シンプルな動的リソースのサンプル

*このサンプルは Microsoft Game Development Kit (2021 年 4 月) と互換性があります*

# 説明

このサンプルでは、HLSL シェーダー モデル 6.6 で HLSL 動的リソースを使用する方法を示します。HLSL で ResourceDescriptorHeap[] と SamplerDescriptorHeap[] を使用して、ヒープを介してリソースに直接アクセスすること以外は、 SimpleTexture と機能的には同じです。



# サンプルのビルド

Xbox One 開発キットを使用している場合は、アクティブ ソリューション プラットフォームを Gaming.Xbox.XboxOne.x64 に設定します。

Xbox Series X|S を使用している場合は、アクティブ ソリューション プラットフォームを Gaming.Xbox.Scarlett.x64 に設定します。

PC 用のビルド (Gaming.Desktop.x64) では、HLSL SM 6.6 機能が使用されているため、[DirectX Agility SDK](https://devblogs.microsoft.com/directx/gettingstarted-dx12agility/) が必要となります。Agility SDK は、サンプル内には NuGet パッケージとして含まれています。また、Microsoft.Windows.SDK.CPP NuGet パッケージを使用して、DXC.exe コンパイラの最新の Windows SDK (22000) バージョンを取得します。開発者は、最新の DXC を直接 [Github](https://github.com/microsoft/DirectXShaderCompiler/releases) から使用することも可能です。

*詳細については、GDK のドキュメントの*「サンプルの実行」*を参照してください。*

# サンプルの使用方法

このサンプルには、終了以外のコントロールはありません。

# 実装上の注意

このサンプルでは、SimpleTexture からほぼすべてのコードを借ります。唯一の違いは、リソースへのアクセスです。

このサンプルでは、バインドされたリソースをルート署名から削除し、HLSL シェーダー コード内の ResourceDescriptorHeap[] および SamplerDescriptorHeap[] アクセスに置き換えます。これには、SetGraphicsRootSignature() の前に SetDescriptorHeaps() が呼び出されていることを確認し、フラグ CBV\_SRV\_UAV\_HEAP\_DIRECTLY\_INDEXED および SAMPLER\_HEAP\_DIRECTLY\_INDEXED をルート署名に追加する必要があります。

HLSL 6.6 動的リソースの詳細については、「[HLSL SM 6.6 動的リソース](https://microsoft.github.io/DirectX-Specs/d3d/HLSL_SM_6_6_DynamicResources.html)」を参照してください。

動的リソースの高度な使用方法については、 Graphics\VisibilityBuffer サンプルを参照してください。

# 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場合、サンプルの使用状況を追跡するために、サンプル実行ファイルのファイル名が Microsoft に送信されます。このデータ コレクションからオプトアウトするには、Main.cpp の「Sample Usage Telemetry」というラベルの付いたコードのブロックを削除します。

全般的な Microsoft のプライバシー ポリシーの詳細については、「[Microsoft プライバシー ステートメント](https://privacy.microsoft.com/en-us/privacystatement/)」を参照してください。