

지능형 전달

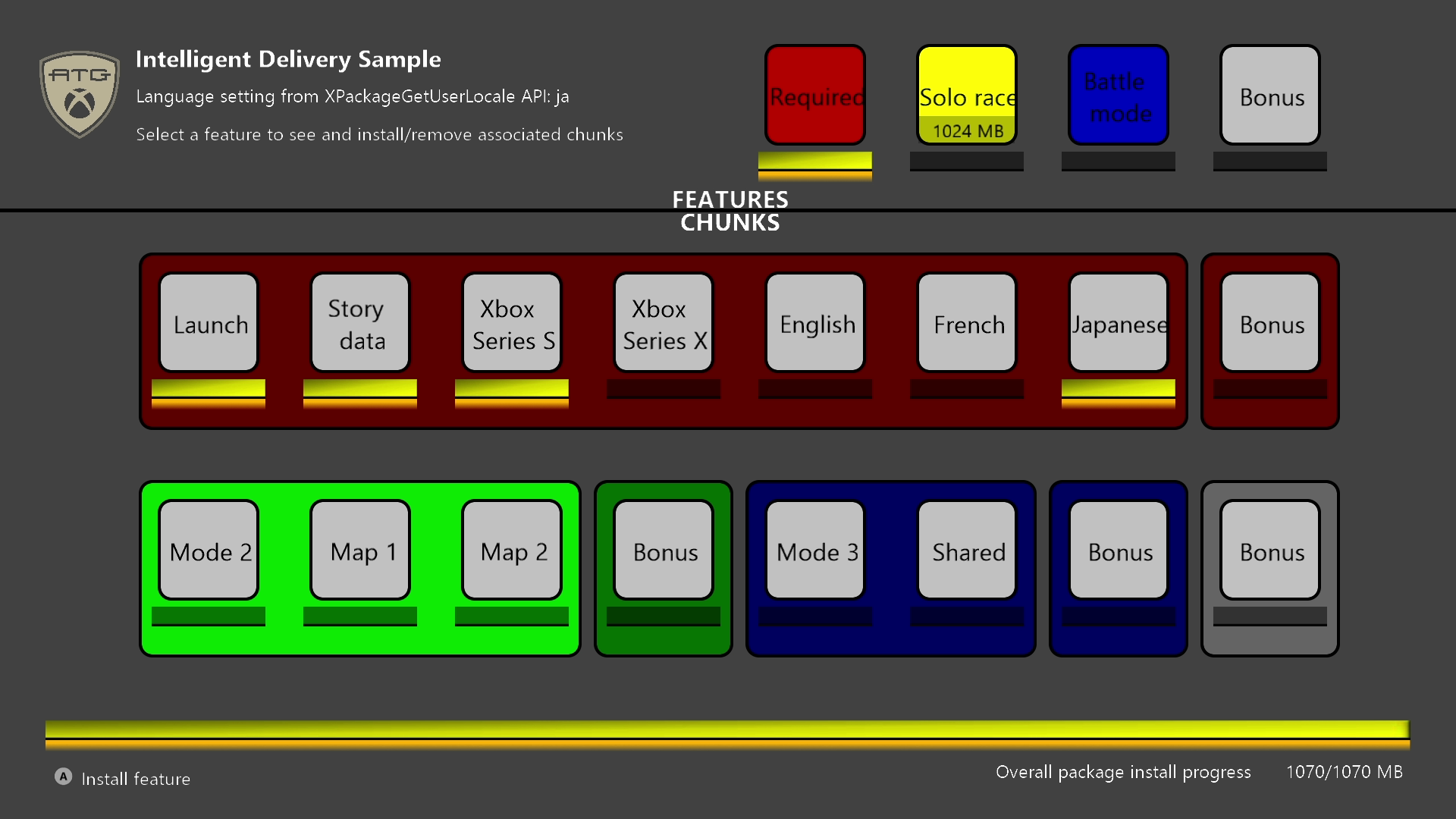
*(이전의 StreamingInstall)*

*이 샘플은 Microsoft 게임 개발 키트와 호환 가능(2020년 8월)*

# 설명

이 샘플에서는 **지능형 전달** API를 보여 줍니다. 시작 청크가 설치되면 이 샘플을 실행하여 나머지 청크의 설치 진행률을 시각화할 수 있습니다.

또한 기능 및 레시피, 지역화된 청크 및 하드웨어 특정 청크를 비롯한 지능형 설치 기능을 보여 줍니다. 대상 하드웨어의 대상 SKU(Xbox Series 제품군의 경우 Lockhart/Anaconda, Xbox One 제품군의 경우 Durango/Xbox One X) 및 장치의 언어/로캘 설정에 따라 다른 콘텐츠가 설치됩니다. 그러면 샘플이 실행되는 동안 사용자는 설치 및 제거할 여러 기능 중에서 선택하여 설치 및 진행 상황을 관찰할 수 있습니다.



이 샘플은 레시피와 기능을 보여주기 위해 이전 *StreamingInstall* 샘플을 기반으로 합니다. 레이아웃 파일 참조(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>

이 패키지는 feature1만 설치하는 기본 활성 레시피로 빌드됩니다. 총 네 가지 기능을 설치할 수 있으며 일반적으로 그리드에 다양한 색으로 표시됩니다. 각 기능을 설치하면 태그가 활성으로 표시되고 해당 태그가 있는 청크가 설치됩니다. 샘플에서 기능을 선택하면 해당 기능과 연결된 청크가 기능 사양에 따라 태그로 지정되어 설치 또는 제거됩니다.

다음과 같이 몇 가지 추가 복잡성이 포함됩니다.

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

feature2는 mode2 또는 맵으로 태그가 지정된 청크를 설치합니다.

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

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

</Chunk>

이 청크는 mode1 및 bonus 태그가 **모두** 활성인 경우만 설치됩니다. 즉, feature1 **및** feature4가 설치됩니다. 이는 feature4에 대한 독립 실행형으로써 샘플뿐만 아니라 각 기능에 하나씩 있는 샘플에 표시된 모든 “보너스” 항목입니다.

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

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

</Chunk>

이 청크는 mode1 **또는** mode2 태그가 활성인 경우 설치됩니다. 즉, 이 청크는 feature2 또는 feature3 중 먼저 설치되어 있는 경우에 설치됩니다. 반대로, 두 기능 모두 제거된 경우에만 제거할 수 있습니다. 이는 샘플의 "Battle Mode" 기능에서 그룹화된 "Shared" 항목으로 표시됩니다.

**PC**의 경우 장치별 청크가 없으므로 기능 및 청크가 약간 수정됩니다. 대신, feature4에 따라 feature1에 청크가 있습니다.

# 샘플 빌드

Scarlett 개발 키트를 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Xbox.Scarlett.x64로 설정합니다.

Xbox One 개발 키트를 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Xbox.XboxOne.x64로 설정합니다.

PC를 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Desktop.x64로 설정합니다.

*자세한 내용은 GDK 설명서에서* 샘플 실행을 *참조하세요.*

# 샘플 사용

대부분의 GDK 샘플과는 달리 *IntelligentDelivery* 샘플은 Visual Studio에서 빌드하고 배포할 수 없습니다. Visual Studio에서 직접 배포하고 실행할 수는 있으나 이는 원래 이 샘플에서 보여 주려고 의도한 설치 도구 및 프로세스를 우회합니다. 이 샘플을 사용하는 단계는 다음과 같습니다.

1. 배포 및 실행 **없이** Visual Studio에서 샘플 빌드
2. 설치 패키지용 콘텐츠 생성
3. 설치 패키지 만들기
4. 패키지 설치 및 샘플 실행

이 샘플에서는 각 콘솔 장치 제품군에 대해 별도의 패키지가 필요합니다. 각 장치에 적합한 패키지를 빌드하는 데 사용하도록 서로 다른 .bat 및 .config 파일이 제공됩니다. 플랫폼에 관계없이 XPackage 코드는 GDK에서 변경되지 않습니다. *MicrosoftGame.config* 파일에서의 유일한 차별화 특성은 TargetDeviceFamily로 XboxOne, Scarlett 또는 PC입니다.

첫 번째 단계는 Visual Studio에서 샘플을 빌드하는 것입니다. 이렇게 하면 패키지를 만드는 데 사용될 이진 및 디렉터리 레이아웃이 만들어집니다. 샘플이 만들어지면 명령줄에서 **CreateInstallPackage\_[XboxOne/Scarlett/PC].bat**를 실행합니다. 배치 파일은 여러 개의 대용량 파일을 만들어 실시간으로 설치 진행률을 모니터링하고 TargetDeviceFamily 집합을 사용하여 적절한 MicrosoftGame.config를 준비합니다.

다음 단계는 GDK와 함께 설치된 **makepkg** 도구를 사용하여 설치 패키지를 만드는 것입니다. **Xbox 게임 명령 프롬프트**를 열고*IntelligentDelivery.vcxproj*가 포함된 디렉터리로 이동합니다. 열어 놓은 명령 프롬프트에서 **CreateXVC\_[XboxOne/Scarlett].bat** 또는 **CreateMSIXVC\_PC.bat**를 실행하거나 직접 명령을 실행할 수 있습니다.

Xbox One 장치 제품군:

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

Scarlett 장치 제품군:

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

PC:

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

명령을 직접 실행하는 경우(특히 일부가 업데이트된 경우) .bat를 참조하여 모든 파일이 올바른 위치에 준비되어 있는지 확인합니다.

패키지가 여러 기가바이트의 필러 콘텐츠로 채워졌기 때문에 실행하는 데 몇 분 정도 걸릴 수 있습니다. 실행이 완료되면 **/pd** 옵션 다음에 지정된 위치에 여러 파일이 만들어지는 데 이 경우, **“.\Gaming.[Xbox/Scarlett/PC].x64\Layout\Image”** 디렉터리가 만들어집니다. 실제 설치 패키지 파일에는 *MicrosoftGame.config*에서 파생된 패키지의 **패키지 제품군 이름**이 지정됩니다.콘솔의 경우 TargetDeviceFamily에 따라 **.xvc** 접미사가 있는 파일이 만들어집니다: Xbox One의 경우 **\_x**, Scarlett의 경우 **\_xs**. PC의 경우 **.msixvc** 파일이 만들어집니다.

마지막 단계는 장치에 패키지를 설치하는 것입니다. **Xbox**의 경우, 테스트하려는 시나리오에 따라 설치할 콘텐츠를 결정하는 몇 가지 옵션이 있습니다. 스트리밍 설치 구현을 테스트하려는 경우 다음 명령을 사용할 수 있습니다.

xbapp install [/l] [/w] [/i] <패키지 이름>

|  |  |
| --- | --- |
| 옵션 | 세부 정보 |
| <패키지 이름> | MakePkg 명령으로 만든 설치 패키지입니다. |
| /l | 시작 청크만 설치하도록 지정합니다. 시작 청크가 설치되면 게임을 실행할 수 있지만 레이아웃 파일에서 "Launch"로 표시된 청크에 열거된 콘텐츠만 사용할 수 있습니다. |
| /w | 설치 시 파이프를 열린 상태로 유지하도록 지정합니다. 이렇게 하면 처음에 설치되지 않은 기능 및 추가 언어 청크의 동적 설치를 테스트할 수 있습니다. |
| /i | 대화형 모드를 지정합니다. 그러면 대화형 설치 및 제거를 위해 패키지에 정의된 사용 가능한 기능을 표시하는 설치 시스템 관리 UI가 표시됩니다. |
| <전체 패키지 이름> | 콘솔에 설치된 후의 게임 이름입니다. 패키지 제품군 이름과 혼동하지 말아야 합니다. 명령 프롬프트에서 **xbapp list** 명령을 실행하여 콘솔에 설치된 패키지 목록을 얻을 수 있습니다. |

/l 및 /w를 사용하여 다른 게임 설치 시나리오를 테스트할 수 있습니다. /l만 사용하면 시스템에 초기 청크만 설치됩니다. /l 및 /w를 사용하면 초기 청크는 시스템에 의해 설치되고 나머지는 응용 프로그램을 시작한 후에 설치됩니다. /w만 사용하면 이 게임은 레이아웃 파일에 정의된 활성 레시피를 기반으로 전체 기본 패키지를 설치하고 파이프를 열어 둡니다. 이러한 매개 변수를 사용하지 않으면 간단한 기본 패키지가 설치되며 샘플로 추가 설치를 수행할 수 없습니다.

/w는 일반적으로 CDN에서 호스팅되는 패키지 소스의 가용성을 모방하는 개방형 파이프를 사용하는 소매 동작과 가장 유사합니다.

일부 청크는 사용자의 시스템 및 언어를 기반으로 자동 설치됩니다. 예를 들어 "en-US"로 설정된 Scarlett 제품군 패키지를 사용하여 Scarlett 콘솔에 배포하는 경우 다른 언어 또는 장치 "Xbox-Lockhart"로 태그가 지정된 청크는 설치되지 않습니다. Durango 제품군 패키지를 사용하면 대상 장치(또는 개발 키트에서 디버그 설정)를 기반으로 Durango 또는 Scorpio 콘텐츠가 설치됩니다.

명령에 전달할 수 있는 일부 선택적 인수가 있는 xbapp install을 사용하여 다른 청크를 강제로 설치할 수 있습니다.

|  |  |
| --- | --- |
| 옵션 | 세부 정보 |
| /Languages | 세미콜론으로 구분된 언어 목록에 설치할 언어를 지정합니다. 콘솔 설정을 변경한 후 설치 명령을 다시 실행하지 않고도 여러 언어의 지역화된 콘텐츠를 테스트하는 데 사용할 수 있습니다. |
| /Devices | 설치할 하드웨어 특정 청크를 지정합니다. 포함할 여러 하드웨어 SKU의 세미콜론으로 구분된 목록입니다. |
| /Recipes | 설치할 레시피를 지정합니다. 레이아웃 파일에 정의된 레시피 지정자의 세미콜론으로 구분된 목록입니다. |
| /Features | 설치할 기능을 지정합니다. 레이아웃 파일에 정의된 기능 지정자의 세미콜론으로 구분된 목록입니다. |
| /AllChunks | 콘솔 하드웨어 또는 설정에 관계없이 모든 청크를 설치하도록 지정합니다. |

이 목록은 **xbapp 설치** 명령과 함께 사용할 수 있는 옵션의 전체 목록이 아닙니다. 포괄적인 옵션 목록은 GDK 설명서를 참조하세요.

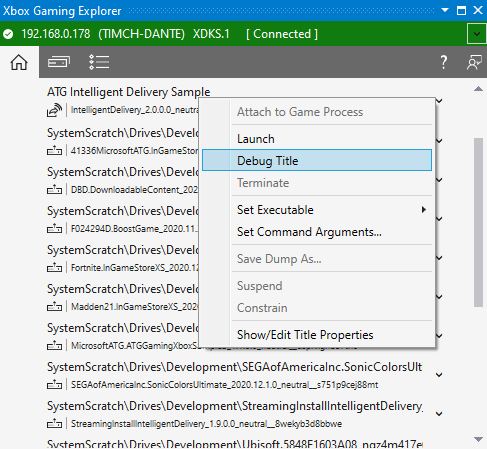
**PC**의 경우 wdapp 명령이 대신 사용됩니다. /l 플래그만 시작 청크를 설치하는 데 사용할 수 있습니다. 샘플이 패키지 원본을 호스트하는 동일한 장치에서 실행 중이기 때문에 샘플은 활성 프로세스 없이 지능형 전달 작업을 수행할 수 있습니다.

wdapp install [/l] <패키지 이름>

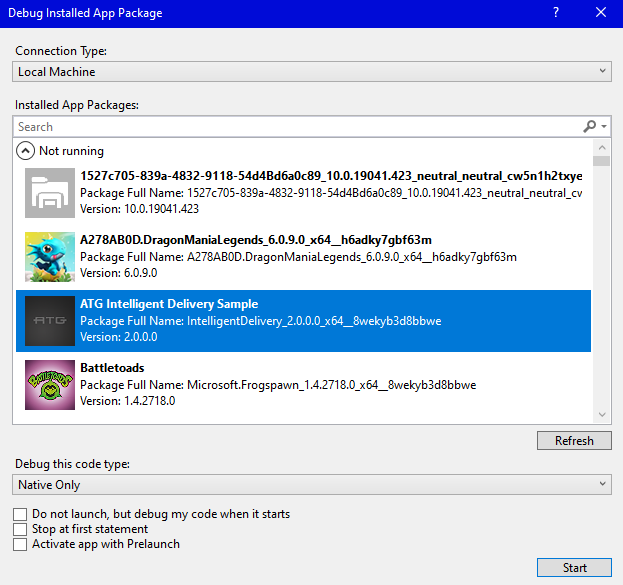
# 샘플 디버깅

일반적으로 이 샘플은 Visual Studio에서 시작하여 디버거에서 실행할 수 없으므로 패키지된 빌드를 디버그하려면 다른 방법을 사용해야 합니다.

**Xbox**에서는 Xbox 게임 탐색기를 사용하여 등록된 타이틀 목록에 있는 샘플의 디버깅 세션을 시작합니다.



**PC**에서 디버그 > 기타 디버그 대상 > 설치된 앱 패키지 디버그를 사용합니다.



이 샘플에서는 다음 컨트롤을 사용합니다.

|  |  |  |  |
| --- | --- | --- | --- |
| 조치 | 게임 패드 컨트롤러 | 마우스/키보드 | |
| 그리드 메뉴 탐색 | 방향 패드 | 화살표 키 |
| 그리드 메뉴에서 항목을 선택합니다.  설치 가능한 경우 기능을 설치합니다.  설치 가능한 경우 언어 청크를 설치합니다.  여러 항목이 보류 중인 경우 먼저 항목을 설치합니다.  설치되어 있고 제거 가능한 경우 현재 항목을 제거합니다. | A | 왼쪽 클릭 / Enter 키 |
| 모든 기능 및 청크를 열거하고 XPackageChunkAvailability 값 표시(디버그 텍스트) | Y | Y 키 |
| 화면 디버그 텍스트 토글 | 메뉴 | 물결표( ~ ) |
| 샘플 종료 | 보기 | Esc 키 |

# 업데이트 기록

2019년 4월 샘플 첫 번째 릴리스.

2020년 3월 XboxOne 및 Scarlett 장치 제품군의 별도 구성을 위한 샘플 업데이트.

2021년 2월 샘플 이름을 IntelligentDelivery로 변경하고 레시피 및 기능 지원 개선

2021년 3월 지원되는 플랫폼으로 PC 추가

# 개인정보처리방침

샘플을 컴파일하고 실행하는 경우 샘플 사용량을 추적할 수 있도록 샘플 실행 파일의 파일 이름이 Microsoft에 전송됩니다. 이 데이터 수집을 옵트아웃하려면 "샘플 사용량 원격 분석"으로 레이블이 지정된 Main.cpp에서 코드 블록을 제거할 수 있습니다.

Microsoft의 개인정보 정책에 대한 자세한 내용은 [Microsoft 개인정보처리방침](https://privacy.microsoft.com/en-us/privacystatement/)을 참조하세요.