

# Metal Feature Set Tables

## Apple GPUs

Apple GPU Family	GPUs in Family	Corresponding Feature Sets
<b>MTLGPUFamilyApple1</b>	A7	iOS GPU family 1
<b>MTLGPUFamilyApple2</b>	A8	iOS GPU family 2, tvOS GPU family 1
<b>MTLGPUFamilyApple3</b>	A9, A10	iOS GPU family 3, tvOS GPU family 2
<b>MTLGPUFamilyApple4</b>	A11	iOS GPU family 4
<b>MTLGPUFamilyApple5</b>	A12	iOS GPU family 5
<b>MTLGPUFamilyApple6</b>	A13	—
This table lists current Apple GPU families, which processors are in each family, and how they relate to older feature sets.		

## Feature Availability

This table lists the availability of major Metal features.

GPU Family	Common 1	Common 2	Common 3	Apple 1	Apple 2	Apple 3	Apple 4	Apple 5	Apple 6	Mac 1	Mac 2
GPU Family	MTLGPUPFamilyCommon1	MTLGPUPFamilyCommon2	MTLGPUPFamilyCommon3	MTLGPUPFamilyApple1	MTLGPUPFamilyApple2	MTLGPUPFamilyApple3	MTLGPUPFamilyApple4	MTLGPUPFamilyApple5	MTLGPUPFamilyApple6	MTLGPUPFamilyMac1	MTLGPUPFamilyMac2
<b>Features</b>											
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			✓					✓	✓	✓	✓
Layered rendering to multisample textures <sup>1</sup>											✓
Memoryless render targets				✓	✓	✓	✓	✓	✓		
Dual-source blending	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Combined MSAA store and resolve action		✓	✓			✓	✓	✓	✓	✓	✓
MSAA blits	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Programmable sample positions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Deferred store action	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Texture barriers										✓	✓

Memory barriers										✓	✓
Tessellation		✓	✓			✓	✓	✓	✓	✓	✓
Indirect tessellation arguments								✓	✓	✓	✓
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 <sup>2</sup>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
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			✓				✓	✓	✓	✓ <sup>3</sup>	✓
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											✓
Read or Write Cube Map Textures							✓	✓	✓	✓	✓
Sparse Textures									✓		
Variable Rasterization Rate									✓		
Vertex Amplification									✓		

<sup>1</sup> Available on MacOS 10.14+.

<sup>2</sup> See [About Argument Buffers](#) for more information about argument buffer tiers, limits, and capabilities.

<sup>3</sup> 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	Apple	Mac	Mac
Family	1	2	3	4	5	6	1	2
Feature Set	MTLGPUFamil yApple1	MTLGPUFamil yApple2	MTLGPUFamil yApple3	MTLGPUFamil yApple4	MTLGPUFamil yApple5	MTLGPUFamil yApple6	MTLGPUFamil yMac1	MTLGPUFamil yMac2
<b>Function arguments</b>								
Maximum number of vertex attributes, per vertex descriptor	31	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	31
Maximum number of entries in the texture argument table, per graphics or compute function	31	31	31	96	96	128	128	128
Maximum number of entries in the sampler state argument table, per graphics or compute function <sup>3</sup>	16	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	31
Maximum number of inlined constant data buffers, per graphics or compute function	31	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	4 KB
Maximum threads per threadgroup	512	512	512	1024	1024	1024	1024	1024
Maximum total threadgroup memory allocation	16352 B	16352 B	16 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Maximum total tile memory allocation <sup>4</sup>	Not accessible	Not accessible	Not accessible	32 KB	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	16 B
Maximum function memory allocation for a buffer in the constant address space	No limit	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 <sup>5</sup>	60	60	60	124	124	124	32	32
Maximum number of input components to a fragment function, declared with the stage_in qualifier <sup>5</sup>	60	60	60	124	124	124	124	124
Maximum number of function constants	65536	65536	65536	65536	65536	65536	65536	65536
Maximum tessellation factor	Not available	Not available	16	16	64	64	64	64
Maximum number of viewports and scissor rectangles, per vertex function	1	1	1	1	16	16	16	16
Maximum number of raster order groups, per fragment function	Not available	Not available	Not available	8	8	8	8	8
<b>Resources</b>								

Minimum buffer offset alignment	4 B	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	16384 px
Maximum 2D texture width and height	8192 px	8192 px	16384 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	16384 px
Maximum 3D texture width, height, and depth	2048 px	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	2048
Buffer alignment for copying an existing texture to a buffer	64 B	64 B	16 B	16 B	16 B	16 B	256 B	256 B
<b>Render Targets</b>								
Maximum number of color render targets per render pass descriptor	4	8	8	8	8	8	8	8
Maximum size of a point primitive	511	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	512 bits	No limit	No limit
Maximum visibility query offset	65528 B	65528 B	65528 B	65528 B	65528 B	65528 B	256 KB	256 KB
<b>Synchronization</b>								
Maximum number of fences	32768	32768	32768	32768	32768	32768	32768	32768
Maximum vertex count for vertex amplification	Not available	Not available	Not available	Not available	Not available	2	Not available	Not available

<sup>3</sup> 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).

<sup>4</sup> 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.

<sup>5</sup> 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.<sup>6</sup>
- 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	Apple	Mac	Mac
Family	1	2	3	4	5	6	1	2
Feature Set	MTLGPUFamil yApple1	MTLGPUFamil yApple2	MTLGPUFamil yApple3	MTLGPUFamil yApple4	MTLGPUFamil yApple5	MTLGPUFamil yApple6	MTLGPUFamil yMac1	MTLGPUFamil yMac2
<b>Ordinary 8-bit pixel formats</b>								
A8Unorm	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
R8Unorm <sup>8</sup>	All	All	All	All	All	All	All	All
R8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	Not available	Not available
R8Snorm	Filter Write Color MSAA Blend	All	All	All	All	All	All	All
R8Uint <sup>8</sup> R8Sint	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
<b>Ordinary 16-bit pixel formats</b>								
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	Filter Write Color MSAA Blend	All	All
R16Uint <sup>8</sup> R16Sint <sup>8</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
R16Float <sup>8</sup>	All	All	All	All	All	All	All	All

RG8Unorm	All	All	All	All	All	All	All	All
RG8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	Not available Not available
RG8Snorm	Filter Write Color MSAA Blend	All	All	All	All	All	All	All
RG8Uint RG8Sint	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
<b>Packed 16-bit pixel formats</b>								
B5G6R5Unorm A1BGR5Unorm ABGR4Unorm BGR5A1Unorm	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Not available Not available
<b>Ordinary 32-bit pixel formats</b>								
R32Uint <sup>8</sup> R32Sint <sup>8</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color MSAA	Write Color MSAA
R32Float <sup>8</sup>	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	All	All
RG16Unorm RG16Snorm	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	Filter Write Color MSAA Blend	All	All
RG16Uint RG16Sint	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RG16Float	All	All	All	All	All	All	All	All
RGBA8Unorm	All	All	All	All	All	All	All	All
RGBA8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	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	All
RGBA8Uint <sup>8</sup> RGBA8Sint <sup>8</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
BGRA8Unorm <sup>8</sup>	All	All	All	All	All	All	All	All
BGRA8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend
<b>Packed 32-bit pixel formats</b>								
RGB10A2Unorm	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	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	Write Color MSAA
RG11B10Float	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	All	All	All	All	All
RGB9E5Float	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	All	All	All	Filter	Filter
<b>Ordinary 64-bit pixel formats</b>								
RG32Uint RG32Sint	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color MSAA	Write Color MSAA
RG32Float	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend	All	All
RGBA16Unorm RGBA16Snorm	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	Filter Write Color MSAA Blend	All	All

RGBA16UInt <sup>8</sup> RGBA16Sint <sup>8</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RGBA16Float <sup>8</sup>	All	All	All	All	All	All	All	All
<b>Ordinary 128-bit pixel formats</b>								
RGBA32UInt <sup>8</sup> RGBA32Sint <sup>8</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color MSAA	Write Color MSAA
RGBA32Float <sup>8</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color	All	All
<b>Compressed pixel formats</b>								
PVRTC pixel formats <sup>7</sup>	Filter	Filter	Filter	Filter	Filter	Filter	Not available	Not available
EAC/ETC pixel formats	Filter	Filter	Filter	Filter	Filter	Filter	Not available	Not available
ASTC pixel formats	Not available	Filter	Filter	Filter	Filter	Filter	Not available	Not available
HDR ASTC pixel formats	Not available	Not available	Not available	Not available	Not available	Filter	Not available	Not available
BC pixel formats	Not available	Not available	Not available	Not available	Not available	Not available	Filter	Filter
<b>YUV pixel formats</b>								
GBGR422 BGRG422	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
<b>Depth and stencil pixel formats</b>								
Depth16Unorm	Filter MSAA	Filter MSAA	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
Depth32Float	MSAA	MSAA	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
Stencil8	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA
Depth24Unorm_Stencil8 <sup>8</sup>	Not available	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	MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
X24_Stencil8	Not available	Not available	Not available	Not available	Not available	Not available	MSAA	MSAA
X32_Stencil8	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA
<b>Extended range and wide color pixel formats</b>								
BGRA10_XR BGRA10_XR_sRGB BGR10_XR BGR10_XR_sRGB	Not available	Not available	All	All	All	All	Not available	Not available

BGR10A2Unorm	All	All	All	All	All	All	Filter Color MSAA Resolve Blend	Filter Write Color MSAA Resolve Blend
--------------	-----	-----	-----	-----	-----	-----	---	--

<sup>6</sup> 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.

<sup>7</sup> For PVRTC pixel formats, the `clamp_to_zero` sampler state is supported only in the iOS GPU Family 3 and 4 feature sets.

<sup>8</sup> Support for this pixel format varies by device. Query the `depth24Stencil8PixelFormatSupported` property of `MTLDevice` to determine support.



Apple Inc.  
Copyright © 2019 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer or device for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-branded products.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple is a trademark of Apple Inc., registered in the U.S. and other countries.

**APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.**