

智能交付

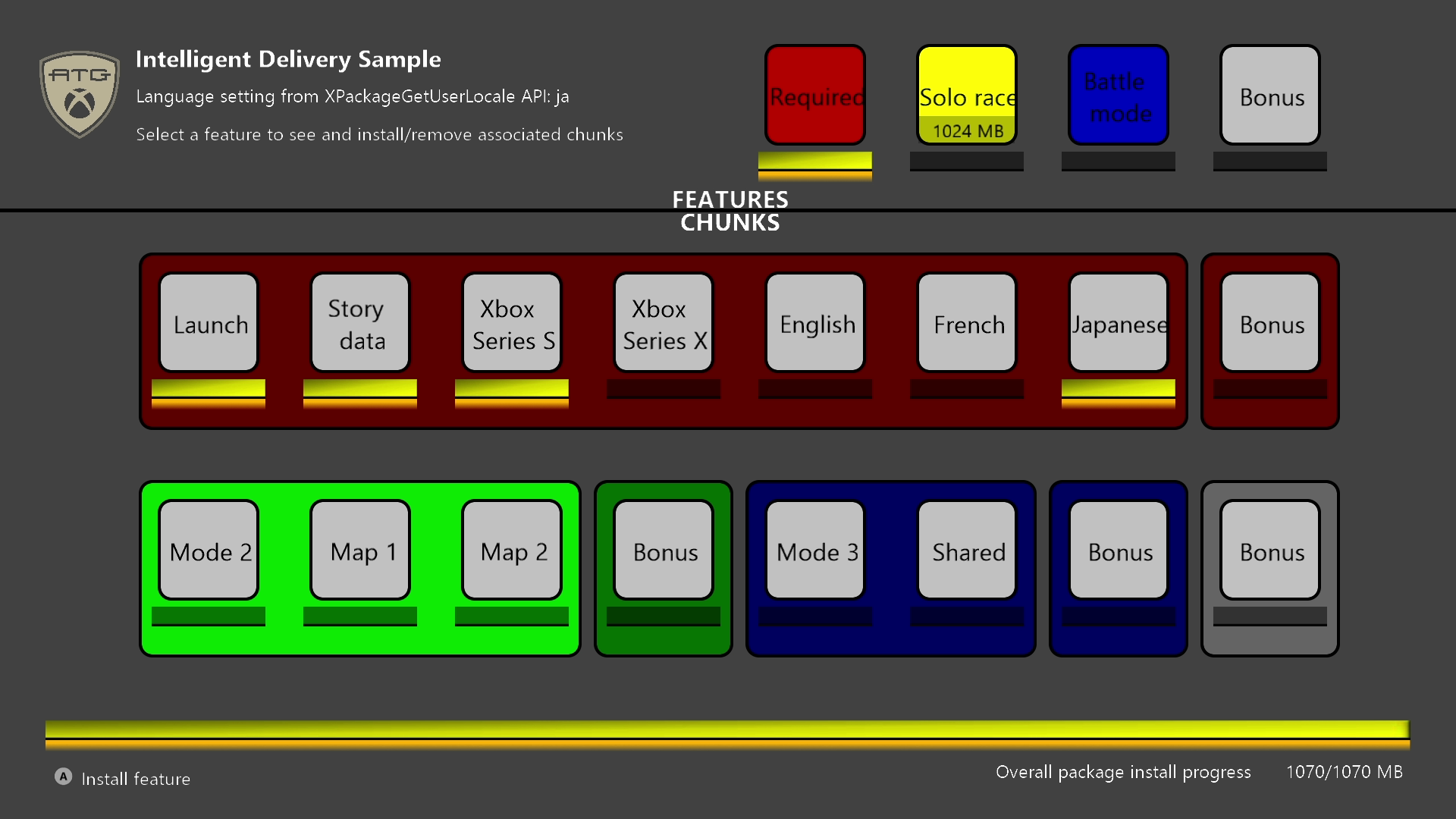
*(之前称为流式安装)*

*此示例兼容于 Microsoft 游戏开发工具包(2020 年 8 月)*

# 说明

此示例演示内容为**智能交付** API。安装启动区块后，可以运行该示例以直观显示其余区块的安装进度。

示例还演示了智能安装功能，包括功能和配方、本地化区块、以及特定于硬件的区块。根据目标硬件（Xbox Series Family 的 Lockhart/Anaconda，Xbox One Family 的 Durango/Xbox One X）的目标 SKU 和设备的语言/区域设置，将安装不同的内容。用户将能够在示例运行时在多项功能中选择安装和删除，并观察安装和进度结果。



此示例基于早期的*流式安装*示例构建，用以演示配方和功能。引用布局文件 (Chunks\_[XboxOne/Scarlett].xml)：

<Recipes>

<Recipe Id="default" IncludedFeatures="feature1" />

</Recipes>

<Features>

<Feature Id="feature1" Hidden="true" Tags="mode1" />

<Feature Id="feature2" DisplayName="Mode 2" Tags="mode2;maps"/>

<Feature Id="feature3" DisplayName="Mode 3" Tags="mode3"/>

<Feature Id="feature4" DisplayName="Bonus Content" Tags="bonus"/>

</Features>

这个程序包将使用默认的活动配方构建，而这个配方将只安装 feature1。一共可以安装四个功能，在网格中通常用不同的颜色项来表示。安装每个功能后其标签将标记为活动，并将安装带有这些标签的区块。在示例中选择功能后，则将安装或删除该功能相关联的区块（如功能规范所标记的那些区块）。

所涉到的一些其他复杂性有：

<Feature Id="feature2" DisplayName="Mode 2" Tags="mode2;maps"/>

feature2 将安装标记为 mode2 或 maps 的区块。

<Chunk Id="1007" Tags="mode1#bonus">

<FileGroup DestinationPath="\" SourcePath="." Include="filler256mb7.txt"/>

</Chunk>

如果 mode1 和 bonus 标签**都**处于活动状态，则将仅安装此区块，即安装 feature1 **和** feature4。以下是示例中的所有 “Bonus” 项目，因为每个功能都有一项，而且 feature4 单独有一项。

<Chunk Id="3001" Tags="mode3;mode2">

<FileGroup DestinationPath="\" SourcePath="." Include="filler256mb13.txt"/>

</Chunk>

如果 mode1 **或** mode2 处于活动状态，则将安装此区块；也就是说，如果安装了 feature2 或 feature3（以先安装者为准），则安装此区块。反之，如果将这两个功能都删除，则将只能删除此区块。这一点在示例中表示为分组到“Battle Mode”功能中的“Shared”项。

对于**电脑**来说，因为没有设备特定区块，功能和区块略有修改。相反，feature1 中有一个依赖于 feature4 的区块。

# 构建示例

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

如果使用电脑，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

和大多数 GDK 示例不同，这个*智能交付*示例不是在 Visual Studio 中构建和部署的。虽然可以直接从 Visual Studio 部署和运行，但是这样会避开此示例原本要演示的安装工具和过程。以下是使用此示例的步骤：

1. 在 Visual Studio 中构建示例，**但不要**部署和执行
2. 生成安装包内容
3. 创建安装包
4. 安装这个包并运行示例

此示例需要每个主机设备系列使用单独的安装包。提供了不同的 .bat 和 .config 文件，用于为每个设备构建适当的安装包。无论使用哪个平台，Xpackage 代码和 GDK 保持不变。在 *MicrosoftGame.config* 文件中，唯一构成区别的属性是 TargetDeviceFamily，这个值可能是 XboxOne、Scarlett、或 PC。

第一步只需从 Visual Studio 构建示例。此操作将创建最终用于创建程序包的二进制和目录布局。构建示例后，从命令行运行 **CreateInstallPackage\_[XboxOne/Scarlett/PC].bat**。这个批处理文件将创建几个大文件作为内容文件来实时监视安装进度，并通过 TargetDeviceFamily 集为对应的 MicrosoftGame.config 分配位置。

下一步是使用与 GDK 一起安装的 **makepkg** 工具创建安装包。打开 **Xbox Gaming 命令提示符**，导航到含有 *IntelligentDelivery.vcxproj* 的目录。 可以从打开的命令提示符运行 **CreateXVC\_[XboxOne/Scarlett].bat** 或 **CreateMSIXVC\_PC.bat**，或者直接运行命令：

Xbox One 设备系列：

makepkg.exe pack /v /f Chunks\_XboxOne.xml /d ".\Gaming.Xbox.x64\Layout\Image\Loose" /pd ".\Gaming.Xbox.x64\Layout\Image"

Scarlett 设备系列：

makepkg.exe pack /v /f Chunks\_Scarlett.xml /d ".\Gaming.Scarlett.x64\Layout\Image\Loose" /pd ".\Gaming.Scarlett.x64\Layout\Image"

电脑：

makepkg.exe pack /pc /v /f Chunks\_PC.xml /d .\Gaming.Desktop.x64\Layout\Image\Loose /pd .\Gaming.Desktop.x64\Layout\Image

注意，如果直接运行命令，请参考 .bat 以确保所有文件分配到正确位置，特别是一些已更新的文件。

由于程序包里已经含有几个 GB 的填充内容，可能需要几分钟时间来执行。完成执行后，在此案例中，将在 **/pd** 选项， **“.\Gaming.[Xbox/Scarlett/PC].x64\Layout\Image”** 目录后的指定位置创建几个文件。将命名实际安装包文件，而且名称中包含派生自 *MicrosoftGame.config* 的安装包的**安装包系列名**。如果是主机，则将创建一个 **.xvc** 文件。文件后缀名根据 TargetDeviceFamily 确定：若是 Xbox One，则为 **\_x**，若是 Scarlett，则为 **\_xs**。如果是电脑，则创建一个 **.msixvc** 文件。

最后一步是将包安装到设备上。对于 **Xbox** 设备，需要设置几个选项，以根据想要测试的方案来确定想要安装的内容。如果想要测试流式安装实现，则可以使用以下命令：

xbapp install [/l] [/w] [/i] <package name>

|  |  |
| --- | --- |
| 选项 | 详细信息 |
| <包名> | 根据 MakePkg 命令创建的安装包。 |
| /l | 指定只应该安装启动区块。安装启动区块后，可以运行游戏，但只有布局文件里标记为 “Launch” 区块中枚举的内容可用。 |
| /w | 指定安装将保持管道打开状态。此操作允许测试功能的动态安装，以及开始阶段未安装的额外的语言区块。 |
| /i | 指定交互模式。这将显示管理安装系统 UI，该 UI 显示交互式安装和删除时包中定义的可用功能。 |
| <完整的包名> | 安装到主机上之后的游戏名称。此名称不应与包系列名相混淆。可通过在命令提示符下运行 **xbapp list** 命令在主机上获取已安装包的列表。 |

使用 /l 和 /w，可以测试安装游戏的不同场景。只使用 /l，系统将只安装初始区块。使用 /l 和 /w，系统将安装初始区块，其它区块将在启动应用程序后安装。只使用 /w，游戏将根据布局文件中定义的活动配方安装全部基础包，并保持管道打开。如果不使用这些参数，则只安装基础包，而且不可能执行示例的其他安装。

/w 最类似于零售行为，其开放管道模拟了通常在 CDN 上托管的包源可用性。

根据用户系统和语言自动安装一些区块。例如，如果部署至带有语言设置为 “en-US” 的 Scarlett 系列包的 Scarlett 主机，将不会安装标记为其他语言或设带有设备 “Xbox-Lockhart” 标记的区块。如果使用 Durango 系列包，将基于目标设备安装 Durango 或 Scorpio 内容（或在开发人员设备上调试设置）。

可以通过一些可传递到命令的可选参数使用 xbapp 安装 强制安装其他区块。

|  |  |
| --- | --- |
| 选项 | 详细信息 |
| /语言 | 指定哪些语言将安装在以分号分隔的语言列表中安装哪些语言。这可以用于测试多种语言的本地化内容，而无需在更改主机设置后重新运行安装命令。 |
| /设备 | 指定应安装哪些硬件特定区块。这是要包括的以分号分隔的不同硬件 SKU 列表。 |
| /配方 | 指定应安装哪些配方。这是在布局文件中定义的以分号分隔的配方说明符列表。 |
| /功能 | 指定应安装哪些功能。这是布局文件中定义的以分号分隔的功能说明符列表。 |
| /AllChunks | 指定应安装所有区块，无论主机硬件或设置如何。 |

这个列表列出了使用 **xbapp 安装**命令时可以使用的选项，但不是一个详尽的列表。有关选项的综合列表，请参阅 GDK 文档。

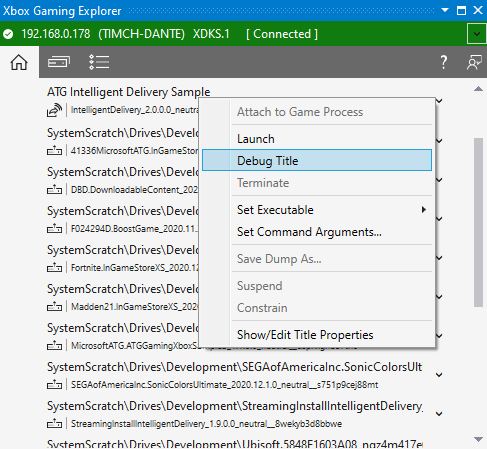
对于**电脑**，将改为使用 wdapp 命令。只有 /l 标记可用于安装启动区块。由于示例在托管包源的同一设备上运行，因此该示例将能够执行智能交付操作，而无需流程处于活动状态：

wdapp 安装 [/l] <包名>

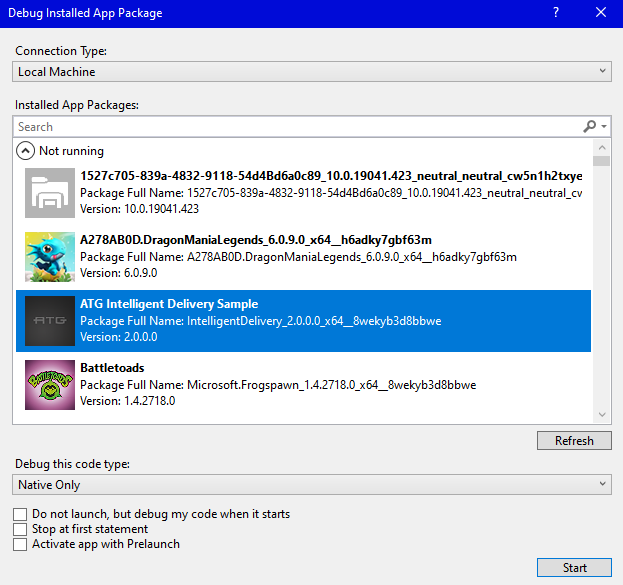
# 调试示例

由于此示例通常不能仅通过从 Visual Studio 启动来在调试器中运行，因此必须使用其他方法调试打包版本。

在 **Xbox** 上，使用 Xbox Gaming Explorer启动示例调试会话，该会话应位于已注册标题列表中：



在**电脑**上，使用“调试”>“其他调试目标”>“调试已安装的应用包”：



此示例使用以下控件：

|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 游戏板控制器 | 鼠标/键盘 | |
| 导航网格菜单 | D-pad | 箭头键 |
| 从网格菜单中选择一个项目。  安装功能（如果可以安装）。  安装语言区块（如果可以安装）。  如果多个项挂起，请首先安装该项。  卸载当前项目（如果已安装且可能删除）。 | A | 左键单击/输入 |
| 枚举所有功能和区块，并显示其 XPackageAvakAvailability 值 (在调试文本中) | Y | Y 键 |
| 切换屏幕调试文本 | 菜单 | 波形符 ( ` ) |
| 退出示例 | 查看 | Esc |

# 更新历史记录

2019 年 4 月首次发布示例。

2020 年 3 月更新示例，以单独配置 XboxOne 和 Scarlett 设备系列。

2021 年 2 月将示例重命名为 IntelligentDelivery，更新配方和功能支持

2021 年 3 月将 PC 添加为受支持的平台

# 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“Sample Usage Telemetry”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/)。