

extended header contains the format to use in **dxgiFormat**.

```
if ( ( pHeader->ddspf.dwFlags & DDS_FOURCC )
    && (MAKEFOURCC( 'D', 'X', '1', '0' ) == pHeader->ddspf.dwFourCC ) )
{
    DDS_HEADER_DXT10* d3d10ext = (DDS_HEADER_DXT10*)( (char*)pHeader + sizeof(DDS_HEADER) );
    desc.ArraySize = d3d10ext->arraySize;
    desc.Format = d3d10ext->dxgiFormat;
}
```

Some Direct3D 9 formats may not be compatible with Direct3D 11. Formats such as D3DFMT_X8R8G8B8 and D3DFMT_A8R8G8B8 will have to be converted to DXGI_FORMAT_R8G8B8A8_UNORM, which requires that some of the components be rearranged due to the more restricted color channel ordering for DXGI.

```
// swizzle if it's a format that may not be completely compatible with D3D11
if( D3DFMT_X8R8G8B8 == fmt ||
    D3DFMT_A8R8G8B8 == fmt )
{
    for( UINT i=0; i<BitSize; i+=4 )
    {
        BYTE a = pBitData[i];
        pBitData[i] = pBitData[i+2];
        pBitData[i+2] = a;
    }
}
```

When creating a Direct3D 11 texture, the sample does not need to create a staging texture. Instead a **D3D11_SUBRESOURCE_DATA** structure is created for each mipmap in the texture. The **pSysMem** member of the structure is set to the start of the data for that mipmap, and the **SysMemPitch** member is set to the stride in bytes of that mipmap's data. When creating the texture, the array of **D3D11_SUBRESOURCE_DATA** structures is passed in as the second parameter, and the texture is automatically initialized with this data. The function then creates the resource view necessary to use the texture in a shader.

See Also

[DDSWWithoutD3DX Sample](#)

© 2010 Microsoft Corporation. All rights reserved.
Send feedback to DxSdkDoc@microsoft.com.
Version: 1962.00