

本地存储示例

*此示例兼容于 Microsoft 游戏开发工具包（2020 年 6 月）*

# 说明

此示例演示如何在适用于主机和电脑平台的游戏中使用不同的本地存储位置。此外，该示例还演示了一些与本地存储相关的额外行为如何作用于主机。

# 生成示例

如果使用 Xbox One 开发工具包，请将活动解决方案平台设置为 Gaming.Xbox.XboxOne.x64。

如果使用 Xbox Series X|S 开发工具包，请将活动解决方案平台设置为 Gaming.Xbox.Scarlett.x64。

如果使用 Windows 10，请将活动解决方案平台设置为 Gaming.Desktop.x64。

*有关详细信息，请参阅 GDK 文档中的“*运行示例”*。*

# 使用示例

从 Visual Studio 启动示例，并使用示例的交互式按钮运行不同的测试用例。使用示例中的左右箭头按钮切换当前存储位置。可用测试用例根据平台和所选存储类型而异。

电脑：

Text

Description automatically generated

主机：

Text

Description automatically generated

# 实现说明

该示例允许使用不同的本地存储选项测试不同的存储行为。

## 本地存储选项

主机上有多种不同的本地存储选项，电脑上也有许多选项。本示例中介绍了所有主机中的选项。然而，对于电脑，仅介绍了最常见的情况。

存储选项（按平台）：

|  |  |  |
| --- | --- | --- |
| 存储选项 | 主机可用性 | 电脑可用性 |
| 永久性本地存储 (PLS) | **是** | **是\*** |
| 临时驱动器 (T:\) | **是** | 否 |
| 安装的游戏数据 (G:\) | **是** | 否 |
| 系统暂存驱动器 (D:\) | **是** | 否 |
| 临时文件夹 | 否 | **是** |
| LocalAppData 文件夹 | 否 | **是** |

*\*在卸载游戏时，没有专门为游戏预配或管理 PLS 存储空间。*

*注意：有关电脑上内置本地存储位置的更完整的列表，请参阅* [*SHGetKnownFolderPath 函数 (shlobj\_core.h) - Win32 应用 | Microsoft Docs*](https://docs.microsoft.com/windows/win32/api/shlobj_core/nf-shlobj_core-shgetknownfolderpath)*。可在此处找到已知文件夹的列表：*[*KNOWNFOLDERID (Knownfolders.h) - Win32 应用 | Microsoft Docs*](https://docs.microsoft.com/windows/win32/shell/knownfolderid)*。*

**永久性本地存储 (PLS)**

PLS 是用于存储非游戏保存数据的长期存储位置。常见用法可能包括生成的缓存数据、重播存储或用户生成的内容 (UGC) 存储。

主机平台上的 PLS 具有以下特征：

* 特定于游戏：此存储空间只能由请求它的游戏访问。
* 特定于主机：此存储空间始终在特定于主机的 XVD 中创建（即使游戏安装在外部驱动器上），不能在另一台主机上使用。
* 保证的分配：系统确保允许游戏启动之前已经分配了空间。如果硬盘驱动器可用空间不足，系统将提示用户释放空间以便允许游戏运行。
* 用户控制的：用户可以从系统 shell 删除的存储空间。系统从不会自动删除该空间中的任何项。
* 绑定到游戏安装生存期：卸载游戏时，也会删除关联的本地存储空间。如果重新安装游戏，不会还原以前存在的任何数据。
* 可复原和防篡改：该存储空间已加密并已进行完整性检查，所以无法篡改游戏所保存的数据。

但是，电脑上的 PLS 不提供上述保证。PLS 返回的位置是设备上的正常存储位置，如果卸载游戏，则会将其删除。

若要使用 PLS，必须在 MicrosoftGame.config 文件中启用它：

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<Game configVersion=**"0"**>

<PersistentLocalStorage>

<SizeMB>**1024**</SizeMB> <!-- Required Minimum Allocation Size -->

<GrowableToMB>**20480**</GrowableToMB> <!-- Max Growable Size -->

</PersistentLocalStorage>

</Game>

*SizeMB* 参数指定主机平台将确保始终可用于游戏安装的最小分配大小。请记住，电脑不专门管理 PLS 存储空间，因此不保证电脑上的大小可用性。

*GrowableToMB* 参数是可选的，允许指定最大可增长大小。PLS 支持将主机上的分配增长到最大指定大小。虽然 *SizeMB* 是最小保证，但可增长的最大大小可能并不始终可用。如果需要，游戏可以使用 *XPersistentLocalStoragePromptUserForSpaceAsync* 来增大其当前 PLS 大小此操作将提示用户释放硬盘空间，以腾出空间来扩展 PLS。

**临时驱动器 (T:\)**

主机可以使用驱动器号“T”访问特殊临时驱动器此驱动器具有以下具体信息：

* 最大 2GB 存储空间
* 保证在游戏运行且跨暂停/恢复边界时可访问且持久
* 游戏终止后删除

*注意：前 Xbox 开发工具包 (XDK) 行为经常允许临时驱动器在启动过程中持久保留。但是，仍然无法保证持久保留。在 Microsoft 游戏开发工具包 (GDK) 中，PLS 作为永久存储解决方案提供，而临时驱动器并不持久。*

**安装的游戏数据 (G:\)**

特殊驱动器号“G”在主机内部版本上提供，用于访问已安装的游戏数据。此驱动器对于打包的生成是只读的。

*注意：出于开发目的，游戏只能写入已安装的游戏数据以进行松散生成。但是，不建议这样做，因为其他本地存储选项可能更合适。*

**系统暂存驱动器 (D:\)**

Devkit 具有对驱动器号为“D”的系统暂存驱动器的特殊访问权限。此仅限开发的存储位置允许编写开发所需的任何内容。常见的用法可能是开发人员日志、故障转储或其他非零售用途数据。

系统暂存驱动器是永久性的，没有写入限制，可以由任何游戏访问。它直接写入硬盘驱动器，因此可以占用包安装所需的空间。

开发电脑也可以通过网络路径“\\[DevkitIP]\SystemScratch”直接访问此驱动器或在右键单击 Xbox One 管理器工具中的开发工具包时使用“浏览主机文件”。

**临时文件夹**

在 Windows 电脑上，许多应用程序使用 TEMP 文件夹来存储临时数据。此位置由当前用户的 TMP/TEMP 环境变量确定。

若要获取 TEMP 文件夹，请使用 [GetTempPath](https://docs.microsoft.com/windows/win32/api/fileapi/nf-fileapi-gettemppatha) 方法。

**LocalAppData 文件夹**

Windows 电脑上的 LocalAppData 文件夹通常由应用程序用来存储应用程序安装中未包含的每用户永久性应用程序数据。

若要获取 LocalAppData 文件夹，请使用 [SHGetKnownFolderPath](https://docs.microsoft.com/windows/win32/api/shlobj_core/nf-shlobj_core-shgetknownfolderpath) 方法，并使用 FOLDERID\_LocalAppData 参数。有关可用的已知文件夹的信息记录在 [KNOWNFOLDERID (Knownfolders.h) - Win32 应用 | Microsoft Docs](https://docs.microsoft.com/windows/win32/shell/knownfolderid) 中。

## 测试用例

可以为每个存储类型运行几种不同的测试用例。可用测试根据平台和存储类型而变化。

测试用例：

|  |  |
| --- | --- |
| 测试用例 | 说明 |
| 获取写入统计数据 | *仅限主机：*报告游戏向跟踪的驱动器（PLS 或临时驱动器）的写入行为。 |
| 获取存储信息 | 报告有关当前所选存储类型的一些基本信息，例如要使用的路径或根位置的文件/文件夹数。 |
| 查询可用空间 | 查询并报告当前所选存储类型有多少可用空间。 |
| 写入文件 | 尝试使用当前选定的存储类型写入文件。 |
| 读取文件 | 尝试使用当前选定的存储类型读取文件。 |
| 获取 PLS 信息 | *仅限主机：*返回有关 PLS 存储分区的信息，包括所用空间、可用空间、最大值等。 |
| 填充可用 PLS | *仅限主机：*通过 [XR-133](#XR133) 以允许的最大速率填充 PLS 的压力测试。 |
| 压力写入统计数据 | 仅限主机：一种压力测试，可通过 [XR-133](#XR133) 以允许的最大速率将 2GB 数据写入当前所选存储类型。 |

**获取写入统计数据测试用例**

此仅限主机的测试使用 *XPackageGetWriteStats* 报告游戏的写入行为。*XPackageGetWriteStats* 返回以下信息：

|  |  |
| --- | --- |
| 统计数据 | 说明 |
| 间隔 | 当前间隔的总毫秒数。 |
| 预算 | 允许在当前间隔内写入的、不会导致事件超出和 [XR-133](#XR133) 失败的总字节数。 |
| 写入的字节 | 在当前间隔内写入的总字节数。 |
| 流逝的时间 | 当前间隔中已流逝多少毫秒。 |

只跟踪写入临时驱动器 (T:\) 或 PLS 的写入。在零售版中，这是主机的唯一可写入本地存储位置。

*注意：前 1GB 的写入数据未被跟踪到定时间隔。这是有意设计的，使游戏在需要时更轻松地在前面突发写入。写入前 1GB 后，将开始新的间隔，且所有将来的间隔都会计时。*

**获取存储信息测试用例**

此简单的测试用例报告当前存储位置的根路径，以及根路径中存在的文件和文件夹数。

**查询可用空间测试用例**

此测试使用 [GetDiskFreeSpaceEx](https://docs.microsoft.com/windows/win32/api/fileapi/nf-fileapi-getdiskfreespaceexa) 计算存储位置的可用空间。报表将包括可用字节数、驱动器总大小以及磁盘上的可用字节总数。

**写入文件测试用例**

此测试尝试将文件写入当前选定的本地存储位置。在大多数情况下会成功。但是，在某些情况下可能会失败：

* 如果 PLS 已满，则尝试写入 PLS
  + 仅对于主机，如果 PLS 已满，但有可用的可增长空间，则将调用释放空间的提示。
* 尝试写入包生成上已安装的游戏数据时
* 在没有可用空间的情况下尝试写入

**读取文件测试用例**

此测试尝试读取之前使用上述写入测试编写的文件。

**获取 PLS 信息测试用例**

此仅限主机的测试使用 *XPersistentLocalStorageGetSpaceInfo* 获取有关当前 PLS 分配的信息。虽然可以在电脑上调用此功能，但不会在电脑上显式管理空间。因此，建议在电脑上使用 [GetDiskFreeSpaceEx](https://docs.microsoft.com/windows/win32/api/fileapi/nf-fileapi-getdiskfreespaceexa)。

*XPersistentLocalStorageGetSpaceInfo* 返回的数据如下所示：

|  |  |
| --- | --- |
| 数据 | 说明 |
| 可用的免费字节数 | 可立即写入 PLS 的字节数。 |
| 可用字节总数 | 可根据 MicrosoftGame.config 中指定的最大大小写入 PLS 的剩余空间总量。可能需要通过 *XPersistentLocalStoragePromptUserForSpaceAsync* 使此数据可用。 |
| 使用的字节数 | PLS 中当前使用的字节数。 |
| 总字节数 | PLS 配置的总大小。 |

**填充可用 PLS 测试用例**

此仅限主机的测试可检查 PLS 中有多少可用数据，并可写入数据，直到数据填满。写入的速率为固定速度，等于 [XR-133](#XR133) 允许的最大值。

此测试的预期用法是，轻松地填充可用的空闲字节，以允许测试可增长 PLS。但是，可用的空闲字节通常与空闲字节总数匹配，因此很难测试 *XPersistentLocalStoragePromptUserForSpaceAsync*。若要正确测试释放空间的提示，应使用应用程序或其他数据填充 devkit 的硬盘驱动器。然后，可用的空闲字节应报告小于总可用字节数的值。此时，可以填充可用数据，可以测试 *XPersistentLocalStoragePromptUserForSpaceAsync*。

**压力写入统计数据测试用例**

此仅限主机的测试将 2GB 的数据写入当前选定的本地存储位置。每隔几秒钟就会查询写入统计数据并将其报告到屏幕日志。

此测试的目的是展示如何跟踪写入统计数据，以便更好地了解限制行为。

**XR-133：本地存储写入限制**

XR-133 要求在滑动的 5 分钟窗口期中将不超过 1GB 的数据写入主机的硬盘驱动器。这包括使用 PLS 和临时存储驱动器 (T:\) 进行写入。

上面的两个压力测试以约 3.41MB/s 的恒定速率进行写入。但是，5 分钟窗口期允许高速突发 IO，只要管理按间隔写入的总数据即可。

此外，游戏生存期内写入的前 1GB 的数据不会被跟踪到 5 分钟窗口期。写入前 1GB 后，使用 *XPackageGetWriteStats* 跟踪的所有未来间隔都会显示计时间隔信息。

# 更新历史记录

**初始版本：**Microsoft 游戏开发工具包（2021 年 10 月）

# 隐私声明

在编译和运行示例时，将向 Microsoft 发送示例可执行文件的文件名以帮助跟踪示例使用情况。若要选择退出此数据收集，你可以删除 Main.cpp 中标记为“示例使用遥测”的代码块。

有关 Microsoft 的一般隐私策略的详细信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/en-us/privacystatement/)。