



# RenderScript Index

## Constants

<a href="#">M_1_PI</a>	1 / pi, as a 32 bit float
<a href="#">M_2_PI</a>	2 / pi, as a 32 bit float
<a href="#">M_2_SQRTPI</a>	2 / sqrt(pi), as a 32 bit float
<a href="#">M_E</a>	e, as a 32 bit float
<a href="#">M_LN10</a>	log_e(10), as a 32 bit float
<a href="#">M_LN2</a>	log_e(2), as a 32 bit float
<a href="#">M_LOG10E</a>	log_10(e), as a 32 bit float
<a href="#">M_LOG2E</a>	log_2(e), as a 32 bit float
<a href="#">M_PI</a>	pi, as a 32 bit float
<a href="#">M_PI_2</a>	pi / 2, as a 32 bit float
<a href="#">M_PI_4</a>	pi / 4, as a 32 bit float
<a href="#">M_SQRT1_2</a>	1 / sqrt(2), as a 32 bit float
<a href="#">M_SQRT2</a>	sqrt(2), as a 32 bit float

## Types

<a href="#">char2</a>	Two 8 bit signed integers
<a href="#">char3</a>	Three 8 bit signed integers
<a href="#">char4</a>	Four 8 bit signed integers
<a href="#">double2</a>	Two 64 bit floats
<a href="#">double3</a>	Three 64 bit floats
<a href="#">double4</a>	Four 64 bit floats
<a href="#">float2</a>	Two 32 bit floats
<a href="#">float3</a>	Three 32 bit floats
<a href="#">float4</a>	Four 32 bit floats
<a href="#">half</a>	16 bit floating point value
<a href="#">half2</a>	Two 16 bit floats
<a href="#">half3</a>	Three 16 bit floats
<a href="#">half4</a>	Four 16 bit floats
<a href="#">int16_t</a>	16 bit signed integer
<a href="#">int2</a>	Two 32 bit signed integers
<a href="#">int3</a>	Three 32 bit signed integers
<a href="#">int32_t</a>	32 bit signed integer

int4	Four 32 bit signed integers
int64_t	64 bit signed integer
int8_t	8 bit signed integer
long2	Two 64 bit signed integers
long3	Three 64 bit signed integers
long4	Four 64 bit signed integers
rs_allocation	Handle to an allocation
rs_allocation_cubemap_face	Enum for selecting cube map faces
rs_allocation_usage_type	Bitfield to specify how an allocation is used
rs_data_kind	Element data kind
rs_data_type	Element basic data type
rs_element	Handle to an element
rs_for_each_strategy_t	Suggested cell processing order
rs_kernel	Handle to a kernel function
rs_kernel_context	Handle to a kernel invocation context
rs_matrix2x2	2x2 matrix of 32 bit floats
rs_matrix3x3	3x3 matrix of 32 bit floats
rs_matrix4x4	4x4 matrix of 32 bit floats
rs_quaternion	Quaternion
rs_sampler	Handle to a Sampler
rs_sampler_value	Sampler wrap T value
rs_script	Handle to a Script
rs_script_call_t	Cell iteration information
rs_time_t	Seconds since January 1, 1970
rs_tm	Date and time structure
rs_type	Handle to a Type
rs_yuv_format	YUV format
short2	Two 16 bit signed integers
short3	Three 16 bit signed integers
short4	Four 16 bit signed integers
size_t	Unsigned size type
ssize_t	Signed size type
uchar	8 bit unsigned integer
uchar2	Two 8 bit unsigned integers
uchar3	Three 8 bit unsigned integers
uchar4	Four 8 bit unsigned integers
uint	32 bit unsigned integer
uint16_t	16 bit unsigned integer
uint2	Two 32 bit unsigned integers

uint3	Three 32 bit unsigned integers
uint32_t	32 bit unsigned integer
uint4	Four 32 bit unsigned integers
uint64_t	64 bit unsigned integer
uint8_t	8 bit unsigned integer
ulong	64 bit unsigned integer
ulong2	Two 64 bit unsigned integers
ulong3	Three 64 bit unsigned integers
ulong4	Four 64 bit unsigned integers
ushort	16 bit unsigned integer
ushort2	Two 16 bit unsigned integers
ushort3	Three 16 bit unsigned integers
ushort4	Four 16 bit unsigned integers

## Functions

abs	Absolute value of an integer
acos	Inverse cosine
acosh	Inverse hyperbolic cosine
acospi	Inverse cosine divided by pi
asin	Inverse sine
asinh	Inverse hyperbolic sine
asinpi	Inverse sine divided by pi
atan	Inverse tangent
atan2	Inverse tangent of a ratio
atan2pi	Inverse tangent of a ratio, divided by pi
atanh	Inverse hyperbolic tangent
atanpi	Inverse tangent divided by pi
cbrt	Cube root
ceil	Smallest integer not less than a value
clamp	Restrain a value to a range
clz	Number of leading 0 bits
convert	Convert numerical vectors
copysign	Copies the sign of a number to another
cos	Cosine
cosh	Hypebolic cosine
cospi	Cosine of a number multiplied by pi
cross	Cross product of two vectors
degrees	Converts radians into degrees
distance	Distance between two points

dot	Dot product of two vectors
erf	Mathematical error function
erfc	Mathematical complementary error function
exp	e raised to a number
exp10	10 raised to a number
exp2	2 raised to a number
expm1	e raised to a number minus one
fabs	Absolute value of a float
fast_distance	Approximate distance between two points
fast_length	Approximate length of a vector
fast_normalize	Approximate normalized vector
fdim	Positive difference between two values
floor	Smallest integer not greater than a value
fma	Multiply and add
fmax	Maximum of two floats
fmin	Minimum of two floats
fmod	Modulo
fract	Positive fractional part
frexp	Binary mantissa and exponent
half_recip	Reciprocal computed to 16 bit precision
half_rsqr	Reciprocal of a square root computed to 16 bit precision
half_sqrt	Square root computed to 16 bit precision
hypot	Hypotenuse
ilogb	Base two exponent
ldexp	Creates a floating point from mantissa and exponent
length	Length of a vector
lgamma	Natural logarithm of the gamma function
log	Natural logarithm
log10	Base 10 logarithm
log1p	Natural logarithm of a value plus 1
log2	Base 2 logarithm
logb	Base two exponent
mad	Multiply and add
max	Maximum
min	Minimum
mix	Mixes two values
modf	Integral and fractional components
nan	Not a Number
nan_half	Not a Number
native_acos	Approximate inverse cosine

native_acosh	Approximate inverse hyperbolic cosine
native_acospi	Approximate inverse cosine divided by pi
native_asin	Approximate inverse sine
native_asinh	Approximate inverse hyperbolic sine
native_asinpi	Approximate inverse sine divided by pi
native_atan	Approximate inverse tangent
native_atan2	Approximate inverse tangent of a ratio
native_atan2pi	Approximate inverse tangent of a ratio, divided by pi
native_atanh	Approximate inverse hyperbolic tangent
native_atanpi	Approximate inverse tangent divided by pi
native_cbrt	Approximate cube root
native_cos	Approximate cosine
native_cosh	Approximate hypebolic cosine
native_cospi	Approximate cosine of a number multiplied by pi
native_distance	Approximate distance between two points
native_divide	Approximate division
native_exp	Approximate e raised to a number
native_exp10	Approximate 10 raised to a number
native_exp2	Approximate 2 raised to a number
native_expm1	Approximate e raised to a number minus one
native_hypot	Approximate hypotenuse
native_length	Approximate length of a vector
native_log	Approximate natural logarithm
native_log10	Approximate base 10 logarithm
native_log1p	Approximate natural logarithm of a value plus 1
native_log2	Approximate base 2 logarithm
native_normalize	Approximately normalize a vector
native_powr	Approximate positive base raised to an exponent
native_recip	Approximate reciprocal
native_rootn	Approximate nth root
native_rsqrt	Approximate reciprocal of a square root
native_sin	Approximate sine
native_sincos	Approximate sine and cosine
native_sinh	Approximate hyperbolic sine
native_sinpi	Approximate sine of a number multiplied by pi
native_sqrt	Approximate square root
native_tan	Approximate tangent
native_tanh	Approximate hyperbolic tangent
native_tanpi	Approximate tangent of a number multiplied by pi

<a href="#">nextafter</a>	Next floating point number
<a href="#">normalize</a>	Normalize a vector
<a href="#">pow</a>	Base raised to an exponent
<a href="#">pown</a>	Base raised to an integer exponent
<a href="#">powr</a>	Positive base raised to an exponent
<a href="#">radians</a>	Converts degrees into radians
<a href="#">remainder</a>	Remainder of a division
<a href="#">remquo</a>	Remainder and quotient of a division
<a href="#">rint</a>	Round to even
<a href="#">rootn</a>	Nth root
<a href="#">round</a>	Round away from zero
<a href="#">rsAllocationCopy1DRange</a>	Copy consecutive cells between allocations
<a href="#">rsAllocationCopy2DRange</a>	Copy a rectangular region of cells between allocations
<a href="#">rsAllocationGetDimFaces</a>	Presence of more than one face
<a href="#">rsAllocationGetDimLOD</a>	Presence of levels of detail
<a href="#">rsAllocationGetDimX</a>	Size of the X dimension
<a href="#">rsAllocationGetDimY</a>	Size of the Y dimension
<a href="#">rsAllocationGetDimZ</a>	Size of the Z dimension
<a href="#">rsAllocationGetElement</a>	Get the object that describes the cell of an Allocation
<a href="#">rsAllocationIoReceive</a>	Receive new content from the queue
<a href="#">rsAllocationIoSend</a>	Send new content to the queue
<a href="#">rsAllocationVLoadX</a>	Get a vector from an allocation of scalars
<a href="#">rsAllocationVStoreX</a>	Store a vector into an allocation of scalars
<a href="#">rsAtomicAdd</a>	Thread-safe addition
<a href="#">rsAtomicAnd</a>	Thread-safe bitwise and
<a href="#">rsAtomicCas</a>	Thread-safe compare and set
<a href="#">rsAtomicDec</a>	Thread-safe decrement
<a href="#">rsAtomicInc</a>	Thread-safe increment
<a href="#">rsAtomicMax</a>	Thread-safe maximum
<a href="#">rsAtomicMin</a>	Thread-safe minimum
<a href="#">rsAtomicOr</a>	Thread-safe bitwise or
<a href="#">rsAtomicSub</a>	Thread-safe subtraction
<a href="#">rsAtomicXor</a>	Thread-safe bitwise exclusive or
<a href="#">rsClearObject</a>	Release an object
<a href="#">rsCreateAllocation</a>	Create an rs_allocation object of given Type.
<a href="#">rsCreateElement</a>	Creates an rs_element object of the specified data type
<a href="#">rsCreatePixelElement</a>	Creates an rs_element object of the specified data type and data kind
<a href="#">rsCreateType</a>	Creates an rs_type object with the specified Element and shape attributes
<a href="#">rsCreateVectorElement</a>	Creates an rs_element object of the specified data type and vector width

rsDebug	Log a message and values
rsElementGetBytesSize	Size of an Element
rsElementGetDataKind	Kind of an Element
rsElementGetDataType	Data type of an Element
rsElementGetSubElement	Sub-element of a complex Element
rsElementGetSubElementArraySize	Array size of a sub-element of a complex Element
rsElementGetSubElementCount	Number of sub-elements
rsElementGetSubElementName	Name of a sub-element
rsElementGetSubElementNameLength	Length of the name of a sub-element
rsElementGetSubElementOffsetBytes	Offset of the instantiated sub-element
rsElementGetVectorSize	Vector size of the Element
rsExtractFrustumPlanes	Compute frustum planes
rsForEach	Launches a kernel
rsForEachInternal	(Internal API) Launch a kernel in the current Script (with the slot number)
rsForEachWithOptions	Launches a kernel with options
rsGetArray0	Index in the Array0 dimension for the specified kernel context
rsGetArray1	Index in the Array1 dimension for the specified kernel context
rsGetArray2	Index in the Array2 dimension for the specified kernel context
rsGetArray3	Index in the Array3 dimension for the specified kernel context
rsGetDimArray0	Size of the Array0 dimension for the specified kernel context
rsGetDimArray1	Size of the Array1 dimension for the specified kernel context
rsGetDimArray2	Size of the Array2 dimension for the specified kernel context
rsGetDimArray3	Size of the Array3 dimension for the specified kernel context
rsGetDimHasFaces	Presence of more than one face for the specified kernel context
rsGetDimLod	Number of levels of detail for the specified kernel context
rsGetDimX	Size of the X dimension for the specified kernel context
rsGetDimY	Size of the Y dimension for the specified kernel context
rsGetDimZ	Size of the Z dimension for the specified kernel context
rsGetDt	Elapsed time since last call
rsGetElementAt	Return a cell from an allocation
rsGetElementAtYuv_uchar_U	Get the U component of an allocation of YUVs
rsGetElementAtYuv_uchar_V	Get the V component of an allocation of YUVs
rsGetElementAtYuv_uchar_Y	Get the Y component of an allocation of YUVs
rsGetFace	Coordinate of the Face for the specified kernel context
rsGetLod	Index in the Levels of Detail dimension for the specified kernel context
rsIsObject	Check for an empty handle
rsIsSphereInFrustum	Checks if a sphere is within the frustum planes
rsLocaltime	Convert to local time
rsMatrixGet	Get one element
rsMatrixInverse	Inverts a matrix in place

rsMatrixInverseTranspose	Inverts and transpose a matrix in place
rsMatrixLoad	Load or copy a matrix
rsMatrixLoadFrustum	Load a frustum projection matrix
rsMatrixLoadIdentity	Load identity matrix
rsMatrixLoadMultiply	Multiply two matrices
rsMatrixLoadOrtho	Load an orthographic projection matrix
rsMatrixLoadPerspective	Load a perspective projection matrix
rsMatrixLoadRotate	Load a rotation matrix
rsMatrixLoadScale	Load a scaling matrix
rsMatrixLoadTranslate	Load a translation matrix
rsMatrixMultiply	Multiply a matrix by a vector or another matrix
rsMatrixRotate	Apply a rotation to a transformation matrix
rsMatrixScale	Apply a scaling to a transformation matrix
rsMatrixSet	Set one element
rsMatrixTranslate	Apply a translation to a transformation matrix
rsMatrixTranspose	Transpose a matrix place
rsPackColorTo8888	Create a uchar4 RGBA from floats
rsQuaternionAdd	Add two quaternions
rsQuaternionConjugate	Conjugate a quaternion
rsQuaternionDot	Dot product of two quaternions
rsQuaternionGetMatrixUnit	Get a rotation matrix from a quaternion
rsQuaternionLoadRotate	Create a rotation quaternion
rsQuaternionLoadRotateUnit	Quaternion that represents a rotation about an arbitrary unit vector
rsQuaternionMultiply	Multiply a quaternion by a scalar or another quaternion
rsQuaternionNormalize	Normalize a quaternion
rsQuaternionSet	Create a quaternion
rsQuaternionSlerp	Spherical linear interpolation between two quaternions
rsRand	Pseudo-random number
rsSample	Sample a value from a texture allocation
rsSamplerGetAnisotropy	Anisotropy of the Sampler
rsSamplerGetMagnification	Sampler magnification value
rsSamplerGetMinification	Sampler minification value
rsSamplerGetWrapS	Sampler wrap S value
rsSamplerGetWrapT	Sampler wrap T value
rsSendToClient	Send a message to the client, non-blocking
rsSendToClientBlocking	Send a message to the client, blocking
rsSetElementAt	Set a cell of an allocation
rsTime	Seconds since January 1, 1970
rsUnpackColor8888	Create a float4 RGBA from uchar4



<a href="#">rsUptimeMillis</a> <a href="#">rsUptimeNanos</a>	System uptime in milliseconds System uptime in nanoseconds
<a href="#">rsYuvToRGBA</a>	Convert a YUV value to RGBA
<a href="#">rsqrt</a>	Reciprocal of a square root
<a href="#">sign</a>	Sign of a value
<a href="#">sin</a>	Sine
<a href="#">sincos</a>	Sine and cosine
<a href="#">sinh</a>	Hyperbolic sine
<a href="#">sinpi</a>	Sine of a number multiplied by pi
<a href="#">sqrt</a>	Square root
<a href="#">step</a>	0 if less than a value, 0 otherwise
<a href="#">tan</a>	Tangent
<a href="#">tanh</a>	Hyperbolic tangent
<a href="#">tanpi</a>	Tangent of a number multiplied by pi
<a href="#">tgamma</a>	Gamma function
<a href="#">trunc</a>	Truncates a floating point

## Deprecated Types

<a href="#">rs_blend_dst_func</a>	<b>Deprecated.</b> Blend destination function
<a href="#">rs_blend_src_func</a>	<b>Deprecated.</b> Blend source function
<a href="#">rs_cull_mode</a>	<b>Deprecated.</b> Culling mode
<a href="#">rs_depth_func</a>	<b>Deprecated.</b> Depth function
<a href="#">rs_font</a>	<b>Deprecated.</b> Handle to a Font
<a href="#">rs_mesh</a>	<b>Deprecated.</b> Handle to a Mesh
<a href="#">rs_primitive</a>	<b>Deprecated.</b> How to intepret mesh vertex data
<a href="#">rs_program_fragment</a>	<b>Deprecated.</b> Handle to a ProgramFragment
<a href="#">rs_program_raster</a>	<b>Deprecated.</b> Handle to a ProgramRaster
<a href="#">rs_program_store</a>	<b>Deprecated.</b> Handle to a ProgramStore
<a href="#">rs_program_vertex</a>	<b>Deprecated.</b> Handle to a ProgramVertex

## Deprecated Functions

<a href="#">rsClamp</a>	<b>Deprecated.</b> Restrain a value to a range
<a href="#">rsFrac</a>	<b>Deprecated.</b> Returns the fractional part of a float
<a href="#">rsGetAllocation</a>	<b>Deprecated.</b> Return the Allocation for a given pointer
<a href="#">rsgAllocationSyncAll</a>	<b>Deprecated.</b> Sync the contents of an allocation
<a href="#">rsgBindColorTarget</a>	<b>Deprecated.</b> Set the color target
<a href="#">rsgBindConstant</a>	<b>Deprecated.</b> Bind a constant allocation
<a href="#">rsgBindDepthTarget</a>	<b>Deprecated.</b> Set the depth target

<a href="#">rsgBindFont</a>	<b>Deprecated.</b> Bind a font object
<a href="#">rsgBindProgramFragment</a>	<b>Deprecated.</b> Bind a ProgramFragment
<a href="#">rsgBindProgramRaster</a>	<b>Deprecated.</b> Bind a ProgramRaster
<a href="#">rsgBindProgramStore</a>	<b>Deprecated.</b> Bind a ProgramStore
<a href="#">rsgBindProgramVertex</a>	<b>Deprecated.</b> Bind a ProgramVertex
<a href="#">rsgBindSampler</a>	<b>Deprecated.</b> Bind a sampler
<a href="#">rsgBindTexture</a>	<b>Deprecated.</b> Bind a texture allocation
<a href="#">rsgClearAllRenderTargets</a>	<b>Deprecated.</b> Clear all color and depth targets
<a href="#">rsgClearColor</a>	<b>Deprecated.</b> Clear the specified color from the surface
<a href="#">rsgClearColorTarget</a>	<b>Deprecated.</b> Clear the color target
<a href="#">rsgClearDepth</a>	<b>Deprecated.</b> Clear the depth surface
<a href="#">rsgClearDepthTarget</a>	<b>Deprecated.</b> Clear the depth target
<a href="#">rsgDrawMesh</a>	<b>Deprecated.</b> Draw a mesh
<a href="#">rsgDrawQuad</a>	<b>Deprecated.</b> Draw a quad
<a href="#">rsgDrawQuadTexCoords</a>	<b>Deprecated.</b> Draw a textured quad
<a href="#">rsgDrawRect</a>	<b>Deprecated.</b> Draw a rectangle
<a href="#">rsgDrawSpriteScreenspace</a>	<b>Deprecated.</b> Draw rectangles in screenspace
<a href="#">rsgDrawText</a>	<b>Deprecated.</b> Draw a text string
<a href="#">rsgFinish</a>	<b>Deprecated.</b> End rendering commands
<a href="#">rsgFontColor</a>	<b>Deprecated.</b> Set the font color
<a href="#">rsgGetHeight</a>	<b>Deprecated.</b> Get the surface height
<a href="#">rsgGetWidth</a>	<b>Deprecated.</b> Get the surface width
<a href="#">rsgMeasureText</a>	<b>Deprecated.</b> Get the bounding box for a text string
<a href="#">rsgMeshComputeBoundingBox</a>	<b>Deprecated.</b> Compute a bounding box
<a href="#">rsgMeshGetIndexAllocation</a>	<b>Deprecated.</b> Return an allocation containing index data
<a href="#">rsgMeshGetPrimitive</a>	<b>Deprecated.</b> Return the primitive
<a href="#">rsgMeshGetPrimitiveCount</a>	<b>Deprecated.</b> Return the number of index sets
<a href="#">rsgMeshGetVertexAllocation</a>	<b>Deprecated.</b> Return a vertex allocation
<a href="#">rsgMeshGetVertexAllocationCount</a>	<b>Deprecated.</b> Return the number of vertex allocations
<a href="#">rsgProgramFragmentConstantColor</a>	<b>Deprecated.</b> Set the constant color for a fixed function emulation program
<a href="#">rsgProgramRasterGetCullMode</a>	<b>Deprecated.</b> Get program raster cull mode
<a href="#">rsgProgramRasterIsPointSpriteEnabled</a>	<b>Deprecated.</b> Get program raster point sprite state
<a href="#">rsgProgramStoreGetBlendDstFunc</a>	<b>Deprecated.</b> Get program store blend destination function
<a href="#">rsgProgramStoreGetBlendSrcFunc</a>	<b>Deprecated.</b> Get program store blend source function
<a href="#">rsgProgramStoreGetDepthFunc</a>	<b>Deprecated.</b> Get program store depth function
<a href="#">rsgProgramStoreIsColorMaskAlphaEnabled</a>	<b>Deprecated.</b> Get program store alpha component color mask
<a href="#">rsgProgramStoreIsColorMaskBlueEnabled</a>	<b>Deprecated.</b> Get program store blue component color mask
<a href="#">rsgProgramStoreIsColorMaskGreenEnabled</a>	<b>Deprecated.</b> Get program store green component color mask

<del>rsgProgramStoreIsColorMaskRedEnabled</del>	<b>Deprecated.</b> Get program store red component color mask
<del>rsgProgramStoreIsDepthMaskEnabled</del>	<b>Deprecated.</b> Get program store depth mask
rsgProgramStoreIsDitherEnabled	<b>Deprecated.</b> Get program store dither state
rsgProgramVertexGetProjectionMatrix	<b>Deprecated.</b> Get the projection matrix for a fixed function vertex program
rsgProgramVertexLoadModelMatrix	<b>Deprecated.</b> Load the model matrix for a bound fixed function vertex program
rsgProgramVertexLoadProjectionMatrix	<b>Deprecated.</b> Load the projection matrix for a bound fixed function vertex program
rsgProgramVertexLoadTextureMatrix	<b>Deprecated.</b> Load the texture matrix for a bound fixed function vertex program