**rocSPARSELt Data Types**

**Data Structures**

**rocsparselt\_handle**

The structure holds the rocSPARSELt library context (device properties, system information, etc.).

The handle must be initialized and destroyed with rocsparselt\_init() and rocsparselt\_destroy() functions respectively.

**rocsparselt\_mat\_descr**

The structure captures the shape and characteristics of a matrix.

It is initialized with rocsparselt\_dense\_descr\_init() or rocsparselt\_structured\_descr\_init() functions and destroyed with rocsparselt\_mat\_descr\_destroy().

**rocsparselt\_matmul\_descr**

The structure holds the description of the matrix multiplication operation.

It is initialized with rocsparselt\_matmul\_descr\_init() function.

**rocsparserlt\_matmul\_alge\_selection**

The structure holds the description of the matrix multiplication algorithm.

It is initialized with rocsparselt\_matmul\_alg\_selection\_init() function.

**rocsparselt\_matmul\_plan**

The structure holds the matrix multiplication execution plan, namely all the information necessary to execute the rocsparselt\_matmul() operation.

It is initialized and destroyed with [rocsparselt\_matmul\_plan\_init()](https://docs.nvidia.com/cuda/cusparselt/functions.html#cusparseltmatmulplaninit-label) and [rocsparselt\_matmul\_plan\_destroy()](https://docs.nvidia.com/cuda/cusparselt/functions.html#cusparseltmatmulplandestroy-label) functions respectively.

**Enumerators**

**rocsparselt\_sparsity**

The enumerator specifies the sparsity ratio of the structured matrix as

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_sparsity\_50\_percent | 50% Sparsity Ratio:  **-** **2:4** for half, bfloat16, int |

The sparsity property is used in the rocsparselt\_structured\_descr\_init() function.

**rocsparselt\_compute\_type**

The enumerator specifies the compute precision modes of the matrix

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_compute\_f32 | **-** Default mode for 32-bit floating-point precision  **-** All computations and intermediate storage ensure at least 32-bit precision  **-** Matrix Core will be used whenever possible |
| rocsparselt\_compute\_i32 | **-** Default mode for 32-bit integer precision  **-** All computations and intermediate storage ensure at least 32-bit integer precision  **-** Matrix Core will be used whenever possible |

The compute precision is used in the rocsparselt\_matmul\_descr\_init() function.

**rocsparselt\_mat\_descr\_attribute**

The enumerator specifies the additional attributes of a matrix descriptor

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_mat\_num\_batches | Number of matrices in a batch (int data type) |
| rocsparselt\_mat\_batch\_stride | Stride between consecutive matrices in a batch expressed in terms of matrix elements (int64\_t data type) |

The algorithm enumerator is used in the rocsparselt\_mat\_descr\_set\_attribute() and rocsparselt\_mat\_descr\_get\_attribute() functions.

**rocsparselt\_matmul\_descr\_attribute**

The enumerator specifies the additional attributes of a matrix multiplication descriptor

| **Value** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| rocsparselt\_matmul\_activation\_relu | int 0: **false**, **true** otherwise | false | ReLU activation function |
| rocsparselt\_matmul\_activation\_relu\_upperbound | float | inf | Upper bound of the ReLU activation function |
| rocsparselt\_matmul\_activation\_relu\_threshold | float | 0.0f | Lower threshold of the ReLU activation function |
| rocsparselt\_matmul\_activation\_gelu | int 0: **false**, **true** otherwise | false | GeLU activation function |
| rocsparselt\_matmul\_activation\_abs | int 0: **false**, **true** otherwise | false | ABS activation function |
| rocsparselt\_matmul\_activation\_leakyrelu | int 0: **false**, **true** otherwise | false | LeakyReLU activation function |
| rocsparselt\_matmul\_activation\_leakyrelu\_alpha | float | 1.0f | Alpha value of the LeakyReLU activation function |
| rocsparselt\_matmul\_activation\_sigmoid | int 0: **false**, **true** otherwise | false | Sigmoid activation function |
| rocsparselt\_matmul\_activation\_tanh | int 0: **false**, **true** otherwise | false | Tanh activation function |
| rocsparselt\_matmul\_activation\_tanh\_alpha | float | 1.0f | Alpha value of the Tanh activation function |
| rocsparselt\_matmul\_activation\_tanh\_beta | float | 1.0f | Beta value of the Tanh activation function |

where the *ReLU* activation function is defined as:

ReLU(v) =

The algorithm enumerator is used in the rocsparselt\_matmul\_descr\_set\_attribute() and rocsparselt\_matmul\_descr\_get\_attribute() functions.

**rocsparselt\_matmul\_alg**

The enumerator specifies the algorithm for matrix-matrix multiplication

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_matmul\_alg\_default | Default algorithm |

The algorithm enumerator is used in the rocsparselt\_matmul\_alg\_selection\_init() function.

**rocsparselt\_matmul\_alg\_attribute**

The enumerator specifies the matrix multiplication algorithm attributes

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_matmul\_alg\_config\_id | Algorithm ID (set and query) |
| rocsparselt\_matmul\_alg\_config\_max\_id | Algorithm ID limit (query only) |
| rocsparselt\_matmul\_search\_iterations | Number of iterations (kernel launches per algorithm) for rocsparselt\_matmul\_search(), default=10 |

The algorithm attribute enumerator is used in the rocsparselt\_matmul\_alg\_get\_attribute() and rocsparselt\_matmul\_alg\_set\_attribute() functions.

**rocsparselt\_prune\_alg**

The enumerator specifies the pruning algorithm to apply to the structured matrix before the compression

| **Value** | **Description** |
| --- | --- |
| rocsparselt\_prune\_smfmac\_tile | **-** half, bfloat16, int8, float8, bfloat8: Zero-out eight values in a 4x4 tile to maximize the *L1-norm* of the resulting tile, under the constraint of selecting exactly two elements for each row and column |
| rocsparselt\_prune\_smfmac\_strip | **-** half, bfloat16, int8, float8, bfloat8: Zero-out two values in a 1x4 strip to maximize the *L1-norm* of the resulting strip    The strip direction is chosen according to the operation op and matrix layout applied to the structured (sparse) matrix |
|  |  |

The pruning algorithm is used in the rocsparselt\_smfmac\_prune() function.

# rocsparselt Functions

## Library Management Functions

### rocsparselt\_init

rocsparselt\_status

rocsparselt\_init(rocsparselt\_handle\* handle)

The function initializes the rocsparselt library handle (rocsparselt\_handle) which holds the rocsparselt library context. It allocates light hardware resources on the host, and must be called prior to making any other rocsparselt library calls. Calling any rocsparselt function which uses rocsparselt\_handle without a previous call of rocsparselt\_init() will return an error.

The rocsparselt library context is tied to the current ROCm device. To use the library on multiple devices, one rocsparselt handle should be created for each device.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | OUT | rocsparselt library handle |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_destroy

rocsparselt\_status

rocsparselt\_destroy(**const** rocsparselt\_handle\* handle)

The function releases hardware resources used by the rocsparselt library. This function is the last call with a particular handle to the rocsparselt library.

Calling any rocsparselt function which uses rocsparselt\_handle after rocsparselt\_destroy() will return an error.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |

See rocsparselt\_status for the description of the return status.

## Matrix Descriptor Functions

### rocsparselt\_dense\_descr\_init

rocsparselt\_status

rocsparselt\_dense\_descr\_init(**const** rocsparselt\_handle\* handle,

rocsparselt\_mat\_descr\* matDescr,

int64\_t rows,

int64\_t cols,

int64\_t ld,

uint32\_t alignment,

rocsparselt\_datatype valueType,

rocsparselt\_order order);

The function initializes the descriptor of a dense matrix.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | OUT | Dense matrix description |  |
| rows | Host | IN | Number of rows |  |
| cols | Host | IN | Number of columns |  |
| ld | Host | IN | Leading dimension | ≥ rows if column-major, ≥ cols if row-major |
| alignment | Host | IN | Memory alignment in bytes | Multiple of 16 (not-used) |
| [valueType](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-data-types) | Host | IN | Data type of the matrix | rocsparselt\_datatype\_f16\_r, rocsparselt\_datatype\_bf16\_r, rocsparselt\_datatype\_i8\_r, rocsparselt\_datatype\_f8\_r, rocsparselt\_datatype\_bf8\_r |
| [order](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-layout) | Host | IN | Memory layout | rocsparselt\_order\_column,  rocsparselt\_order\_row (not supported) |

**Constrains**:

* row, col must ≥ 8
* For matrix B = K x N, K must be a multiple of 8

See rocsparselt\_status for the description of the return status.

### rocsparselt\_structured\_descr\_init

rocsparselt\_status

rocsparselt\_structured\_descr\_init(**const** rocsparselt\_handle\* handle,

rocsparselt\_mat\_descr\* matDescr,

int64\_t rows,

int64\_t cols,

int64\_t ld,

uint32\_t alignment,

rocsparselt\_datatype valueType,

rocsparselt\_order order,

rocsparselt\_sparsity\_ sparsity)

The function initializes the descriptor of a structured matrix.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | OUT | Dense matrix description |  |
| rows | Host | IN | Number of rows |  |
| cols | Host | IN | Number of columns |  |
| ld | Host | IN | Leading dimension | ≥ rows if column-major, ≥ cols if row-major |
| alignment | Host | IN | Memory alignment in bytes | Multiple of 16 (not used) |
| [valueType](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-data-types) | Host | IN | Data type of the matrix | rocsparselt\_datatype\_f16\_r, rocsparselt\_datatype\_bf16\_r, rocsparselt\_datatype\_i8\_r, rocsparselt\_datatype\_f8\_r, rocsparselt\_datatype\_bf8\_r |
| [order](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-layout) | Host | IN | Memory layout | rocsparselt\_order\_column,  rocsparselt\_order\_row (not supported) |
| [sparsity](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltsparsity-t) | Host | IN | Matrix sparsity ratio | rocsparselt\_sparsity\_50\_percent |

**Constrains**:

* row, col must ≥ 8
* For op = rocsparselt\_operation\_non
  + col must be the multiplication of 8
* For op = rocsparselt\_operation\_transpose
  + row must be the multiplication of 8

See rocsparselt\_status for the description of the return status.

### rocsparselt\_mat\_descr\_destroy

rocsparselt\_status

rocsparselt\_mat\_descr\_destroy(**const** rocsparselt\_mat\_descr\* matDescr)

The function releases the resources used by an instance of a matrix descriptor. After this call, the matrix descriptor and the matmul descriptor can no longer be used.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [matDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | IN | Matrix descriptor |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_mat\_descr\_set\_attribute

rocsparselt\_status

rocsparselt\_mat\_descr\_set\_attribute(**const** rocsparselt\_handle\* handle,

rocsparselt\_mat\_descr\* matDescr,

rocsparselt\_mat\_descr\_attribute\* matAttribute,

**const** void\* data,

size\_t dataSize)

The function sets the value of the specified attribute belonging to matrix descriptor such as number of batches and their stride.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | OUT | Matrix descriptor |  |
| [matAttribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescattribute-t) | Host | IN | Attribute to set | rocsparselt\_mat\_num\_batches, rocsparselt\_mat\_batch\_stride |
| data | Host | IN | Pointer to the value to which the specified attribute will be set |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_mat\_descr\_get\_attribute

rocsparselt\_status

rocsparselt\_mat\_descr\_get\_attribute(**const** rocsparselt\_handle handle,

**const** rocsparselt\_mat\_descr matmulDescr,

rocsparselt\_mat\_descr\_attribute matAttribute,

void\* data,

size\_t dataSize)

The function gets the value of the specified attribute belonging to matrix descriptor such as number of batches and their stride.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix descriptor |  |
| [matAttribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescattribute-t) | Host | IN | Attribute to retrieve | rocsparselt\_mat\_num\_batches, rocsparselt\_mat\_batch\_stride |
| data | Host | OUT | Memory address containing the attribute value retrieved by this function |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See rocsparselt\_status for the description of the return status.

## Matmul Descriptor Functions

### rocsparselt\_matmul\_descr\_init

rocsparselt\_status

rocsparselt\_matmul\_descr\_init(**const** rocsparselt\_handle\* handle,

rocsparselt\_matmul\_descr\* matmulDescr,

rocsparselt\_operation opA,

rocsparselt\_operation opB,

**const** rocsparselt\_mat\_descr\* matA,

**const** rocsparselt\_mat\_descr\* matB,

**const** rocsparselt\_mat\_descr\* matC,

**const** rocsparselt\_mat\_descr\* matD,

rocsparselt\_compute\_type computeType)

The function initializes the matrix multiplication descriptor.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | OUT | Matrix multiplication descriptor |  |
| [opA](https://docs.nvidia.com/cuda/cusparse/index.html#cusparseOperation_t) | Host | IN | Operation applied to the matrix A | rocsparselt\_operation\_non, rocsparselt\_operation\_transpose |
| [opB](https://docs.nvidia.com/cuda/cusparse/index.html#cusparseOperation_t) | Host | IN | Operation applied to the matrix B | rocsparselt\_operation\_non, rocsparselt\_operation\_transpose |
| [matA](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | IN | Structured matrix descriptor A |  |
| [matB](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | IN | Dense matrix descriptor B |  |
| [matC](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | IN | Dense matrix descriptor C |  |
| [matD](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatdescriptor-t) | Host | IN | Dense matrix descriptor D |  |
| [computeType](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparsecomputetype) | Host | IN | Compute precision | rocsparselt\_compute\_f32, rocsparselt\_computelt\_32i |

The structured matrix descriptor can used for [matA](https://docs.nvidia.com/cuda/cusparselt/types.html" \l "cusparseltmatdescriptor-t) or [matB](https://docs.nvidia.com/cuda/cusparselt/types.html" \l "cusparseltmatdescriptor-t) but not both.

**Data types Supported:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| rocsparselt\_datatype\_f16\_r | rocsparselt\_datatype\_f16\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_f16\_r | rocsparselt\_datatype\_f16\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_i8\_r | rocsparselt\_datatype\_i8\_r | rocsparselt\_compute\_i32 |
| rocsparselt\_datatype\_f8\_r | rocsparselt\_datatype\_f8\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_bf8\_r | rocsparselt\_datatype\_bf8\_r | rocsparselt\_compute\_f32 |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_descr\_set\_attribute

rocsparselt\_status

rocsparselt\_matmul\_descr\_set\_attribute(**const** rocsparselt\_handle\* handle,

rocsparselt\_matmul\_descr\* matmulDescr,

rocsparselt\_matmul\_descr\_attribute matmulAttribute,

**const** void\* data,

size\_t dataSize)

The function sets the value of the specified attribute belonging to matrix descriptor such as activation function and bias.

| **Parameter** | **Memory** | **In/Out** | **Description** |  |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | OUT | Matrix descriptor |  |
| [matmulAttribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescattribute-t) | Host | IN | Attribute to set | rocsparselt\_matmul\_activation\_relu, rocsparselt\_matmul\_activation\_relu\_upperbound, rocsparselt\_matmul\_activation\_relu\_threshold, rocsparselt\_matmul\_activation\_gelu, rocsparselt\_matmul\_bias\_pointer, rocsparselt\_matmul\_bias\_stride |
| data | Host | IN | Pointer to the value to which the specified attribute will be set |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_descr\_get\_attribute

rocsparselt\_status

rocsparselt\_matmul\_descr\_get\_attribute(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_descr\* matmulDescr,

rocsparselt\_matmul\_descr\_attribute matmulAttribute,

void\* data,

size\_t dataSize)

The function gets the value of the specified attribute belonging to matrix descriptor such as activation function and bias.

| **Parameter** | **Memory** | **In/Out** | **Description** |  |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix descriptor |  |
| [matmulAttribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescattribute-t) | Host | IN | Attribute to retrieve | rocsparselt\_matmul\_activation\_relu, rocsparselt\_matmul\_activation\_relu\_upperbound, rocsparselt\_matmul\_activation\_relu\_threshold, rocsparselt\_matmul\_activation\_gelu, rocsparselt\_matmul\_bias\_pointer, rocsparselt\_matmul\_bias\_stride |
| data | Host | OUT | Memory address containing the attribute value retrieved by this function |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See rocsparselt\_status for the description of the return status.

## Matmul Algorithm Functions

### rocsparselt\_matmul\_alg\_selection\_init

rocsparselt\_status

rocsparselt\_matmul\_alg\_selection\_init(**const** rocsparselt\_handle\* handle,

rocsparselt\_matmul\_alg\_selection\* algSelection,

**const** rocsparselt\_matmul\_descr\* matmulDescr,

rocsparselt\_matmul\_alg alg)

The function initializes the algorithm selection descriptor.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [algSelection](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | OUT | Algorithm selection descriptor |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix multiplication descriptor |  |
| [alg](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalg-t) | Host | IN | Algorithm mode | rocsparselt\_matmul\_alg\_default |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_alg\_set\_attribute

rocsparselt\_status

rocsparselt\_matmul\_alg\_set\_attribute(**const** rocsparselt\_handle\* handle,

rocsparselt\_matmul\_alg\_selection\* algSelection,

rocsparselt\_matmul\_alg\_attribute attribute,

**const** void\* data,

size\_t dataSize)

The function sets the value of the specified attribute belonging to algorithm selection descriptor.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [algSelection](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | OUT | Algorithm selection descriptor |  |
| [attribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t) | Host | IN | The attribute to set | rocsparselt\_matmul\_alg\_config\_id, rocsparselt\_matmul\_alg\_config\_max\_id, rocsparselt\_matmul\_search\_iterations |
| data | Host | IN | Pointer to the value to which the specified attribute will be set |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See [rocsparselt\_status](https://docs.nvidia.com/cuda/cusparse/index.html" \l "cusparseStatus_t) for the description of the return status.

### rocsparselt\_matmul\_alg\_get\_attribute

rocsparselt\_status

rocsparselt\_matmul\_alg\_get\_attribute(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_alg\_selection\* algSelection,

rocsparselt\_matmul\_alg\_attribute attribute,

void\* data,

size\_t dataSize)

The function returns the value of the queried attribute belonging to algorithm selection descriptor.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [algSelection](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | IN | Algorithm selection descriptor |  |
| [attribute](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t) | Host | IN | The attribute that will be retrieved by this function | rocsparselt\_matmul\_alg\_config\_id, rocsparselt\_matmul\_alg\_config\_max\_id, rocsparselt\_matmul\_search\_iterations |
| data | Host | OUT | Memory address containing the attribute value retrieved by this function |  |
| dataSize | Host | IN | Size in bytes of the attribute value used for verification |  |

See [rocsparselt\_status](https://docs.nvidia.com/cuda/cusparse/index.html" \l "cusparseStatus_t) for the description of the return status.

## Matmul Functions

### rocsparselt\_matmul\_get\_workspace

rocsparselt\_status

rocsparselt\_matmul\_get\_workspace(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_alg\_selection\* algSelection,

size\_t\* workspaceSize)

The function determines the required workspace size associated to the selected algorithm.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [algSelection](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | IN | Algorithm selection descriptor |
| workspaceSize | Host | OUT | Workspace size in bytes |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_plan\_init

rocsparselt\_status

rocsparselt\_matmul\_plan\_init(**const** rocsparselt\_handle\* handle,

rocsparselt\_matmul\_plan\* plan,

**const** rocsparselt\_matmul\_descr\* matmulDescr,

**const** rocsparselt\_matmul\_alg\_selection\* algSelection,

size\_t workspaceSize)

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulplan-t) | Host | OUT | Matrix multiplication plan |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix multiplication descriptor |
| [algSelection](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | IN | Algorithm selection descriptor |
| workspaceSize | Host | IN | Workspace size in bytes |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_plan\_destroy

rocsparselt\_status

rocsparselt\_matmul\_plan\_destroy(**const** rocsparselt\_matmul\_plan\* plan)

The function releases the resources used by an instance of the matrix multiplication plan. This function is the last call with a specific plan instance.

Calling any rocsparselt function which uses rocsparselt\_matmul\_plan after rocsparselt\_matmul\_plan\_destroy() will return an error.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulplan-t) | Host | IN | Matrix multiplication plan |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul

rocsparselt\_status

rocsparselt\_matmul(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_plan\* plan,

**const** void\* alpha,

**const** void\* d\_A,

**const** void\* d\_B,

**const** void\* beta,

**const** void\* d\_C,

void\* d\_D,

void\* workspace,

hipStream\_t\* streams,

int32\_t numStreams)

The function computes the matrix multiplication of matrices A and B to produce the output matrix D, according to the following operation:

D = Activation( αop(A) \* op(B) + βC + bias)

where A, B, and C are input matrices, and α and β are input scalars.

**Note**: The function currently only supports the case where D has the same shape of C

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulplan-t) | Host | IN | Matrix multiplication plan |
| alpha | Host | IN | α scalar used for multiplication (float data type) |
| d\_A | Device | IN | Pointer to the structured matrix A |
| d\_B | Device | IN | Pointer to the dense matrix B |
| beta | Host | IN | β scalar used for multiplication (float data type) |
| d\_C | Device | OUT | Pointer to the dense matrix C |
| d\_D | Device | OUT | Pointer to the dense matrix D |
| workspace | Device | IN | Pointer to workspace |
| streams | Host | IN | Pointer to HIP stream array for the computation |
| numStreams | Host | IN | Number of HIP streams in streams |

**Data types Supported:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| rocsparselt\_datatype\_f16\_r | rocsparselt\_datatype\_f16\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_f16\_r | rocsparselt\_datatype\_f16\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_i8\_r | rocsparselt\_datatype\_i8\_r | rocsparselt\_compute\_i32 |
| rocsparselt\_datatype\_f8\_r | rocsparselt\_datatype\_f8\_r | rocsparselt\_compute\_f32 |
| rocsparselt\_datatype\_bf8\_r | rocsparselt\_datatype\_bf8\_r | rocsparselt\_compute\_f32 |

The structured matrix A  (before the compression) must respect the following constrains depending on the operation applied on it:

* For op = rocsparselt\_operation\_non
  + each row must have at least two zero values every four elements
* For op = rocsparselt\_operation\_transpose
  + each column must have at least two zero values every four elements

The correctness of the pruning result (matrix A) can be check with the function rocsparselt\_smfmac\_prune\_check().

**Properties**

* The routine requires no extra storage
* The routine supports asynchronous execution with respect to streams[0]

See rocsparselt\_status for the description of the return status.

### rocsparselt\_matmul\_search

rocsparselt\_status

rocsparselt\_matmul\_search(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_plan\* plan,

**const** void\* alpha,

**const** void\* d\_A,

**const** void\* d\_B,

**const** void\* beta,

**const** void\* d\_C,

void\* d\_D,

void\* workspace,

hipStream\_t\* streams,

int32\_t numStreams)

The function evaluates all available algorithms for the matrix multiplication and automatically updates the plan by selecting the fastest one. The functionality is intended to be used for auto-tuning purposes when the same operation is repeated multiple times over different inputs.

The function behavior is the same of rocsparselt\_matmul().

* The function is NOT asynchronous with respect to streams[0] (blocking call)
* The number of iterations for the evaluation can be set by using rocsparselt\_matmul\_alg\_set\_attribute() with [rocsparselt\_matmul\_search\_iterations](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t).
* The selected algorithm id can be retrieved by using rocsparselt\_matmul\_alg\_get\_attribute() with [rocsparselt\_matmul\_alg\_config\_id](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t).

## Helper Functions

### rocsparselt\_smfmac\_prune

rocsparselt\_status

rocsparselt\_smfmac\_prune(**const** rocsparselt\_handl\*e handle,

**const** rocsparselt\_matmul\_descr\* matmulDescr,

**const** void\* d\_in,

void\* d\_out,

rocsparselt\_prune\_alg pruneAlg,

hipStream\_t stream)

The function prunes a dense matrix d\_in according to the specified algorithm pruneAlg.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix multiplication descriptor |  |
| d\_in | Device | IN | Pointer to the dense matrix |  |
| d\_out | Device | OUT | Pointer to the pruned matrix |  |
| [pruneAlg](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltprunealg-t) | Device | IN | Pruning algorithm | rocsparselt\_prune\_smfmac\_tile, rocsparselt\_prune\_smfmac\_strip |
| stream | Host | IN | HIP stream for the computation |  |

**Properties**

* The routine requires no extra storage
* The routine supports asynchronous execution with respect to stream

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_prune2

rocsparselt\_status

rocsparselt\_smfmac\_prune2(**const** rocsparselt\_handl\*e handle,

**const** rocsparselt\_mat\_descr\* sparseMatDescr,

bool isSparse,

rocsparselt\_operation op,

**const** void\* d\_in,

void\* d\_out,

rocsparselt\_prune\_alg pruneAlg,

hipStream\_t stream)

The function prunes a dense matrix d\_in according to the specified algorithm pruneAlg.

| **Parameter** | **Memory** | **In/Out** | **Description** | **Possible Values** |
| --- | --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |  |
| [sparseMatDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | structured(sparse) matrix descriptor |  |
| [isSparse](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | specify if the structured (sparse) matrix is in the first position (matA or matB) (only support matA) |  |
| [op](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | operation that will be applied to the structured (sparse) matrix in the multiplication |  |
| d\_in | Device | IN | Pointer to the dense matrix |  |
| d\_out | Device | OUT | Pointer to the pruned matrix |  |
| [pruneAlg](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltprunealg-t) | Device | IN | Pruning algorithm | rocsparselt\_prune\_smfmac\_tile, rocsparselt\_prune\_smfmac\_strip |
| stream | Host | IN | HIP stream for the computation |  |

**Properties**

* The routine requires no extra storage
* The routine supports asynchronous execution with respect to stream

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_prune\_check

rocsparselt\_status

rocsparselt\_smfmac\_prune\_check(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_descr\* matmulDescr,

**const** void\* d\_in,

int\* d\_valid,

hipStream\_t stream)

The function checks the correctness of the pruning structure for a given matrix.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [matmulDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | Matrix multiplication descriptor |
| d\_in | Device | IN | Pointer to the matrix to check |
| d\_valid | Device | OUT | Validation results (0 correct, 1 wrong) |
| stream | Host | IN | HIP stream for the computation |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_prune\_check2

rocsparselt\_status

rocsparselt\_smfmac\_prune\_check2(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_mat\_descr\* sparseMatDescr,

bool isSparse,

rocsparselt\_operation op,

**const** void\* d\_in,

int\* d\_valid,

hipStream\_t stream)

The function checks the correctness of the pruning structure for a given matrix.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [sparseMatDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | structured(sparse) matrix descriptor |
| [isSparse](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | specify if the structured (sparse) matrix is in the first position (matA or matB) (only support matA) |
| [op](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | operation that will be applied to the structured (sparse) matrix in the multiplication |
| d\_in | Device | IN | Pointer to the matrix to check |
| d\_valid | Device | OUT | Validation results (0 correct, 1 wrong) |
| stream | Host | IN | HIP stream for the computation |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_compressed\_size

rocsparselt\_status

rocsparselt\_smfmac\_compressed\_size(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_plan\* plan,

size\_t\* compressedSize)

The function provides the size of the compressed matrix to be allocated before calling rocsparselt\_smfmac\_compress().

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulplan-t) | Host | IN | Matrix plan descriptor |
| compressedSize | Host | OUT | Size in bytes of the compressed matrix |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_compressed\_size2

rocsparselt\_status

rocsparselt\_smfmac\_compressed\_size2(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_mat\_descr\* sparseMatDescr,

size\_t\* compressedSize)

The function provides the size of the compressed matrix to be allocated before calling rocsparselt\_smfmac\_compress().

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [sparseMatDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | structured(sparse) matrix descriptor |
| compressedSize | Host | OUT | Size in bytes of the compressed matrix |

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_compress

rocsparselt\_status

rocsparselt\_smfmac\_compress(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_matmul\_plan\* plan,

**const** void\* d\_dense,

void\* d\_compressed,

hipStream\_t stream)

The function compresses a dense matrix d\_dense. The compressed matrix is intended to be used as the first operand A in the rocsparselt\_matmul() function.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulplan-t) | Host | IN | Matrix multiplication plan |
| d\_dense | Device | IN | Pointer to the dense matrix |
| d\_compressed | Device | OUT | Pointer to the compressed matrix |
| stream | Host | IN | HIP stream for the computation |

**Properties**

* The routine requires no extra storage
* The routine supports asynchronous execution with respect to stream

See rocsparselt\_status for the description of the return status.

### rocsparselt\_smfmac\_compress2

rocsparselt\_status

rocsparselt\_smfmac\_compress2(**const** rocsparselt\_handle\* handle,

**const** rocsparselt\_mat\_descr\* sparseMatDescr,

bool isSparse,

rocsparselt\_operation op,

**const** void\* d\_dense,

void\* d\_compressed,

hipStream\_t stream)

The function compresses a dense matrix d\_dense. The compressed matrix is intended to be used as the first operand A in the rocsparselt\_matmul() function.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | rocsparselt library handle |
| [sparseMatDescr](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | structured(sparse) matrix descriptor |
| [isSparse](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | specify if the structured (sparse) matrix is in the first position (matA or matB) (only support matA) |
| [op](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmuldescriptor-t) | Host | IN | operation that will be applied to the structured (sparse) matrix in the multiplication |
| d\_dense | Device | IN | Pointer to the dense matrix |
| d\_compressed | Device | OUT | Pointer to the compressed matrix |
| stream | Host | IN | HIP stream for the computation |

**Properties**

* The routine requires no extra storage
* The routine supports asynchronous execution with respect to stream

See rocsparselt\_status for the description of the return status.