

ダウンロード可能なコンテンツのサンプル

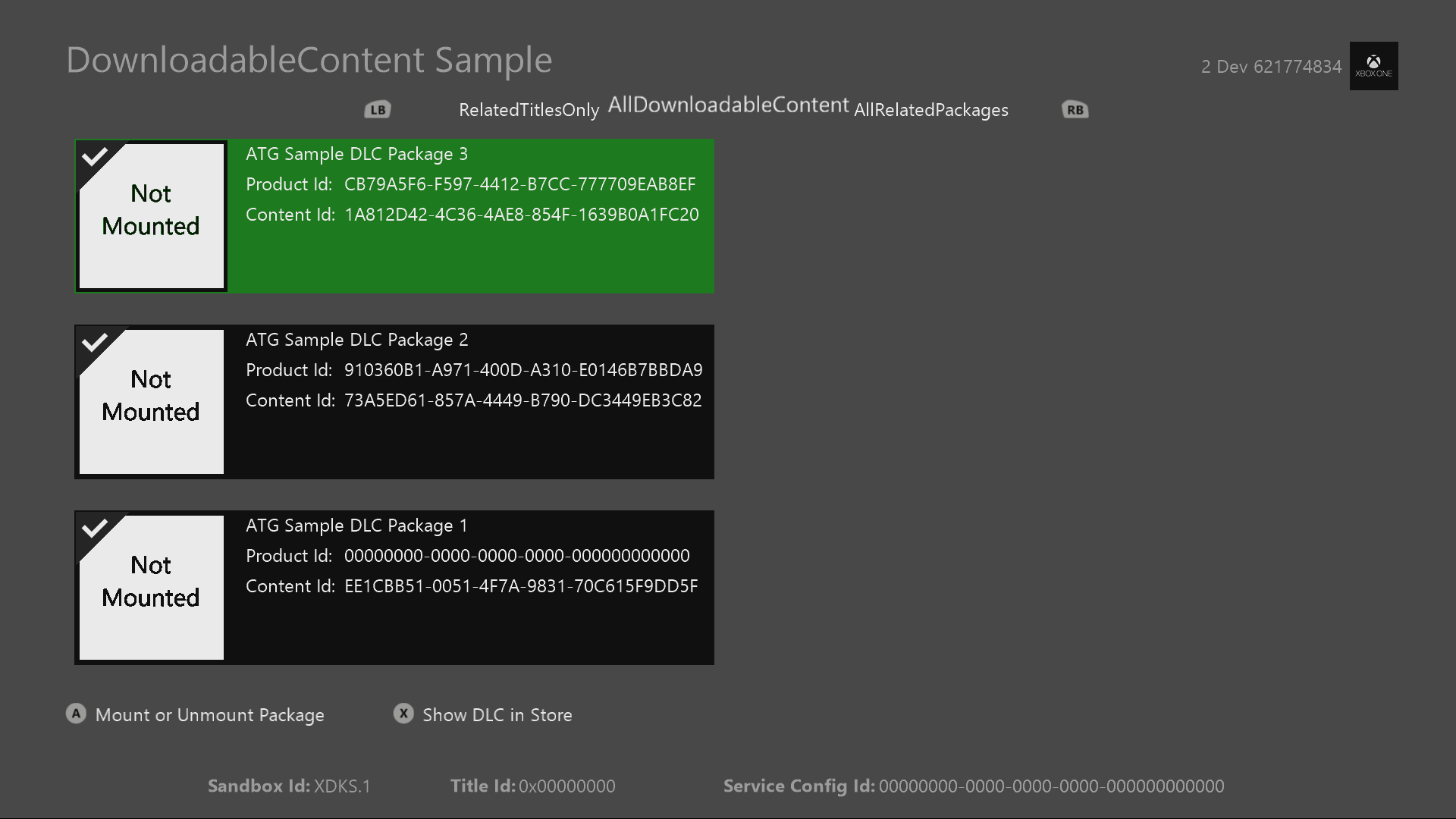
*\* このサンプルは、2016 年 3 月の Xbox One XDK と互換性があります。*

# 概要

このサンプルでは、Xbox One のタイトル内でダウンロード可能なコンテンツの列挙と読み込みを実装する方法を示します。

# サンプルの使用

このサンプルは、XDKS.1 サンドボックスで機能するように構成されています。



|  |  |
| --- | --- |
| 作用 | ゲームパッド |
| DLC パッケージを選択 | 上下D-Pad |
| パッケージのマウントまたはマウント解除 | Aボタン |
| DLC 製品の Marketplace にアクセスする | X ボタン |
| サインインします。 | メニューボタン |
| 終了 | 表示ボタン |

# 実装上の注意

このバージョンの XDK では、このサンプルでは Xbox One 本体に XDKS.1 **SandboxID**を使用する必要があります。この **SandboxID**に切り替えるには Xbox One XDK コマンドプロンプトで、次のように入力します。

xbconfig sandboxid=XDKS.1

# ダウンロード可能なコンテンツ パッケージの作成と展開

タイトル内に DLC として読み込むコンテンツ パッケージを作成するには、パッケージ化する必要がある生ファイルをディレクトリ構造内に配置してから、昇格した Xbox One XDK コマンド プロンプトから次のツールを実行します。

makepkg.exe pack /lt /v /f "./Chunks.xml" /d ./[RawFilesFolder] /pd ./[DestinationOutputDirectory]

簡単な Chunks.xml、appxmanifest.xml、およびパッケージの作成方法の例は、サンプルに付属の SamplePackages.zip ファイルにあります。 appxmanifest.xml はルーズ ファイルのディレクトリに含まれており、コンテンツに合わせて適切に更新する必要があります。 パッケージを作成したら、同じコマンド プロンプトから次のコマンドを使用して、コンソールに展開できます。

xbapp install [DestinationOutputDirectory]\[package name]

ゲーム内でのダウンロード テストのためにコンテンツを Marketplace に公開する準備ができたら、makepkg コマンドに /lt タグの代わりに /l タグを含めて、ライセンスされて暗号化されたバージョンを作成します。 また、必要な送信フォームとプロセスについては、開発者アカウント マネージャーにお問い合わせください。 /l で作成されたパッケージは、そのパッケージの ContentID が、既にサポートされていて Marketplae Catalog に取り込まれている ContentID と一致してない限り、xbapp install を使用して手動でインストールすることはできません。 このため、Marketplace に転送する前に、テスト コンテンツに /lt を使用することをお勧めします。

DLC のデバッグと列挙  
列挙がタイトル パッケージと DLC パッケージで正しく動作するためには、パッケージの appxmaifest.xml 内にタイトルの ProductID と一致している **AllowedProductID** が含まれている必要があります。 Visual Studio を介してデバッグする場合、または生ファイルを開発キットに展開する場合、タイトルの既定の ProductID はすべて 0 になります。タイトルをデバッグしながら、AllowedProductID に実際のタイトルが含まれる DLC セットを列挙する機能を有効にするには、**RelatedProduct** 定義を、タイトルの実際の ProductID の Package.appxmanifest に追加します。 これによって、タイトルの ProductID と、それに関連付けられているすべての DLC が、タイトルのデバッグ バージョンに関連するものとしてマークされます。 その後、**AllDownloadableContentPackages** フラグを付けて DLC を列挙する場合、すべてが 0 の ProductID に関連付けられたパッケージだけではなく、実際のタイトルの DLC の結果が得られます。以下は、RelatedProduct を定義する方法の例です。また、サンプルの Package.appxmanifest も参照してください (提供されている ID はサンプルの ProductID です)。

<Applications>

…

</Applications>

<Extensions>

<mx:PackageExtension Category="xbox.store">

<mx:XboxStore>

<mx:RelatedProducts>

<mx:RelatedProduct Id="eabe6ccb-48f2-4cd2-832c-1d9753bd707b" />

</mx:RelatedProducts>

</mx:XboxStore>

</mx:PackageExtension>

</Extensions>

# DLC パッケージの列挙とマウント

ローカルにインストールされた DLC パッケージを列挙するには、次の API を使用します。

IVectorView< IDownloadableContentPackage^ >^

IDownloadableContentPackageManager::FindPackages(InstalledPackagesFilter *filter*)

これは、渡されたフィルターと一致する、現在インストールされている DLC パッケージを定義するオブジェクトのベクトルを提供します。返されたオブジェクトのいずれかを操作する場合、タイトルでそのオブジェクトに対するあらゆるパッケージ操作としてその特定のオブジェクトへの参照を保持する必要があります。同じパッケージの他の列挙結果とは同期されません。 例：パッケージ 1 をマウントすると、後続の列挙呼び出しで返された同じパッケージに対する **IDownloadableContentPackage** には、パッケージが現在マウントされていることが反映されません。

DLC パッケージをマウントしてパッケージのファイルにアクセスするには、次の API を使用します。

Platform::String^ IDownloadableContentPackage::Mount()

これにより、まずパッケージのマウントが許可されていることを確認し、パッケージが現在のユーザーまたはコンソールの範囲内でライセンスされている場合は、パッケージを仮想ドライブにマウントします。 このドライブの場所は文字列として返され、パスを作成して、パッケージの内容を開くために使用できます。 パッケージがマウントされると、パッケージがコンソールから削除されないように、システムでいくつかのファイル ハンドルを取得して、開いたままにします。 マウントされたパッケージは、タイトルが制限モードまたは一時停止モードのときも同じ仮想ドライブにマウントされたままです。 タイトルが閉じられているか、終了しているか、**Unmount** が呼び出された場合のみ、マウントされたパッケージが解放されます。

注:   
システム全体で一度にマウントできる DLC パッケージの数には制限があります。 現在のところ、ERA のタイトルでは一度に 35 個以下のパッケージをマウントすることをお勧めします。 使用していないときにパッケージを解放して、他のパッケージ用にスロットを解放する必要があります。 この数は現時点での推奨事項であり、変更される可能性があります。 これがタイトルの問題である場合は、開発者アカウント マネージャーに相談してください。

ライセンスの範囲が変更され、タイトルで現在マウントされている DLC をアンロードする場合は、次の API を使用してください。

void IDownloadableContentPackage::Unmount()

これにより、仮想ドライブからパッケージがマウント解除され、システムでファイルに対して持っているハンドルがあれば解放されます。 **Unmount** を呼び出す前に、タイトルで開かれているファイルのハンドルをパッケージに解放することが引き続き想定されています。 パッケージがマウント解除されると、ユーザーのコンソールから削除できます。

新しいパッケージがシステムにインストールされると、**DownloadableContentPackageManager::DownloadbleContentPackageInstallCompleted** イベント ハンドラーに登録されているすべてのアプリに通知が送信されます。 タイトルでこのイベントが表示されたら、もう一度 **FindPackages** を使用して、その範囲内に新しいパッケージがあるかどうかを確認する必要があります。 現在、このイベントはパッケージがインストールされたときに 2 回発生するので、パッケージごとに 2 つのイベント通知を受け取ります。 これは、将来の Marketplace のシナリオで複数のインストール パッケージを展開するために必要でした。 このサンプルでは、このイベントでパッケージを再列挙する方法と、マウントされたパッケージおよびマウント解除されたパッケージを適切に処理する方法の例を示します。

# DLC パッケージのライセンス

タイトルでこの API を使用して、パッケージの現在のライセンス ステータスを検証できます。

bool IDownloadableContentPackage::CheckLicense ( bool \*isTrial )

この API は、現在のサインイン ユーザーまたはコンソールに付与されている権利の範囲に対して、このパッケージをマウントするライセンスがある場合は、ブール値を返します。 また、ブール値を渡してその戻り値を確認することによって、このパッケージで利用可能なライセンスがプレビュー ライセンスかどうかを確認することもできます。

ユーザーが別のコンソールでサインインまたはサインアウトするか、ゲーム セッション中に DLC の Xbox One Marketplace 内で権利を購入すると、パッケージのライセンスが変更される可能性があります。 パッケージに有効なライセンスがあり、マウントされているものの、現在は有効なライセンスがない場合は、DLC パッケージをいつマウント解除するか、マウント解除することが適切かどうかはタイトル次第です。 例えば：ユーザー 1 が DLC Map Pack を所有しており、ユーザー 2 のコンソールでユーザー 2 とプレイしているとします。 DLC で新しいマップの 1 つを再生している間に、ユーザー 1 がサインアウトします。 ユーザー 2 はコンテンツを所有しておらず、コンソールにはそれに対する権利がないため、コンソール上のパッケージのライセンスは無効なライセンスに変更されました。 このゲームでは、ゲーム セッションが終了するまで、またはラウンドが終了して DLC をマウント解除するまで、ユーザー 2 が DLC でプレイすることが引き続き許可される可能性があります。 また、このゲームでは、ユーザー 2 が自分のアカウントで DLC を購入し、Xbox One Store の [Product] でそのアカウントを指定して、DLC のプレイを継続するように案内およびアップセルすることがあります。

タイトルで **LicenseTerminated** イベントの **TypedEventHandler** を作成し、パッケージのオブジェクトの **LicenseTerminated** プロパティに追加できます。 このイベントは、マウントされたパッケージのライセンスが無効になり、最初にマウントされたときの所有権の範囲に含まれなくなったときに発生します。 パッケージがマウント解除されていて、メモリ リークを避けるためにパッケージ オブジェクトがタイトルの範囲から解放される前に **LicenseTerminated** プロパティからパッケージを削除する必要がある場合、このイベントは発生しません。 このサンプルでは、パッケージのマウント時およびマウント解除時にこのイベントを正しく登録、管理、および削除する方法を示します。

# 更新履歴

**初回リリース：**2017 年 8 月

# プライバシーステートメント

サンプルをコンパイルして実行すると、サンプルの実行可能ファイルの名前がMicrosoftに送信され、サンプルの使用状況の追跡に役立ちます。このデータ収集を無効にするには、Main.cppの「Sample Usage Telemetry」というラベルの付いたコードブロックを削除します。

マイクロソフトのプライバシーポリシー全般に関する詳細については、 [Microsoftのプライバシーステートメント](https://privacy.microsoft.com/ja-jp/privacystatement/)をご参照ください。