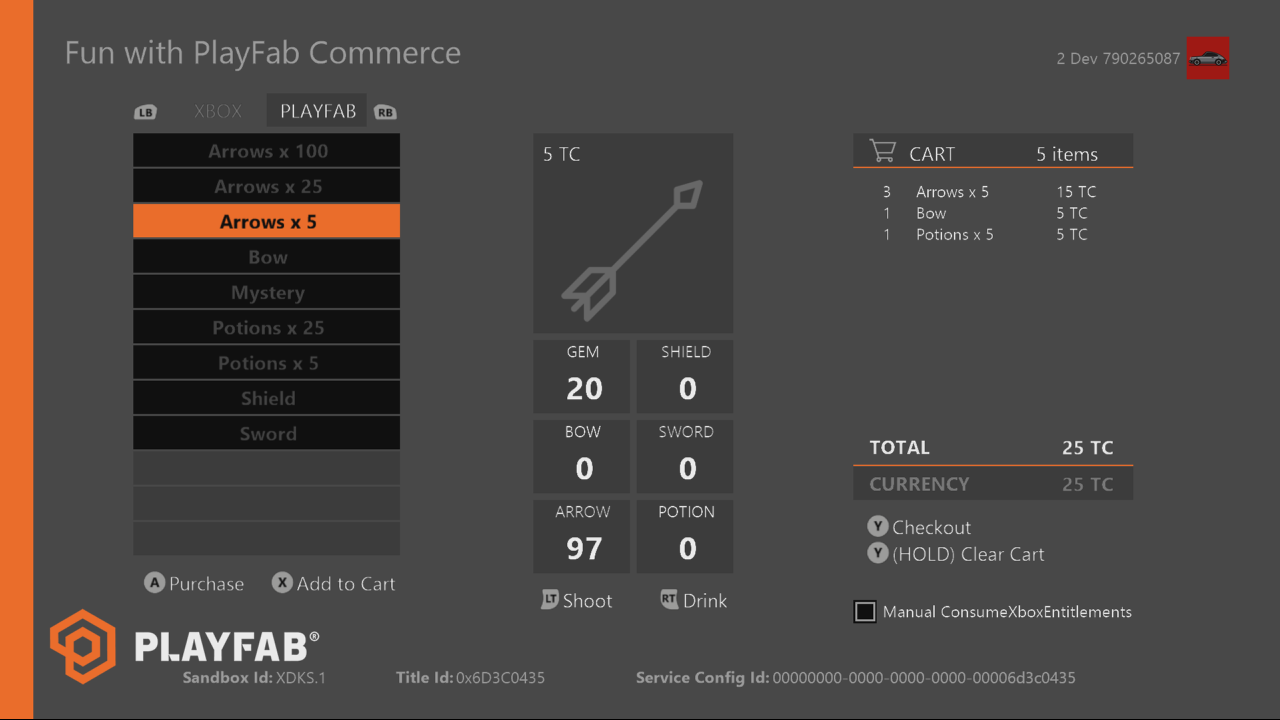
PlayFab 商业版

*\*此样本与 2018 年 6 月发布的 XDK 兼容*

# 描述

此样本演示如何与 Microsoft Store 和 PlayFab Economy 进行交互，以实现由 PlayFab 服务调解的游戏内虚拟物品和货币目录。

# 使用样本

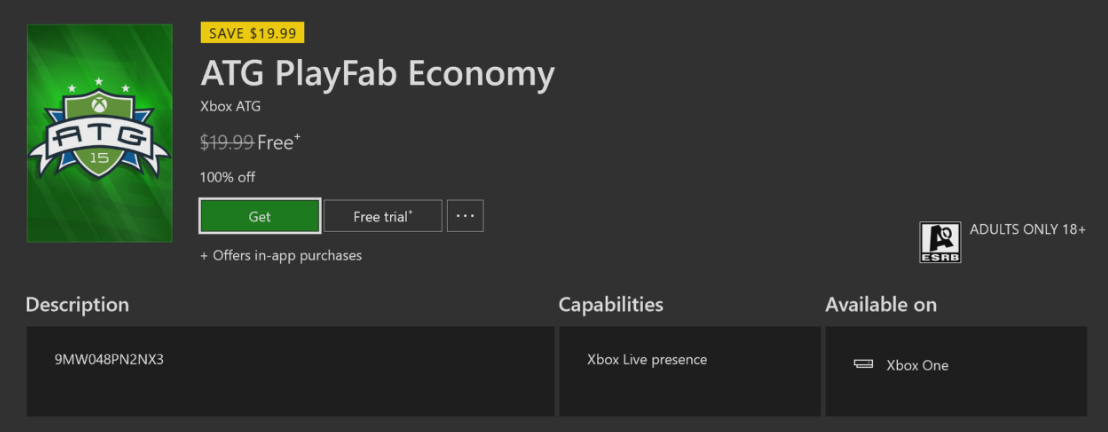


|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 切换物品类别 | 左/右缓冲键 |
| 购买物品 | A 键 |
| 加入购物车（仅限 PlayFab 物品） | X 键 |
| 结账购物车（不为空时） | Y 键 |
| 购物车为空（不为空时） | 按住 Y 键 |
| 消耗箭头 | 左侧扳机键 |
| 消耗药水 | 右侧扳机键 |
| 手动调用 ConsumeXboxEntitlements | 单击左摇杆 |
| 切换主机调试输出 | 查看按键 |

此示例配置为在XDKS.1沙箱中工作。在运行之前，测试帐户必须有权使用已发布产品。

要启动该样本，请在 devkit 上执行以下操作之一：

* 在商店中搜索 ATG PlayFab Economy，或
* 使用 Xbox One XDK 命令提示符：xbapp launch ms-windows-store://pdp/?productid=9NKMD8HL90BN



选择“获取”以获取许可证；无需允许下载完成。

启动后，物品可分为两类。默认类别会显示在 GameManager 的 PlayFab 目录中配置的物品。这些物品的每一个都有 TC 成本，即为此标题配置的虚拟货币。另一类提供配置用于真实货币购买并在“合作伙伴中心”设置的产品，包括 TC 的数量。

(PlayFab)虚拟物品可以单独购买，但也可以添加到购物车中通过一次交易结帐。

有关“Manual ConsumeXboxEntitlements”切换的说明，请参阅[实施说明](#_API_usage)。

**注意：**[合作伙伴中心](https://partner.microsoft.com/) 以前被称为 [开发者中心](https://developer.microsoft.com/) 或 UDC

# 实施说明

必须在样本初始化代码中设置PlayFab标题ID。此示例使用分配给的PlayFab标题ID“4E29” **PlayFabSettings::titleId** 静态场。尽管标题不与任何 Xbox Live 客户端 API 连接，但此标题仍采用全层 Xbox Live 服务在“合作伙伴中心”进行配置。这需要配置“单点登录”，以便可以使用 PlayFab 服务作为依赖方。依赖方业务合作伙伴证书是特定于 PlayFab 标题的，因此，您必须联系 PlayFab 开发技术推广部（[devrel@playfab.com](mailto:devrel@playfab.com)）配置证书。

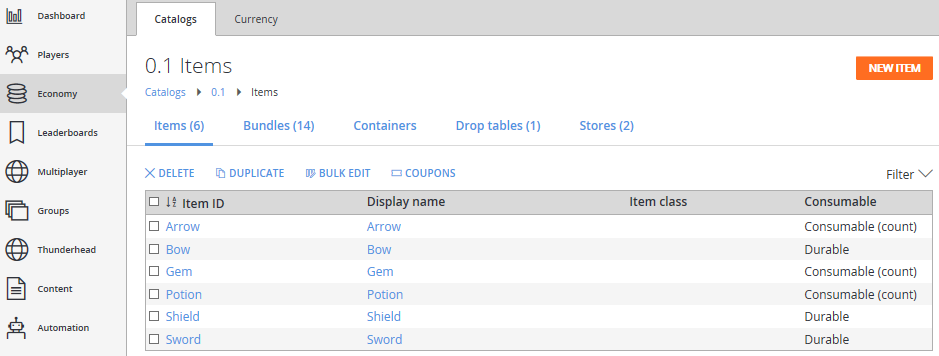
此样本还使用 Windows::Services::Store API 访问商店功能，建议将其用于在“合作伙伴中心”本地设置的产品。因此，它要求帐户有权按上述方式启动。对于从 VS 的调试，还需要 debug\_licensing\_overrides.xml 文件。这已包含在内，并配置有所需 ID。有关覆盖的更多信息可在 [*使用许可证覆盖进行测试*](https://developer.microsoft.com/zh-cn/games/xbox/docs/xdk/testing-license-override)线下的 XDK 文档中找到。

## PlayFab 物品配置

有关如何使用 GameManager 配置物品的完整详细信息，请参阅 PlayFab 官方文档。这可在包含多个子部分的 **Economy** 部分完成，包括“货币”、“物品”、“捆绑包”和“商店”。

货币仅为可通过标题购买、赚取、或授予的虚拟货币。为简单起见，在整个样本中仅使用一种货币 TC。

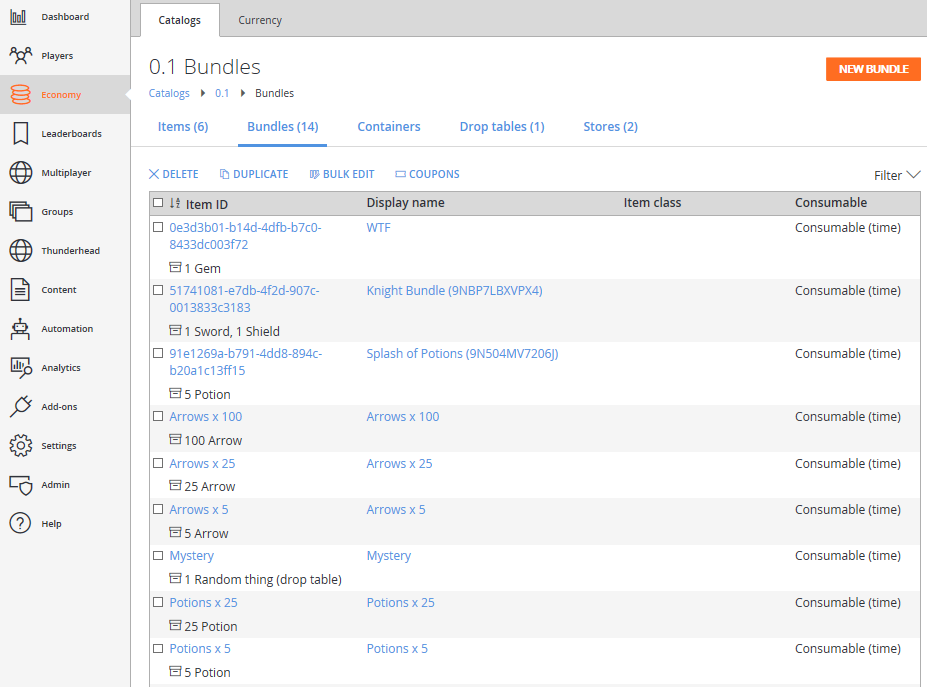
物品是可以授予的原语，可以是消费品或耐用品。



此样本使用的是会在使用时变少的消费品以及永久耐用的物品。这些物品与 Microsoft 产品之间的区别（可在“合作伙伴中心”中配置）如下所示：

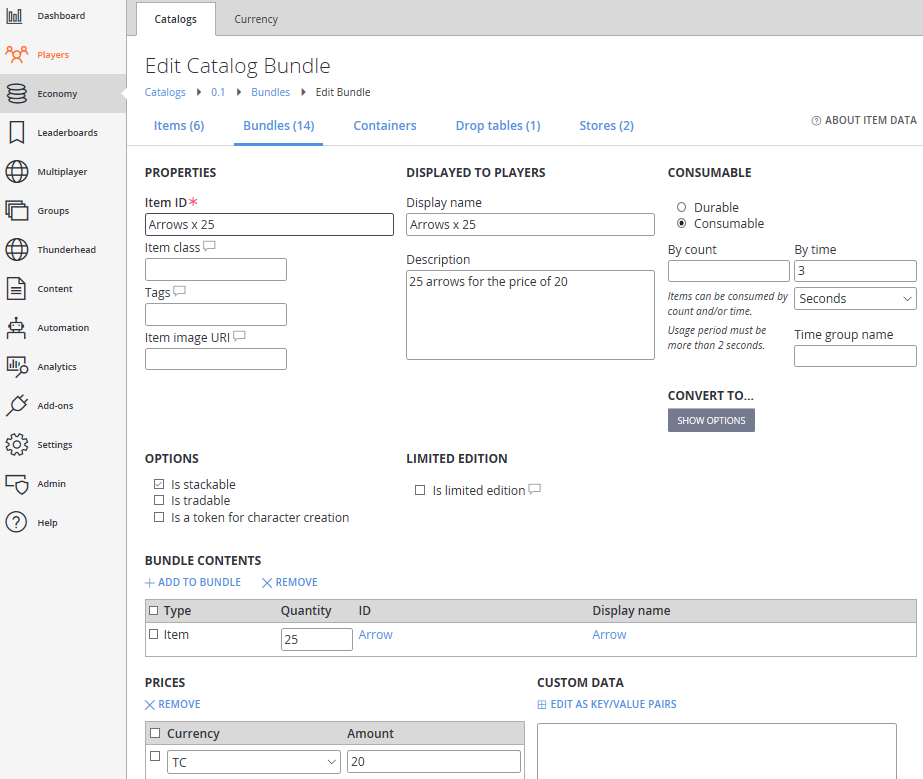
* 消费品也可能有时间限制（设定为在一定时间后过期）。
* 物品作为单独实例在库存中返回。如果不需要，可将其设置为可堆叠；但是，在创建包含大量单个物品的捆绑包时要小心。

在此样本中，PlayFab 类别中列出的物品都是在“捆绑包”选项卡下设置的所有可消费品捆绑包（如下图所示）。这些捆绑包包含各种数量和各种组合的物品和虚拟货币。



虽然 GetUserInventory API 返回所有物品和捆绑包，但是样本会对包含非零数量的捆绑包物品的捆绑消费品进行筛选（请参阅下图中“捆绑包内容”下的“数量”字段）。

另请注意，这些捆绑包本身就是消费品，但它们不是“计数”类型，而是设置为在 3 秒后过期。这样，购买的捆绑包实例也不会在库存中返回，因为很可能只有累积的物品数量才会在库存中看到。



具有 GUID 标识符的其他捆绑包物品是 PlayFab 如何桥接使用真实货币购买的 Microsoft 库存物品以及 GameManager 中配置的虚拟物品。这将在下面的 API 部分中进行描述。

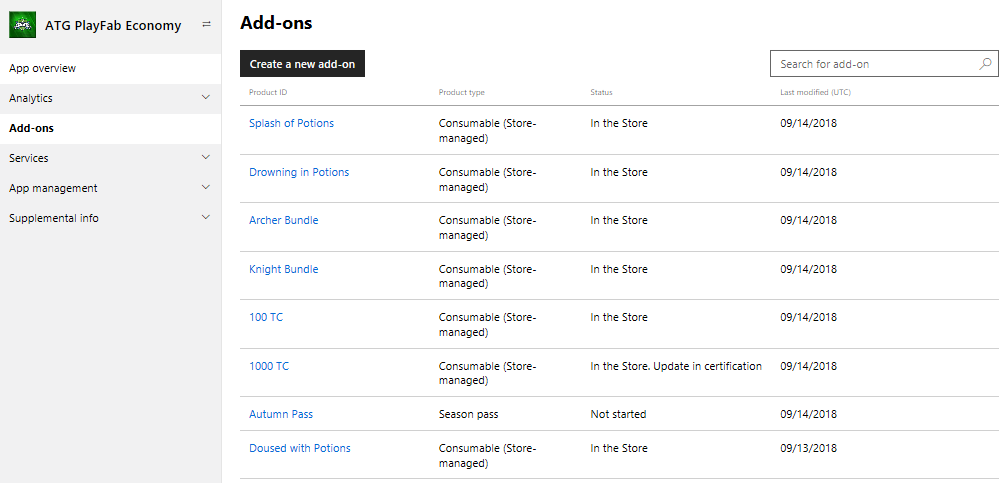
该样本目录采用的其他方面包括：

* 根据玩家分段切换价格的多个商店。在此样本中，定义了一个分段，即如果用户大于 400TC，则某些物品的价格会打折。
* 使用“删除表”配置的物品将根据定义百分比随机授予一个物品。

再次，请参阅 PlayFab 文档，了解有关 Economy 的这些和其他方面的更详细处理。

## Microsoft 物品配置

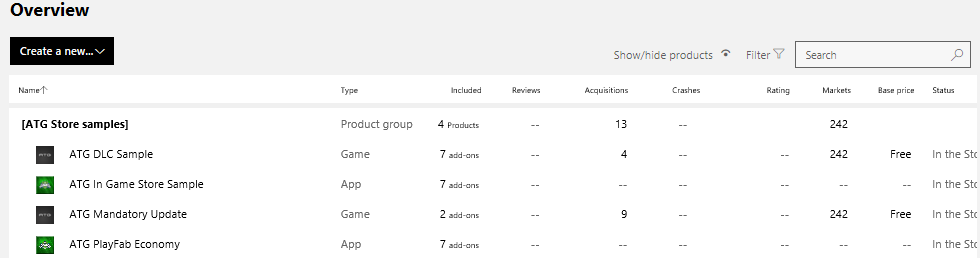
带 GUID 的每个捆绑包物品在“合作伙伴中心中”配置为 1:1 的对应物品。



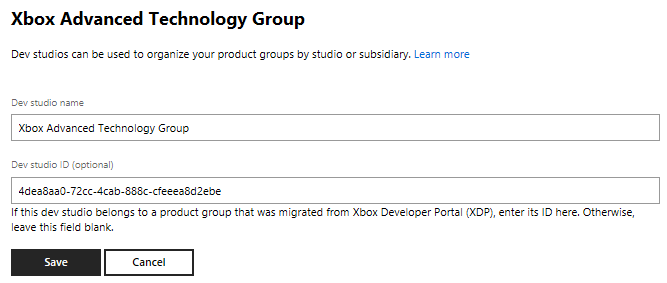
除此之外，这些是不起眼的消费品，可以使用所需属性进行设置。必须对这些消费品进行存储管理才能使 b2b 消耗 API 起作用。这些消费品不必设置为可以看见自己的“商品清单”。强烈建议将每个授权数量设为 1（见下文）。

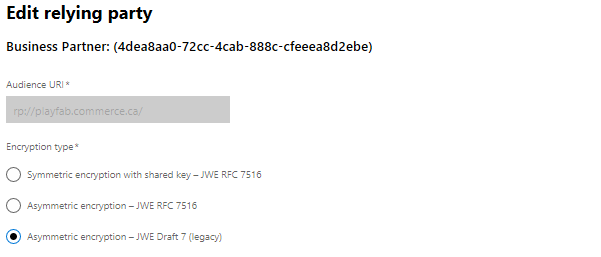
**重要内容!** 要使样本的关键 b2b 方面起作用，有几个方面必须正确配置：

1. “合作伙伴中心”基础游戏产品必须位于产品组中：



此产品组必须与 Dev Studio 相关联，而 Dev Studio 又需要与“依赖方业务合作伙伴 ID”匹配的 Dev Studio ID 相关联。





依赖方、Web 服务和全层 Xbox Live 的配置超出了本自述文件的范围。请联系客户经理和论坛以获取有关这些方面的帮助。

1. 针对通过 ConsumableXboxEntitlements（见下文）执行消费 b2b 的每个产品，必须设置为与传统 Xbox 产品 ID 相关联，这是一个 GUID。这应仅由客户经理通过联系商店研发团队来完成（SMP > Addon > 详细信息 > 旧版 XBox 链接类型：InventoryCallOnly）。

## API 使用情况

以下文档提供了有关此样本中使用的 API 的更多详细信息：

Windows Store API： <https://docs.microsoft.com/zh-cn/uwp/api/Windows.Services.Store.StoreContext>

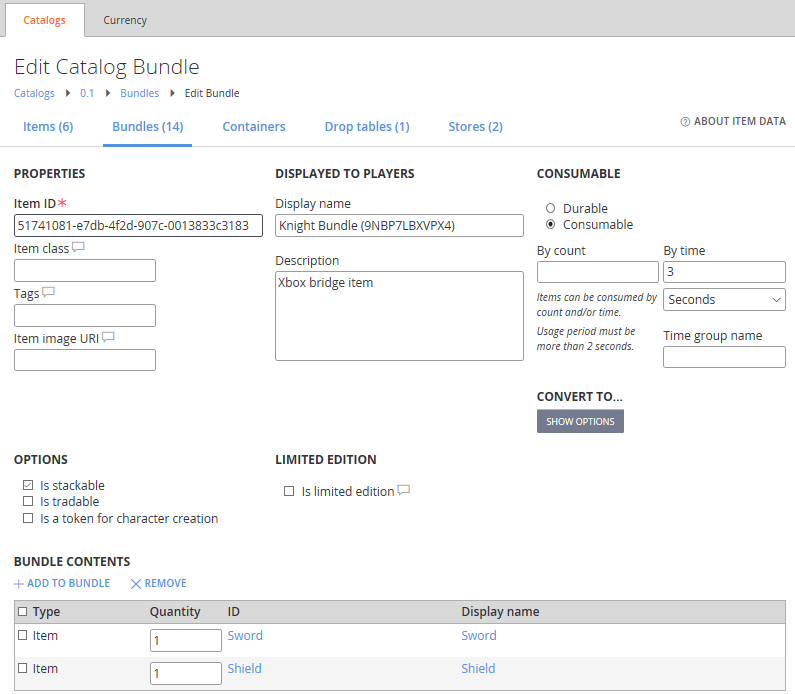
PlayFab API：<https://api.playfab.com/documentation/client>

该样本将这些区域中的操作大致分为两个源文件：分别为 StoreOps.cpp 和 PlayFabOps.cpp。

**ConsumeXboxEntitlements** 是标题必须调用的关键 API，用于协调真实货币购买 Microsoft 产品与相应 PlayFab 物品。调用此 API 时，PlayFab 服务将对 Microsoft 的库存服务执行 b2b 调用，以查询任何拥有非零数量的自有消费品。必须遵循上一节中 b2b 配置步骤才能完成此操作。

由于使用了 Microsoft inventory.xboxlive.com 服务，PlayFab 将会看到使用旧版 Xbox 产品 ID 返回的产品。这些物品将与 PlayFab 目录中配置的任何捆绑包进行比较，这就是为什么上一个图中的物品都配置有 GUID 标识符。

对于与 PlayFab 捆绑包匹配的每个 Microsoft 库存消费品，一次消耗一个未完成的数量，并且授予一个相应的 PlayFab 捆绑包实例。与可购买的捆绑包一样，这些捆绑包也被设置为有时间限制的消费品，捆绑包内容定义了原始真实货币购买所授予的内容。



最终，开发人员应确保“合作伙伴中心”产品的描述与游戏内物品实际授予的内容相匹配。

**注意：**在耗尽之前，Microsoft 消费品会被消耗 1 次，因此，不要创建大量消费品，这非常重要，因为这些消费品会在 Microsoft 和 PlayFab 服务之间产生大量流量，从而限制或禁止标题。

请注意，在正常操作中，ConsumeXboxEntitlements 会在 Microsoft 产品的 RequestPurchaseAsync 成功时被调用。“Manual ConsumeXboxEntitlements”切换将解耦此操作，以便 RequestPurchaseAsync 使 Microsoft 消费品的数量增加，然后**单击左摇杆**，可手动调用 ConsumeXboxEntitlements，以便观察从 Microsoft 数量到 PlayFab 库存的转换。

# 最佳做法

以下是本文档中提到的关键要点的摘要：

* “合作伙伴中心”产品必须位于产品组下，产品组必须与 Dev Studio 相关联，并且 Dev Studio ID 与依赖方业务合作伙伴 ID 相匹配。
* 如果捆绑包实例的数量对用户库存不重要，则将 PlayFab 捆绑包创建为到期时间为 3-5 秒的消费品。
* 对于旨在与 PlayFab 捆绑包关联的 Microsoft 消费品，因为 PlayFab 服务每次仅消耗一个未完成数量，因此每次购买的授予数量应仅为 1。
* 与 Microsoft 产品关联的 PlayFab 捆绑包需要将“物品 ID”设置为 Xbox 产品 ID；要了解这些 ID 是否适用于“合作伙伴中心”中配置的产品，请联系客户经理，以便他们可以在 SMP（内部工具）中进行检查。
* 用于关联的 Microsoft 消费品必须将其“旧版 XBox 链接类型”设置为 InventoryCallOnly。这也可以在 SMP 中查看或设置。

# 更新历史记录

**初始发行：**2018 年 10 月

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。