# Metal 3.0 Feature Set Tables

Developer

**Feature Availability**
This table lists the availability of major Metal features.

| GPU | Common | Common | Common | Apple | Apple | Apple | Apple | Apple | Mac | Mac |
|---|---|---|---|---|---|---|---|---|---|---|
| **Family** | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 5 | 1 | 2 |
| **GPU Family** | MTLGPUFamilyCommon1 | MTLGPUFamilyCommon2 | MTLGPUFamilyCommon3 | MTLGPUFamilyApple1 | MTLGPUFamilyApple2 | MTLGPUFamilyApple3 | MTLGPUFamilyApple4 | MTLGPUFamilyApple5 | MTLGPUFamilyMac1 | MTLGPUFamilyMac2 |
| **Features** | | | | | | | | | | |
| MetalKit | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Metal Performance Shaders | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Programmable blending | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| PVRTC pixel formats | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| EAC/ETC pixel formats | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| ASTC pixel formats | | | | | ✓ | ✓ | ✓ | ✓ | | |
| BC pixel formats | | | | | | | | | ✓ | ✓ |
| Compressed Volume Texture Formats | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Extended range pixel formats | | | | | | ✓ | ✓ | ✓ | | |
| Wide color pixel format | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Depth-16 pixel format | | | | | | | | | ✓ | ✓ |
| Linear textures | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| MSAA depth resolve | | | ✓ | | | ✓ | ✓ | ✓ | | ✓ |
| Array of textures | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cube map texture arrays | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Stencil texture views | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| sRGB writes | | | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Array of samplers | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Sampler max anisotropy | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Sampler LOD clamp | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| MTLSamplerState support for comparison functions | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| 16-bit unsigned integer coordinates | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Border color | | | | | | | | | ✓ | ✓ |
| Counting occlusion query | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Base vertex/instance drawing | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Layered rendering | | | ✓ | | | | | ✓ | ✓ | ✓ |
| Memoryless render targets | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Dual-source blending | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Combined MSAA store and resolve action | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| MSAA blits | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Programmable sample positions | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| Feature | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Deferred store action | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Texture barriers | | | | | | | | | ✓ | ✓ |
| Memory barriers | | | | | | | | | ✓ | ✓ |
| Tessellation | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tessellation in Indirect Command Buffers | | | | | | | | | ✓ | ✓ |
| Resource heaps | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Function specialization | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Function buffer read-writes | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Function texture read-writes | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Extract, insert, and reverse bits | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SIMD barrier | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Indirect draw & dispatch arguments | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Argument buffers[1] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Indirect command buffers | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ |
| Uniform type | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Imageblocks | | | | | | | ✓ | ✓ | | |
| Tile shaders | | | | | | | ✓ | ✓ | | |
| Imageblock sample coverage control | | | | | | | ✓ | ✓ | | |
| Post-depth coverage | | | | | | | ✓ | ✓ | | |
| Quad-scoped permute operations | | | ✓ | | | | ✓ | ✓ | | ✓ |
| SIMD-scoped permute operations | | | | | | | | | ✓ | ✓ |
| SIMD-scoped reduction operations | | | | | | | | | | ✓ |
| Raster order groups | | | ✓ | | | | ✓ | ✓ | ✓ [2] | ✓ |
| Non-uniform threadgroup size | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ |
| Multiple viewports | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ |
| Device notifications | | | | | | | | | ✓ | ✓ |
| Stencil feedback | | | ✓ | | | | | ✓ | | ✓ |
| Stencil resolve | | | ✓ | | | | | ✓ | | ✓ |
| Non-Square Tile Dispatch | | | | | | | | ✓ | | |
| Texture Swizzle | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Placement Heap | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Pre-Clipped Primitive ID and Barycentric Coordinates | | | | | | | | | | ✓ |
| Line Width | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

[1] See About Argument Buffers for more information about argument buffer tiers, limits, and capabilities.
[2] Not all macOS Family 1 devices support raster order groups. You query MTLDevice rasterOrderGroupsSupported at runtime to check.

**Implementation Limits**
This table lists the implementation limits in Metal.

| GPU | Apple | Apple | Apple | Apple | Apple | Mac | Mac |
|---|---|---|---|---|---|---|---|
| **Family** | 1 | 2 | 3 | 4 | 5 | 1 | 2 |
| **Feature Set** | MTLGPUFamilyApple1 | MTLGPUFamilyApple2 | MTLGPUFamilyApple3 | MTLGPUFamilyApple4 | MTLGPUFamilyApple5 | MTLGPUFamilyMac1 | MTLGPUFamilyMac2 |
| **Function arguments** | | | | | | | |
| Maximum number of vertex attributes, per vertex descriptor | 31 | 31 | 31 | 31 | 31 | 31 | 31 |
| Maximum number of entries in the buffer argument table, per graphics or compute function | 31 | 31 | 31 | 31 | 31 | 31 | 31 |
| Maximum number of entries in the texture argument table, per graphics or compute function | 31 | 31 | 31 | 31 | 31 | 128 | 128 |
| Maximum number of entries in the sampler state argument table, per graphics or compute function[3] | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| Maximum number of entries in the threadgroup memory argument table, per compute function | 31 | 31 | 31 | 31 | 31 | 31 | 31 |
| Maximum number of inlined constant data buffers, per graphics or compute function | 31 | 31 | 31 | 31 | 31 | 14 | 14 |
| Maximum length of an inlined constant data buffer, per graphics or compute function | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB | 4 KB |
| Maximum threads per threadgroup | 512 | 512 | 512 | 1024 | 1024 | 1024 | 1024 |
| Maximum total threadgroup memory allocation | 16352 B | 16352 B | 16 KB | 32 KB | 32 KB | 32 KB | 32 KB |
| Maximum total tile memory allocation[4] | Not accessible | Not accessible | Not accessible | 32 KB | 32 KB | Not accessible | Not accessible |
| Threadgroup memory length alignment | 16 B | 16 B | 16 B | 16 B | 16 B | 16 B | 16 B |
| Maximum function memory allocation for a buffer in the constant address space | No limit | No limit | No limit | No limit | No limit | 64 KB | 64 KB |
| Maximum number of inputs (scalars or vectors) to a fragment function, declared with the stage_in qualifier[5] | 60 | 60 | 60 | 124 | 124 | 32 | 32 |
| Maximum number of input components to a fragment function, declared with the stage_in qualifier[5] | 60 | 60 | 60 | 124 | 124 | 124 | 124 |
| Maximum number of function constants | 65536 | 65536 | 65536 | 65536 | 65536 | 65536 | 65536 |
| Maximum tessellation factor | Not available | Not available | 16 | 16 | 64 | 64 | 64 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Maximum number of viewports and scissor rectangles, per vertex function | 1 | 1 | 1 | 1 | 16 | 16 | 16 |
| Maximum number of raster order groups, per fragment function | Not available | Not available | Not available | 8 | 8 | 8 | 8 |
| **Resources** | | | | | | | |
| Minimum buffer offset alignment | 4 B | 4 B | 4 B | 4 B | 4 B | 256 B | 256 B |
| Maximum 1D texture width | 8192 px | 8192 px | 16384 px | 16384 px | 16384 px | 16384 px | 16384 px |
| Maximum 2D texture width and height | 8192 px | 8192 px | 16384 px | 16384 px | 16384 px | 16384 px | 16384 px |
| Maximum cube map texture width and height | 8192 px | 8192 px | 16384 px | 16384 px | 16384 px | 16384 px | 16384 px |
| Maximum 3D texture width, height, and depth | 2048 px | 2048 px | 2048 px | 2048 px | 2048 px | 2048 px | 2048 px |
| Maximum number of layers per 1D texture array, 2D texture array, or 3D texture | 2048 | 2048 | 2048 | 2048 | 2048 | 2048 | 2048 |
| Buffer alignment for copying an existing texture to a buffer | 64 B | 64 B | 16 B | 16 B | 16 B | 256 B | 256 B |
| **Render Targets** | | | | | | | |
| Maximum number of color render targets per render pass descriptor | 4 | 8 | 8 | 8 | 8 | 8 | 8 |
| Maximum size of a point primitive | 511 | 511 | 511 | 511 | 511 | 511 | 511 |
| Maximum total render target size, per pixel, when using multiple color render targets | 128 bits | 256 bits | 256 bits | 512 bits | 512 bits | No limit | No limit |
| Maximum visibility query offset | 65528 B | 65528 B | 65528 B | 65528 B | 65528 B | 65528 B | 65528 B |
| **Synchronization** | | | | | | | |
| Maximum number of fences | 32768 | 32768 | 32768 | 32768 | 32768 | 32768 | 32768 |

[3] Inline constant samplers, declared in Metal shading language code, also count against this limit. For example, if the feature set limit is 16, you can have 12 API samplers and 4 language samplers (16 total) but you cannot have 12 API samplers and 6 language samplers (18 total).

[4] Tile memory can be allocated between imageblocks and threadgroup memory, but the sum of these allocations cannot exceed the maximum total tile memory limit. Some feature sets cannot access tile memory directly, but they can access threadgroup memory.

[5] A vector counts as n scalars, where n is the number of components in the vector. In iOS and tvOS feature sets, you can only reach the maximum number of inputs if you do not exceed the maximum number of input components. For example, you can have 60 float inputs (60 input components) but you cannot have 60 float4 inputs (240 input components).

**Pixel Format Capabilities**

This table lists the capabilities of all Metal pixel formats. These capabilities determine the operations that can be performed on a texture that uses a given pixel format. All graphics and compute functions can read or sample from any texture, regardless of its pixel format. Additional capabilities are defined as follows:

- Filter—the texture can be filtered during sampling.
- Write—the texture can be written to by a function.[6]
- Color—the texture can be used as a color render target.
- Blend—the texture can be blended.
- MSAA—the texture can be used as the destination for multisample antialias (MSAA) data.
- Resolve—the texture can be used as the destination for resolved MSAA data.
- All—the texture has all the previously-listed capabilities.

| GPU | Apple | Apple | Apple | Apple | Apple | Mac | Mac |
|---|---|---|---|---|---|---|---|
| **Family** | 1 | 2 | 3 | 4 | 5 | 1 | 2 |
| **Feature Set** | MTLGPUFamilyApple1 | MTLGPUFamilyApple2 | MTLGPUFamilyApple3 | MTLGPUFamilyApple4 | MTLGPUFamilyApple5 | MTLGPUFamilyMac1 | MTLGPUFamilyMac2 |
| **Ordinary 8-bit pixel formats** | | | | | | | |
| A8Unorm | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| R8Unorm[8] | All | All | All | All | All | All | All |
| R8Unorm_sRGB | Filter Color MSAA Resolve Blend | All | All | All | All | Not available | Not available |
| R8Snorm | Filter Write Color MSAA Blend | All | All | All | All | All | All |
| R8Uint[8] R8Sint | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA |
| **Ordinary 16-bit pixel formats** | | | | | | | |
| R16Unorm R16Snorm | Filter Write Color MSAA Blend | Filter Write Color MSAA Blend | Filter Write Color MSAA Blend | Filter Write Color MSAA Blend | Filter Write Color MSAA Blend | All | All |

| Format | | | | | | | |
|---|---|---|---|---|---|---|---|
| R16Uint[8]<br>R16Sint[8] | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| R16Float[8] | All | All | All | All | All | All | All |
| RG8Unorm | All | All | All | All | All | All | All |
| RG8Unorm_sRGB | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | All | All | All | All | Not available | Not available |
| RG8Snorm | Filter<br>Write<br>Color<br>MSAA<br>Blend | All | All | All | All | All | All |
| RG8Uint<br>RG8Sint | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| **Packed 16-bit pixel formats** | | | | | | | |
| B5G6R5Unorm<br>A1BGR5Unorm<br>ABGR4Unorm<br>BGR5A1Unorm | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | Filter<br>Color<br>MSAA<br>Resolve<br>Blend | Not available | Not available |
| **Ordinary 32-bit pixel formats** | | | | | | | |
| R32Uint[8]<br>R32Sint[8] | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| R32Float[8] | Write<br>Color<br>MSAA<br>Blend | Write<br>Color<br>MSAA<br>Blend | Write<br>Color<br>MSAA<br>Blend | Write<br>Color<br>MSAA<br>Blend | Write<br>Color<br>MSAA<br>Blend | All | All |
| RG16Unorm<br>RG16Snorm | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | All | All |
| RG16Uint<br>RG16Sint | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| RG16Float | All | All | All | All | All | All | All |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RGBA8Unorm | All | All | All | All | All | All | All |
| RGBA8Unorm_sRGB | Filter Color MSAA Resolve Blend | All | All | All | All | Filter Color MSAA Resolve Blend | Filter Color MSAA Resolve Blend |
| RGBA8Snorm | Filter Write Color MSAA Blend | All | All | All | All | All | All |
| RGBA8Uint[8] RGBA8Sint[8] | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA |
| BGRA8Unorm[8] | All | All | All | All | All | All | All |
| BGRA8Unorm_sRGB | Filter Color MSAA Resolve Blend | All | All | All | All | Filter Color MSAA Resolve Blend | Filter Color MSAA Resolve Blend |
| **Packed 32-bit pixel formats** | | | | | | | |
| RGB10A2Unorm | Filter Color MSAA Resolve Blend | Filter Color MSAA Resolve Blend | All | All | All | All | All |
| RGB10A2Uint | Color MSAA | Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA | Write Color MSAA |
| RG11B10Float | Filter Color MSAA Resolve Blend | Filter Color MSAA Resolve Blend | All | All | All | All | All |
| RGB9E5Float | Filter Color MSAA Resolve Blend | Filter Color MSAA Resolve Blend | All | All | All | Filter | Filter |
| **Ordinary 64-bit pixel formats** | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RG32Uint<br>RG32Sint | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| RG32Float | Write<br>Color<br>Blend | Write<br>Color<br>Blend | Write<br>Color<br>Blend | Write<br>Color<br>Blend | Write<br>Color<br>Blend | All | All |
| RGBA16Unorm<br>RGBA16Snorm | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | Filter<br>Write<br>Color<br>MSAA<br>Blend | All | All |
| RGBA16Uint[8]<br>RGBA16Sint[8] | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| RGBA16Float[8] | All | All | All | All | All | All | All |
| **Ordinary 128-bit pixel formats** | | | | | | | |
| RGBA32Uint[8]<br>RGBA32Sint[8] | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color<br>MSAA | Write<br>Color<br>MSAA |
| RGBA32Float[8] | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | Write<br>Color | All | All |
| **Compressed pixel formats** | | | | | | | |
| PVRTC pixel formats[7] | Filter | Filter | Filter | Filter | Filter | Not available | Not available |
| EAC/ETC pixel formats | Filter | Filter | Filter | Filter | Filter | Not available | Not available |
| ASTC pixel formats | Not available | Filter | Filter | Filter | Filter | Not available | Not available |
| BC pixel formats | Not available | Not available | Not available | Not available | Not available | Filter | Filter |
| **YUV pixel formats** | | | | | | | |
| GBGR422<br>BGRG422 | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| **Depth and stencil pixel formats** | | | | | | | |
| Depth16Unorm | Not available | Not available | Not available | Not available | Not available | Filter<br>MSAA<br>Resolve | Filter<br>MSAA<br>Resolve |
| Depth32Float | MSAA | MSAA | MSAA<br>Resolve | MSAA<br>Resolve | MSAA<br>Resolve | Filter<br>MSAA<br>Resolve | Filter<br>MSAA<br>Resolve |
| Stencil8 | MSAA | MSAA | MSAA | MSAA | MSAA | MSAA | MSAA |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Depth24Unorm_Stencil8[8] | Not available | Not available | Not available | Not available | Not available | Filter MSAA Resolve | Filter MSAA Resolve |
| Depth32Float_Stencil8 | MSAA | MSAA | MSAA Resolve | MSAA Resolve | MSAA Resolve | Filter MSAA Resolve | Filter MSAA Resolve |
| X24_Stencil8 | Not available | Not available | Not available | Not available | Not available | MSAA | MSAA |
| X32_Stencil8 | MSAA | MSAA | MSAA | MSAA | MSAA | MSAA | MSAA |
| **Extended range and wide color pixel formats** | | | | | | | |
| BGRA10_XR BGRA10_XR_sRGB BGR10_XR BGR10_XR_sRGB | Not available | Not available | All | All | All | Not available | Not available |
| BGR10A2Unorm | All | All | All | All | All | Filter Color MSAA Resolve Blend | Filter Write Color MSAA Resolve Blend |

[6] Read-write textures are available in some feature sets, where the texture can be both read from and written to by the same function. Query the readWriteTextureSupport property of MTLDevice to determine support.

[7] For PVRTC pixel formats, the clamp_to_zero sampler state is supported only in the iOS GPU Family 3 and 4 feature sets.

[8] Support for this pixel format varies by device. Query the depth24Stencil8PixelFormatSupported property of MTLDevice to determine support.