xtexconv Sample

*This sample is compatible with the August 2016 Xbox One XDK and Windows 8.1 SDK Spring 2015.*

# Description

This sample is a PC side command-line tool which extends the standard TexConv command-line tool for texture conversion and preparation to support Xbox One Exclusive App offline texture tiling for use with the CreatePlacement / CreatePlacedResourceX APIs.

This tool accepts a wide range of image formats--such as Windows Imaging Component supported codecs .jpg, .png, .tiff, .bmp, and HD Photo/JPEG XR plus TargaTruevision .tga files, RGBE .hdr, and OpenEXR .exr files--as well as .dds--as an input texture format. It supports full mip chain generation using a user specified filter, as well as supporting texture arrays, cube maps, cube map arrays, and volume maps.

Running the tool without any parameters shows the help screen, as follows:



# Usage

The XTexConv tool supports the same set of command-line parameters and syntax as the standard TexConv tool. Detailed documentation is available on [GitHub](https://github.com/Microsoft/DirectXTex/wiki/Texconv).

It includes an additional switch ‘**-xbox**’ which causes the output DDS file to contain Xbox One tiled texture data and the ‘XBOX’ DDS file variant. The tile mode used is determined by XGComputeOptimalTileMode. These offline prepared Xbox One textures are assumed to be used with XG\_BIND\_SHADER\_RESOURCE.

If a ‘XBOX’ DDS file variant is used as an input file, it is automatically Detiled before further processing, allowing the tool to be used to convert an ‘XBOX’ DDS file to a standard DDS file.

The tool also supports a switch `**-xgmode**` to select the hardware version for tiling.

Use `-xgmode:xboxonex` to set the preferring tiling for Xbox One X. Otherwise it defaults to Xbox One / Xbox One S.

# DDS files for Xbox One

The standard DDS file format is described on [MSDN](https://msdn.microsoft.com/en-us/library/bb943991.aspx). The ‘XBOX’ DDS file variant is similar to the ‘DX10’ header extension. The ‘XBOX’ DDS files are laid out as:

DWORD dwMagic

DDS\_HEADER header

DDS\_HEADER\_XBOX

{

DXGI\_FORMAT dxgiFormat;

uint32\_t resourceDimension;

uint32\_t miscFlag; // see DDS\_RESOURCE\_MISC\_FLAG

uint32\_t arraySize;

uint32\_t miscFlags2; // see DDS\_MISC\_FLAGS2

uint32\_t tileMode; // see XG\_TILE\_MODE

uint32\_t baseAlignment;

uint32\_t dataSize;

uint32\_t xdkVer; // matching \_XDK\_VER

} headerXbox

<Remainder of file is a tiled texture binary layout suitable for use with CreatePlacement APIs>

Example code for loading and creating textures from ‘XBOX’ variant DDS files is available in the *DirectX Tool Kit* ([DX11](https://github.com/Microsoft/DirectXTK) / [DX 12](https://github.com/Microsoft/DirectXTK12)) in the XboxDDSTextureLoader ([DX11](https://github.com/Microsoft/DirectXTK/wiki/XboxDDSTextureLoader) / [DX 12](https://github.com/Microsoft/DirectXTK12/wiki/XboxDDSTextureLoader)) module.

# DirectXTex for Xbox One

# XTexConv is a slightly modified version of TexConv, with additional capabilities added to the [DirectXTex](https://github.com/Microsoft/DirectXTex/) library. The standard version of [TexConv](https://github.com/Microsoft/DirectXTex/wiki/Texconv) and DirectXTex are available on GitHub.

# The Xbox One auxiliary functionality for DirectXTex (in DirectXTexXbox.h in the Xbox C++ namespace) includes:

# XboxImage: a container for tiled texture data

* Functions for storing and loading the XBOX variant of DDS files
  + GetMetadataFromDDSMemory
  + GetMetadataFromDDSFile
  + LoadFromDDSMemory
  + LoadFromDDSFile
  + SaveToDDSMemory
  + SaveToDDSFile
* Functions for performing tiling of standard linear data to Xbox One tiled textures, as well as the reverse operation:
  + Tile
  + Detile
* Functions for creating texture resources from tiled Xbox One tiled images with Direct3D 11 extensions
  + CreateTexture
  + CreateShaderResourceView
  + FreeTextureMemory
* Functions for creating texture resources from tiled Xbox One tiled images with Direct3D 12 extensions
  + CreateTexture
  + FreeTextureMemory

# Dependencies

This tool and the DirectXTex for Xbox One auxiliary Tile/Detile functions require the XG.DLL (located in the Xbox One XDK under the bin folder) to be located in the standard DLL search paths.

# OpenEXR support

The xtexconv tool uses the [OpenEXR](http://www.openexr.com/) library, making use of the NuGet Packages for [openexr](https://www.nuget.org/packages/openexr-msvc14-x64/) and [zlib](https://www.nuget.org/packages/zlib/) which are subject to their own licensing terms. This support can be disabled by undefining USE\_OPENEXR, deleting DirectXTexEXR.\*, and removing the packages via the NuGet manager.

Note that OpenEXR is subject to its own [license](https://github.com/openexr/openexr/blob/develop/OpenEXR/LICENSE) as is [zlib](http://zlib.net/zlib_license.html).

See [Adding OpenEXR](https://github.com/Microsoft/DirectXTex/wiki/Adding-OpenEXR) for additional details.

# Update history

|  |  |
| --- | --- |
| Release | Notes |
| August 2013 | Initial release |
| October 2016 | Updated for HDR support (RGBE .hdr and optionally OpenEXR) |
| February 2017 | Updated for Direct3D 12 API support.  Breaking changes to Direct3D 11.X CreateTexture and CreateShaderResourceView (now takes ID311DeviceX instead of the deprecated ID3DXboxPerformanceDevice) |
| June 2017 | Added -xgmode switch and use of NuGet packages for OpenEXR support. |