RenderScript Allocation Data Access Functions

Overview

The functions below can be used to get and set the cells that comprise an allocation.

- Individual cells are accessed using the rsGetElementAt* and rsSetElementAt functions.
- Multiple cells can be copied using the rsAllocationCopy* and rsAllocationV* functions.
- For getting values through a sampler, use rsSample.

The rsGetElementAt and rsSetElement* functions are somewhat misnamed. They don't get or set elements, which are akin to data types; they get or set cells. Think of them as rsGetCellAt and and rsSetCellAt.

Summary

Functions	
rsAllocationCopy1DRange	Copy consecutive cells between allocations
rsAllocationCopy2DRange	Copy a rectangular region of cells between allocations
rsAllocationVLoadX	Get a vector from an allocation of scalars
rsAllocationVStoreX	Store a vector into an allocation of scalars
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
rsSample	Sample a value from a texture allocation
rsSetElementAt	Set a cell of an allocation

Functions

rsAllocationCopy1DRange : Copy consecutive cells between allocations

void rsAllocationCopy1DRange(rs_allocation dstAlloc, uint32_t dstOff, uint32_t dstMip, uint32_t count, rs_allocation srcAlloc, uint32_t srcOff, uint32_t srcMip);

Added in API level 14

Parameters

dstAlloc	Allocation to copy cells into.
dstOff	Offset in the destination of the first cell to be copied into.
dstMip	Mip level in the destination allocation. 0 if mip mapping is not used.
count	Number of cells to be copied.
srcAlloc	Source allocation.
srcOff	Offset in the source of the first cell to be copied.
srcMip	Mip level in the source allocation. 0 if mip mapping is not used.

Copies the specified number of cells from one allocation to another.

The two allocations must be different. Using this function to copy whithin the same allocation yields undefined results.

The function does not validate whether the offset plus count exceeds the size of either allocation. Be careful!

This function should only be called between 1D allocations. Calling it on other allocations is undefined.

This function should not be called from inside a kernel, or from any function that may be called directly or indirectly from a kernel. Doing so would cause a runtime error.

rsAllocationCopy2DRange : Copy a rectangular region of cells between allocations

void rsAllocationCopy2DRange(rs_allocation dstAlloc, uint32_t dstXoff, uint32_t dstYoff, uint32_t dstMip, rs_allocation_cubemap_face dstFace, uint32_t width, uint32_t height, rs_allocation srcAlloc, uint32_t srcXoff, uint32_t srcYoff, uint32_t srcMip, rs_allocation_cubemap_face srcFace);

Added in API level 14

Parameters

dstAlloc	Allocation to copy cells into.
dstXoff	X offset in the destination of the region to be set.
dstYoff	Y offset in the destination of the region to be set.
dstMip	Mip level in the destination allocation. 0 if mip mapping is not used.
dstFace	Cubemap face of the destination allocation. Ignored for allocations that aren't cubemaps.
width	Width of the incoming region to update.
height	Height of the incoming region to update.
srcAlloc	Source allocation.
srcXoff	X offset in the source.
srcYoff	Y offset in the source.
srcMip	Mip level in the source allocation. 0 if mip mapping is not used.
srcFace	Cubemap face of the source allocation. Ignored for allocations that aren't cubemaps.

Copies a rectangular region of cells from one allocation to another. (width * heigth) cells are copied.

The two allocations must be different. Using this function to copy whithin the same allocation yields undefined results.

The function does not validate whether the the source or destination region exceeds the size of its respective allocation. Be careful!

This function should only be called between 2D allocations. Calling it on other allocations is undefined.

This function should not be called from inside a kernel, or from any function that may be called directly or indirectly from a kernel. Doing so would cause a runtime error.

rsAllocationVLoadX : Get a vector from an allocation of scalars

char2 rsAllocationVLoadX_char2(rs_allocation a, uint32_t x);	Added in API level 22
char2 rsAllocationVLoadX_char2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
char2 rsAllocationVLoadX_char2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
char3 rsAllocationVLoadX_char3(rs_allocation a, uint32_t x);	Added in API level 22
char3 rsAllocationVLoadX_char3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
char3 rsAllocationVLoadX_char3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
char4 rsAllocationVLoadX_char4(rs_allocation a, uint32_t x);	Added in API level 22
char4 rsAllocationVLoadX_char4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
char4 rsAllocationVLoadX_char4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
double2 rsAllocationVLoadX_double2(rs_allocation a, uint32_t x);	Added in API level 22
double2 rsAllocationVLoadX_double2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22

double2 rsAllocationVLoadX_double2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
double3 rsAllocationVLoadX_double3(rs_allocation a, uint32_t x);	Added in API level 22
double3 rsAllocationVLoadX_double3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
double3 rsAllocationVLoadX_double3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
double4 rsAllocationVLoadX_double4(rs_allocation a, uint32_t x);	Added in API level 22
double4 rsAllocationVLoadX_double4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
double4 rsAllocationVLoadX_double4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
float2 rsAllocationVLoadX_float2(rs_allocation a, uint32_t x);	Added in API level 22
float2 rsAllocationVLoadX_float2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
float2 rsAllocationVLoadX_float2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
float3 rsAllocationVLoadX_float3(rs_allocation a, uint32_t x);	Added in API level 22
float3 rsAllocationVLoadX_float3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
float3 rsAllocationVLoadX_float3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
float4 rsAllocationVLoadX_float4(rs_allocation a, uint32_t x);	Added in API level 22
float4 rsAllocationVLoadX_float4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
float4 rsAllocationVLoadX_float4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
int2 rsAllocationVLoadX_int2(rs_allocation a, uint32_t x);	Added in API level 22
int2 rsAllocationVLoadX_int2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
int2 rsAllocationVLoadX_int2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
int3 rsAllocationVLoadX_int3(rs_allocation a, uint32_t x);	Added in API level 22
int3 rsAllocationVLoadX_int3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
int3 rsAllocationVLoadX_int3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
int4 rsAllocationVLoadX_int4(rs_allocation a, uint32_t x);	Added in API level 22
int4 rsAllocationVLoadX_int4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
int4 rsAllocationVLoadX_int4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
long2 rsAllocationVLoadX_long2(rs_allocation a, uint32_t x);	Added in API level 22
long2 rsAllocationVLoadX_long2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
long2 rsAllocationVLoadX_long2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
long3 rsAllocationVLoadX_long3(rs_allocation a, uint32_t x);	Added in API level 22
long3 rsAllocationVLoadX_long3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
long3 rsAllocationVLoadX_long3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
long4 rsAllocationVLoadX_long4(rs_allocation a, uint32_t x);	Added in API level 22
long4 rsAllocationVLoadX_long4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
long4 rsAllocationVLoadX_long4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
short2 rsAllocationVLoadX_short2(rs_allocation a, uint32_t x);	Added in API level 22
short2 rsAllocationVLoadX_short2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
short2 rsAllocationVLoadX_short2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
short3 rsAllocationVLoadX_short3(rs_allocation a, uint32_t x);	Added in API level 22
short3 rsAllocationVLoadX_short3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
short3 rsAllocationVLoadX_short3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
short4 rsAllocationVLoadX_short4(rs_allocation a, uint32_t x);	Added in API level 22
short4 rsAllocationVLoadX_short4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
short4 rsAllocationVLoadX_short4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uchar2 rsAllocationVLoadX_uchar2(rs_allocation a, uint32_t x);	Added in API level 22
uchar2 rsAllocationVLoadX_uchar2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uchar2 rsAllocationVLoadX_uchar2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uchar3 rsAllocationVLoadX_uchar3(rs_allocation a, uint32_t x);	Added in API level 22

uchar3 rsAllocationVLoadX_uchar3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uchar3 rsAllocationVLoadX_uchar3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uchar4 rsAllocationVLoadX_uchar4(rs_allocation a, uint32_t x);	Added in API level 22
uchar4 rsAllocationVLoadX_uchar4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uchar4 rsAllocationVLoadX_uchar4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uint2 rsAllocationVLoadX_uint2(rs_allocation a, uint32_t x);	Added in API level 22
uint2 rsAllocationVLoadX_uint2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uint2 rsAllocationVLoadX_uint2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uint3 rsAllocationVLoadX_uint3(rs_allocation a, uint32_t x);	Added in API level 22
uint3 rsAllocationVLoadX_uint3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uint3 rsAllocationVLoadX_uint3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
uint4 rsAllocationVLoadX_uint4(rs_allocation a, uint32_t x);	Added in API level 22
uint4 rsAllocationVLoadX_uint4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
uint4 rsAllocationVLoadX_uint4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ulong2 rsAllocationVLoadX_ulong2(rs_allocation a, uint32_t x);	Added in API level 22
ulong2 rsAllocationVLoadX_ulong2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ulong2 rsAllocationVLoadX_ulong2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ulong3 rsAllocationVLoadX_ulong3(rs_allocation a, uint32_t x);	Added in API level 22
ulong3 rsAllocationVLoadX_ulong3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ulong3 rsAllocationVLoadX_ulong3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ulong4 rsAllocationVLoadX_ulong4(rs_allocation a, uint32_t x);	Added in API level 22
ulong4 rsAllocationVLoadX_ulong4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ulong4 rsAllocationVLoadX_ulong4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ushort2 rsAllocationVLoadX_ushort2(rs_allocation a, uint32_t x);	Added in API level 22
ushort2 rsAllocationVLoadX_ushort2(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ushort2 rsAllocationVLoadX_ushort2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ushort3 rsAllocationVLoadX_ushort3(rs_allocation a, uint32_t x);	Added in API level 22
ushort3 rsAllocationVLoadX_ushort3(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ushort3 rsAllocationVLoadX_ushort3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
ushort4 rsAllocationVLoadX_ushort4(rs_allocation a, uint32_t x);	Added in API level 22
ushort4 rsAllocationVLoadX_ushort4(rs_allocation a, uint32_t x, uint32_t y);	Added in API level 22
ushort4 rsAllocationVLoadX_ushort4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22

Parameters

- *a* Allocation to get the data from.
- *x* X offset in the allocation of the first cell to be copied from.
- *y* Y offset in the allocation of the first cell to be copied from.
- *z* Z offset in the allocation of the first cell to be copied from.

This function returns a vector composed of successive cells of the allocation. It assumes that the allocation contains scalars.

The "X" in the name indicates that successive values are extracted by increasing the X index. There are currently no functions to get successive values incrementing other dimensions. Use multiple calls to rsGetElementAt() instead.

For example, when calling rsAllocationVLoadX_int4(a, 20, 30), an int4 composed of a[20, 30], a[21, 30], a[22, 30], and a[23, 30] is returned.

When retrieving from a three dimensional allocations, use the x, y, z variant. Similarly, use the x, y variant for two dimensional allocations and x for the mono dimensional allocations.

For efficiency, this function does not validate the inputs. Trying to wrap the X index, exceeding the size of the allocation, or using indices incompatible with the dimensionality of the allocation yields undefined results.

See also rsAllocationVStoreX().

rsAllocationVStoreX : Store a vector into an allocation of scalars

void rsAllocationVStoreX_char2(rs_allocation a, char2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_char2(rs_allocation a, char2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_char2(rs_allocation a, char2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_char3(rs_allocation a, char3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_char3(rs_allocation a, char3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_char3(rs_allocation a, char3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_char4(rs_allocation a, char4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_char4(rs_allocation a, char4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_char4(rs_allocation a, char4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_double2(rs_allocation a, double2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_double2(rs_allocation a, double2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_double2(rs_allocation a, double2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_double3(rs_allocation a, double3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_double3(rs_allocation a, double3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_double3(rs_allocation a, double3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_double4(rs_allocation a, double4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_double4(rs_allocation a, double4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_double4(rs_allocation a, double4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_float2(rs_allocation a, float2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_float2(rs_allocation a, float2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_float2(rs_allocation a, float2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_float3(rs_allocation a, float3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_float3(rs_allocation a, float3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_float3(rs_allocation a, float3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_float4(rs_allocation a, float4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_float4(rs_allocation a, float4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_float4(rs_allocation a, float4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_int2(rs_allocation a, int2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_int2(rs_allocation a, int2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_int2(rs_allocation a, int2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_int3(rs_allocation a, int3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_int3(rs_allocation a, int3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_int3(rs_allocation a, int3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_int4(rs_allocation a, int4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_int4(rs_allocation a, int4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_int4(rs_allocation a, int4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_long2(rs_allocation a, long2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_long2(rs_allocation a, long2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_long2(rs_allocation a, long2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_long3(rs_allocation a, long3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_long3(rs_allocation a, long3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_long3(rs_allocation a, long3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_long4(rs_allocation a, long4 val, uint32_t x);	Added in API level 22

void rsAllocationVStoreX_long4(rs_allocation a, long4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_long4(rs_allocation a, long4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_short2(rs_allocation a, short2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_short2(rs_allocation a, short2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_short2(rs_allocation a, short2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_short3(rs_allocation a, short3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_short3(rs_allocation a, short3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_short3(rs_allocation a, short3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_short4(rs_allocation a, short4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_short4(rs_allocation a, short4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_short4(rs_allocation a, short4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uchar2(rs_allocation a, uchar2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uchar2(rs_allocation a, uchar2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uchar2(rs_allocation a, uchar2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uchar3(rs_allocation a, uchar3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uchar3(rs_allocation a, uchar3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uchar3(rs_allocation a, uchar3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uchar4(rs_allocation a, uchar4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uchar4(rs_allocation a, uchar4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uchar4(rs_allocation a, uchar4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uint2(rs_allocation a, uint2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uint2(rs_allocation a, uint2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uint2(rs_allocation a, uint2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uint3(rs_allocation a, uint3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uint3(rs_allocation a, uint3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uint3(rs_allocation a, uint3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_uint4(rs_allocation a, uint4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_uint4(rs_allocation a, uint4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_uint4(rs_allocation a, uint4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ulong2(rs_allocation a, ulong2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ulong2(rs_allocation a, ulong2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ulong2(rs_allocation a, ulong2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ulong3(rs_allocation a, ulong3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ulong3(rs_allocation a, ulong3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ulong3(rs_allocation a, ulong3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ulong4(rs_allocation a, ulong4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ulong4(rs_allocation a, ulong4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ulong4(rs_allocation a, ulong4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ushort2(rs_allocation a, ushort2 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ushort2(rs_allocation a, ushort2 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ushort2(rs_allocation a, ushort2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ushort3(rs_allocation a, ushort3 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ushort3(rs_allocation a, ushort3 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ushort3(rs_allocation a, ushort3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22
void rsAllocationVStoreX_ushort4(rs_allocation a, ushort4 val, uint32_t x);	Added in API level 22
void rsAllocationVStoreX_ushort4(rs_allocation a, ushort4 val, uint32_t x, uint32_t y);	Added in API level 22
void rsAllocationVStoreX_ushort4(rs_allocation a, ushort4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 22

Parameters

a	Allocation to store the data into.
val	Value to be stored.
x	X offset in the allocation of the first cell to be copied into.
y	Y offset in the allocation of the first cell to be copied into.
z	Z offset in the allocation of the first cell to be copied into.

This function stores the entries of a vector into successive cells of an allocation. It assumes that the allocation contains scalars.

The "X" in the name indicates that successive values are stored by increasing the X index. There are currently no functions to store successive values incrementing other dimensions. Use multiple calls to rsSetElementAt() instead.

For example, when calling rsAllocationVStoreX_int3(a, v, 20, 30), v.x is stored at a[20, 30], v.y at a[21, 30], and v.z at a[22, 30].

When storing into a three dimensional allocations, use the x, y, z variant. Similarly, use the x, y variant for two dimensional allocations and x for the mono dimensional allocations.

For efficiency, this function does not validate the inputs. Trying to wrap the X index, exceeding the size of the allocation, or using indices incompatible with the dimensionality of the allocation yiels undefined results.

See also rsAllocationVLoadX().

rsGetElementAt : Return a cell from an allocation

char rsGetElementAt_char(rs_allocation a, uint32_t x);

char rsGetElementAt_char(rs_allocation a, uint32_t x, uint32_t y);

char rsGetElementAt_char(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

char2 rsGetElementAt_char2(rs_allocation a, uint32_t x);

char2 rsGetElementAt_char2(rs_allocation a, uint32_t x, uint32_t y);

char2 rsGetElementAt_char2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

char3 rsGetElementAt_char3(rs_allocation a, uint32_t x);

char3 rsGetElementAt_char3(rs_allocation a, uint32_t x, uint32_t y);

char3 rsGetElementAt_char3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

char4 rsGetElementAt_char4(rs_allocation a, uint32_t x);

char4 rsGetElementAt_char4(rs_allocation a, uint32_t x, uint32_t y);

char4 rsGetElementAt_char4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

const void* rsGetElementAt(rs_allocation a, uint32_t x);

const void* rsGetElementAt(rs_allocation a, uint32_t x, uint32_t y);

const void* rsGetElementAt(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

double rsGetElementAt_double(rs_allocation a, uint32_t x);

double rsGetElementAt_double(rs_allocation a, uint32_t x, uint32_t y);

double rsGetElementAt_double(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

double2 rsGetElementAt_double2(rs_allocation a, uint32_t x);

double2 rsGetElementAt_double2(rs_allocation a, uint32_t x, uint32_t y);

double2 rsGetElementAt_double2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

double3 rsGetElementAt_double3(rs_allocation a, uint32_t x);

double3 rsGetElementAt_double3(rs_allocation a, uint32_t x, uint32_t y);

double3 rsGetElementAt_double3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

double4 rsGetElementAt_double4(rs_allocation a, uint32_t x);

double4 rsGetElementAt_double4(rs_allocation a, uint32_t x, uint32_t y);

double4 rsGetElementAt_double4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

float rsGetElementAt_float(rs_allocation a, uint32_t x);

float rsGetElementAt_float(rs_allocation a, uint32_t x, uint32_t y);

float rsGetElementAt_float(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

float2 rsGetElementAt_float2(rs_allocation a, uint32_t x);

float2 rsGetElementAt_float2(rs_allocation a, uint32_t x, uint32_t y);

float2 rsGetElementAt_float2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

float3 rsGetElementAt_float3(rs_allocation a, uint32_t x);

float3 rsGetElementAt_float3(rs_allocation a, uint32_t x, uint32_t y);

float3 rsGetElementAt_float3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

float4 rsGetElementAt_float4(rs_allocation a, uint32_t x);

float4 rsGetElementAt_float4(rs_allocation a, uint32_t x, uint32_t y);

float4 rsGetElementAt_float4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

half rsGetElementAt_half(rs_allocation a, uint32_t x); Added in API level 23

half rsGetElementAt_half(rs_allocation a, uint32_t x, uint32_t y); Added in API level 23

half rsGetElementAt_half(rs_allocation a, uint32_t x, uint32_t y, uint32_t z); Added in API level 23

half2 rsGetElementAt_half2(rs_allocation a, uint32_t x); Added in API level 23

half2 rsGetElementAt_half2(rs_allocation a, uint32_t x, uint32_t y); Added in API level 23

half2 rsGetElementAt_half2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z); Added in API level 23

half3 rsGetElementAt_half3(rs_allocation a, uint32_t x); Added in API level 23

half3 rsGetElementAt_half3(rs_allocation a, uint32_t x, uint32_t y); Added in API level 23

half3 rsGetElementAt_half3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z); Added in API level 23

half4 rsGetElementAt_half4(rs_allocation a, uint32_t x); Added in API level 23

half4 rsGetElementAt_half4(rs_allocation a, uint32_t x, uint32_t y); Added in API level 23

half4 rsGetElementAt_half4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z); Added in API level 23

int rsGetElementAt_int(rs_allocation a, uint32_t x);

int rsGetElementAt_int(rs_allocation a, uint32_t x, uint32_t y);

int rsGetElementAt_int(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

int2 rsGetElementAt_int2(rs_allocation a, uint32_t x);

int2 rsGetElementAt_int2(rs_allocation a, uint32_t x, uint32_t y);

int2 rsGetElementAt_int2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

int3 rsGetElementAt_int3(rs_allocation a, uint32_t x);

int3 rsGetElementAt_int3(rs_allocation a, uint32_t x, uint32_t y);

int3 rsGetElementAt_int3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

int4 rsGetElementAt_int4(rs_allocation a, uint32_t x);

int4 rsGetElementAt_int4(rs_allocation a, uint32_t x, uint32_t y);

int4 rsGetElementAt_int4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

long rsGetElementAt_long(rs_allocation a, uint32_t x);

long rsGetElementAt_long(rs_allocation a, uint32_t x, uint32_t y);

long rsGetElementAt_long(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

long2 rsGetElementAt_long2(rs_allocation a, uint32_t x);

long2 rsGetElementAt_long2(rs_allocation a, uint32_t x, uint32_t y);

long2 rsGetElementAt_long2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

long3 rsGetElementAt_long3(rs_allocation a, uint32_t x);

long3 rsGetElementAt_long3(rs_allocation a, uint32_t x, uint32_t y);

long3 rsGetElementAt_long3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

long4 rsGetElementAt_long4(rs_allocation a, uint32_t x);

long4 rsGetElementAt_long4(rs_allocation a, uint32_t x, uint32_t y);

long4 rsGetElementAt_long4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);
short rsGetElementAt_short(rs_allocation a, uint32_t x);

short rsGetElementAt_short(rs_allocation a, uint32_t x, uint32_t y);

short rsGetElementAt_short(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

short2 rsGetElementAt_short2(rs_allocation a, uint32_t x);

short2 rsGetElementAt_short2(rs_allocation a, uint32_t x, uint32_t y);

short2 rsGetElementAt_short2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

short3 rsGetElementAt_short3(rs_allocation a, uint32_t x);

short3 rsGetElementAt_short3(rs_allocation a, uint32_t x, uint32_t y);

short3 rsGetElementAt_short3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

short4 rsGetElementAt_short4(rs_allocation a, uint32_t x);

short4 rsGetElementAt_short4(rs_allocation a, uint32_t x, uint32_t y);

short4 rsGetElementAt_short4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uchar rsGetElementAt_uchar(rs_allocation a, uint32_t x);

uchar rsGetElementAt_uchar(rs_allocation a, uint32_t x, uint32_t y);

uchar rsGetElementAt_uchar(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uchar2 rsGetElementAt_uchar2(rs_allocation a, uint32_t x);

uchar2 rsGetElementAt_uchar2(rs_allocation a, uint32_t x, uint32_t y);

uchar2 rsGetElementAt_uchar2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uchar3 rsGetElementAt_uchar3(rs_allocation a, uint32_t x);

uchar3 rsGetElementAt_uchar3(rs_allocation a, uint32_t x, uint32_t y);

uchar3 rsGetElementAt_uchar3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uchar4 rsGetElementAt_uchar4(rs_allocation a, uint32_t x);

uchar4 rsGetElementAt_uchar4(rs_allocation a, uint32_t x, uint32_t y);

uchar4 rsGetElementAt_uchar4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uint rsGetElementAt_uint(rs_allocation a, uint32_t x);

uint rsGetElementAt_uint(rs_allocation a, uint32_t x, uint32_t y);

uint rsGetElementAt_uint(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uint2 rsGetElementAt_uint2(rs_allocation a, uint32_t x);

uint2 rsGetElementAt_uint2(rs_allocation a, uint32_t x, uint32_t y);

uint2 rsGetElementAt_uint2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uint3 rsGetElementAt_uint3(rs_allocation a, uint32_t x);

uint3 rsGetElementAt_uint3(rs_allocation a, uint32_t x, uint32_t y);

uint3 rsGetElementAt_uint3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

uint4 rsGetElementAt_uint4(rs_allocation a, uint32_t x);

uint4 rsGetElementAt_uint4(rs_allocation a, uint32_t x, uint32_t y);

uint4 rsGetElementAt_uint4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ulong rsGetElementAt_ulong(rs_allocation a, uint32_t x);

ulong rsGetElementAt_ulong(rs_allocation a, uint32_t x, uint32_t y);

ulong rsGetElementAt_ulong(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ulong2 rsGetElementAt_ulong2(rs_allocation a, uint32_t x);

ulong2 rsGetElementAt_ulong2(rs_allocation a, uint32_t x, uint32_t y);

ulong2 rsGetElementAt_ulong2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ulong3 rsGetElementAt_ulong3(rs_allocation a, uint32_t x);

ulong3 rsGetElementAt_ulong3(rs_allocation a, uint32_t x, uint32_t y);

ulong3 rsGetElementAt_ulong3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ulong4 rsGetElementAt_ulong4(rs_allocation a, uint32_t x);

ulong4 rsGetElementAt_ulong4(rs_allocation a, uint32_t x, uint32_t y);

ulong4 rsGetElementAt_ulong4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ushort rsGetElementAt_ushort(rs_allocation a, uint32_t x);

ushort rsGetElementAt_ushort(rs_allocation a, uint32_t x, uint32_t y);

ushort rsGetElementAt_ushort(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ushort2 rsGetElementAt_ushort2(rs_allocation a, uint32_t x);

ushort2 rsGetElementAt_ushort2(rs_allocation a, uint32_t x, uint32_t y);

ushort2 rsGetElementAt_ushort2(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ushort3 rsGetElementAt_ushort3(rs_allocation a, uint32_t x);

ushort3 rsGetElementAt_ushort3(rs_allocation a, uint32_t x, uint32_t y);

ushort3 rsGetElementAt_ushort3(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

ushort4 rsGetElementAt_ushort4(rs_allocation a, uint32_t x);

ushort4 rsGetElementAt_ushort4(rs_allocation a, uint32_t x, uint32_t y);

ushort4 rsGetElementAt_ushort4(rs_allocation a, uint32_t x, uint32_t y, uint32_t z);

This function extracts a single cell from an allocation.

When retrieving from a three dimensional allocations, use the x, y, z variant. Similarly, use the x, y variant for two dimensional allocations and x for the mono dimensional allocations.

This function has two styles. One returns the address of the value using a void*, the other returns the actual value, e.g. rsGetElementAt() vs. rsGetElementAt_int4(). For primitive types, always use the latter as it is more efficient.

rsGetElementAtYuv_uchar_U : Get the U component of an allocation of YUVs

uchar rsGetElementAtYuv_uchar_U(rs_allocation a, uint32_t x, uint32_t y); Added in API level 18

Extracts the U component of a single YUV value from a 2D allocation of YUVs.

Inside an allocation, Y, U, and V components may be stored if different planes and at different resolutions. The x, y coordinates provided here are in the dimensions of the Y plane.

See rsGetElementAtYuv_uchar_Y().

rsGetElementAtYuv_uchar_V : Get the V component of an allocation of YUVs

uchar rsGetElementAtYuv_uchar_V(rs_allocation a, uint32_t x, uint32_t y); Added in API level 18

Extracts the V component of a single YUV value from a 2D allocation of YUVs.

Inside an allocation, Y, U, and V components may be stored if different planes and at different resolutions. The x, y coordinates provided here are in the dimensions of the Y plane.

See rsGetElementAtYuv_uchar_Y().

rsGetElementAtYuv_uchar_Y : Get the Y component of an allocation of YUVs

uchar rsGetElementAtYuv_uchar_Y(rs_allocation a, uint32_t x, uint32_t y); Added in API level 18

Extracts the Y component of a single YUV value from a 2D allocation of YUVs.

Inside an allocation, Y, U, and V components may be stored if different planes and at different resolutions. The x, y coordinates provided here are in the dimensions of the Y plane.

See rsGetElementAtYuv_uchar_U() and rsGetElementAtYuv_uchar_V().

rsSample : Sample a value from a texture allocation

float4 rsSample(rs_allocation a, rs_sampler s, float location); Added in API level 16

float4 rsSample(rs_allocation a, rs_sampler s, float location, float lod); Added in API level 16

float4 rsSample(rs_allocation a, rs_sampler s, float2 location); Added in API level 16

float4 rsSample(rs_allocation a, rs_sampler s, float2 location, float lod); Added in API level 16

Parameters

a	Allocation to sample from.
s	Sampler state.
location	Location to sample from.
lod	Mip level to sample from, for fractional values mip levels will be interpolated if RS_SAMPLER_LINEAR_MIP_LINEAR is used.

Fetches a value from a texture allocation in a way described by the sampler.

If your allocation is 1D, use the variant with float for location. For 2D, use the float2 variant.

See android.renderscript.Sampler for more details.

rsSetElementAt : Set a cell of an allocation

void rsSetElementAt(rs_allocation a, void* ptr, uint32_t x); Added in API level 18

void rsSetElementAt(rs_allocation a, void* ptr, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_char(rs_allocation a, char val, uint32_t x); Added in API level 18

void rsSetElementAt_char(rs_allocation a, char val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_char(rs_allocation a, char val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_char2(rs_allocation a, char2 val, uint32_t x); Added in API level 18

void rsSetElementAt_char2(rs_allocation a, char2 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_char2(rs_allocation a, char2 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_char3(rs_allocation a, char3 val, uint32_t x); Added in API level 18

void rsSetElementAt_char3(rs_allocation a, char3 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_char3(rs_allocation a, char3 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_char4(rs_allocation a, char4 val, uint32_t x); Added in API level 18

void rsSetElementAt_char4(rs_allocation a, char4 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_char4(rs_allocation a, char4 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_double(rs_allocation a, double val, uint32_t x); Added in API level 18

void rsSetElementAt_double(rs_allocation a, double val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_double(rs_allocation a, double val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_double2(rs_allocation a, double2 val, uint32_t x); Added in API level 18

void rsSetElementAt_double2(rs_allocation a, double2 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_double2(rs_allocation a, double2 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_double3(rs_allocation a, double3 val, uint32_t x); Added in API level 18

void rsSetElementAt_double3(rs_allocation a, double3 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_double3(rs_allocation a, double3 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_double4(rs_allocation a, double4 val, uint32_t x); Added in API level 18

void rsSetElementAt_double4(rs_allocation a, double4 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_double4(rs_allocation a, double4 val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_float(rs_allocation a, float val, uint32_t x); Added in API level 18

void rsSetElementAt_float(rs_allocation a, float val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_float(rs_allocation a, float val, uint32_t x, uint32_t y, uint32_t z); Added in API level 18

void rsSetElementAt_float2(rs_allocation a, float2 val, uint32_t x); Added in API level 18

void rsSetElementAt_float2(rs_allocation a, float2 val, uint32_t x, uint32_t y); Added in API level 18

void rsSetElementAt_float2(rs_allocation a, float2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_float3(rs_allocation a, float3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_float3(rs_allocation a, float3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_float3(rs_allocation a, float3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_float4(rs_allocation a, float4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_float4(rs_allocation a, float4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_float4(rs_allocation a, float4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_half(rs_allocation a, half val, uint32_t x);	Added in API level 23
void rsSetElementAt_half(rs_allocation a, half val, uint32_t x, uint32_t y);	Added in API level 23
void rsSetElementAt_half(rs_allocation a, half val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 23
void rsSetElementAt_half2(rs_allocation a, half2 val, uint32_t x);	Added in API level 23
void rsSetElementAt_half2(rs_allocation a, half2 val, uint32_t x, uint32_t y);	Added in API level 23
void rsSetElementAt_half2(rs_allocation a, half2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 23
void rsSetElementAt_half3(rs_allocation a, half3 val, uint32_t x);	Added in API level 23
void rsSetElementAt_half3(rs_allocation a, half3 val, uint32_t x, uint32_t y);	Added in API level 23
void rsSetElementAt_half3(rs_allocation a, half3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 23
void rsSetElementAt_half4(rs_allocation a, half4 val, uint32_t x);	Added in API level 23
void rsSetElementAt_half4(rs_allocation a, half4 val, uint32_t x, uint32_t y);	Added in API level 23
void rsSetElementAt_half4(rs_allocation a, half4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 23
void rsSetElementAt_int(rs_allocation a, int val, uint32_t x);	Added in API level 18
void rsSetElementAt_int(rs_allocation a, int val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_int(rs_allocation a, int val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_int2(rs_allocation a, int2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_int2(rs_allocation a, int2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_int2(rs_allocation a, int2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_int3(rs_allocation a, int3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_int3(rs_allocation a, int3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_int3(rs_allocation a, int3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_int4(rs_allocation a, int4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_int4(rs_allocation a, int4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_int4(rs_allocation a, int4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_long(rs_allocation a, long val, uint32_t x);	Added in API level 18
void rsSetElementAt_long(rs_allocation a, long val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_long(rs_allocation a, long val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_long2(rs_allocation a, long2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_long2(rs_allocation a, long2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_long2(rs_allocation a, long2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_long3(rs_allocation a, long3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_long3(rs_allocation a, long3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_long3(rs_allocation a, long3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_long4(rs_allocation a, long4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_long4(rs_allocation a, long4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_long4(rs_allocation a, long4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_short(rs_allocation a, short val, uint32_t x);	Added in API level 18
void rsSetElementAt_short(rs_allocation a, short val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_short(rs_allocation a, short val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18

void rsSetElementAt_short2(rs_allocation a, short2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_short2(rs_allocation a, short2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_short2(rs_allocation a, short2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_short3(rs_allocation a, short3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_short3(rs_allocation a, short3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_short3(rs_allocation a, short3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_short4(rs_allocation a, short4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_short4(rs_allocation a, short4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_short4(rs_allocation a, short4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uchar(rs_allocation a, uchar val, uint32_t x);	Added in API level 18
void rsSetElementAt_uchar(rs_allocation a, uchar val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uchar(rs_allocation a, uchar val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uchar2(rs_allocation a, uchar2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uchar2(rs_allocation a, uchar2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uchar2(rs_allocation a, uchar2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uchar3(rs_allocation a, uchar3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uchar3(rs_allocation a, uchar3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uchar3(rs_allocation a, uchar3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uchar4(rs_allocation a, uchar4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uchar4(rs_allocation a, uchar4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uchar4(rs_allocation a, uchar4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uint(rs_allocation a, uint val, uint32_t x);	Added in API level 18
void rsSetElementAt_uint(rs_allocation a, uint val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uint(rs_allocation a, uint val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uint2(rs_allocation a, uint2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uint2(rs_allocation a, uint2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uint2(rs_allocation a, uint2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uint3(rs_allocation a, uint3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uint3(rs_allocation a, uint3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uint3(rs_allocation a, uint3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_uint4(rs_allocation a, uint4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_uint4(rs_allocation a, uint4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_uint4(rs_allocation a, uint4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ulong(rs_allocation a, ulong val, uint32_t x);	Added in API level 18
void rsSetElementAt_ulong(rs_allocation a, ulong val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ulong(rs_allocation a, ulong val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ulong2(rs_allocation a, ulong2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ulong2(rs_allocation a, ulong2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ulong2(rs_allocation a, ulong2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ulong3(rs_allocation a, ulong3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ulong3(rs_allocation a, ulong3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ulong3(rs_allocation a, ulong3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ulong4(rs_allocation a, ulong4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ulong4(rs_allocation a, ulong4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ulong4(rs_allocation a, ulong4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ushort(rs_allocation a, ushort val, uint32_t x);	Added in API level 18
void rsSetElementAt_ushort(rs_allocation a, ushort val, uint32_t x, uint32_t y);	Added in API level 18

void rsSetElementAt_ushort(rs_allocation a, ushort val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ushort2(rs_allocation a, ushort2 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ushort2(rs_allocation a, ushort2 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ushort2(rs_allocation a, ushort2 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ushort3(rs_allocation a, ushort3 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ushort3(rs_allocation a, ushort3 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ushort3(rs_allocation a, ushort3 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18
void rsSetElementAt_ushort4(rs_allocation a, ushort4 val, uint32_t x);	Added in API level 18
void rsSetElementAt_ushort4(rs_allocation a, ushort4 val, uint32_t x, uint32_t y);	Added in API level 18
void rsSetElementAt_ushort4(rs_allocation a, ushort4 val, uint32_t x, uint32_t y, uint32_t z);	Added in API level 18

This function stores a value into a single cell of an allocation.

When storing into a three dimensional allocations, use the x, y, z variant. Similarly, use the x, y variant for two dimensional allocations and x for the mono dimensional allocations.

This function has two styles. One passes the value to be stored using a void*, the other has the actual value as an argument, e.g. rsSetElementAt() vs. rsSetElementAt_int4(). For primitive types, always use the latter as it is more efficient.

See also rsGetElementAt().