**hipSPARSELt Data Types**

**Data Structures**

**hipsparseLtHandle\_t**

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

The handle must be initialized and destroyed with hipsparseLtInit() and hipsparseLtDestroy() functions respectively.

**hipsparseLtMatDescriptor\_t**

The structure captures the shape and characteristics of a matrix.

It is initialized with hipsparseLtDenseDescriptorInit() or hipsparseLtStructuredDescriptorInitfunctions and destroyed with hipsparseLtMatDescriptorDestroy().

**hipsparseLtMatDescriptor\_t**

The structure holds the description of the matrix multiplication operation.

It is initialized with hipsparseLtMatmulDescriptorInit() function.

**hipsparseLtMatmulAlgSelection\_t**

The structure holds the description of the matrix multiplication algorithm.

It is initialized with hipsparseLtMatmulAlgSelectionInit() function.

**hipsparseLtMatmulPlan\_t**

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

It is initialized and destroyed with [hipsparseLtMatmulPlanInit()](https://docs.nvidia.com/cuda/cusparselt/functions.html#cusparseltmatmulplaninit-label) and [hipsparseLtMatmulPlanDestroy()](https://docs.nvidia.com/cuda/cusparselt/functions.html#cusparseltmatmulplandestroy-label) functions respectively.

**Enumerators**

**hipsparseLtSparsity\_t**

The enumerator specifies the sparsity ratio of the structured matrix as

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_SPARSITY\_50\_PERCENT | 50% Sparsity Ratio:  **-** **2:4** for half, bfloat16, int8  **-** 1**:2** for tf32, int(CUDA only) |

The sparsity property is used in the hipsparseLtStructuredDescriptorInit() function.

**hipsparseLtComputetype\_t**

The enumerator specifies the compute precision modes of the matrix

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_COMPUTE\_32F | **-** 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  (ROC only) |
| HIPSPARSELT\_COMPUTE\_32I | **-** Default mode for 32-bit integer precision  **-** All computations and intermediate storage ensure at least 32-bit integer precision  **-** Matrix Core / Tensor Core will be used whenever possible |
| HIPSPARSELT\_COMPUTE\_16F | **-** Default mode for 16-bit floating-point precision  **-** All computations and intermediate storage ensure at least 16-bit precision  **-** Matrix Core / Tensor Core will be used whenever possible |
| HIPSPARSELT\_COMPUTE\_TF32\_FAST | - Default mode for 32-bit floating-point precision  - The inputs are supposed to be directly represented in TensorFloat-32 precision. The 32-bit floating-point values are truncated to TensorFloat-32 before the computation  - All computations and intermediate storage ensure at least TensorFloat-32 precision  - Tensor Cores will be used whenever possible  (CUDA only) |
| HIPSPARSELT\_COMPUTE\_TF32 | **-** All computations and intermediate storage ensure at least TensorFloat-32 precision  **-** The inputs are rounded to TensorFloat-32 precision. This mode is slower than HIPSPARSELT\_COMPUTE\_TF32\_FAST, but could provide more accurate results  **-** Tensor Cores will be used whenever possible  (CUDA only) |

The compute precision is used in the hipsparseLtMatmulDescriptorInit() function.

**hipsparseLtMatDescAttribute\_t**

The enumerator specifies the additional attributes of a matrix descriptor

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_MAT\_NUM\_BATCHES | Number of matrices in a batch (int data type) |
| HIPSPARSELT\_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 hipsparseLtMatDescSetAttribute() and hipsparseLtMatDescGetAttribute() functions.

**hipsparseLtMatmulDescAttribute\_t**

The enumerator specifies the additional attributes of a matrix multiplication descriptor

| **Value** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU | int 0: **false**, **true** otherwise | false | ReLU activation function |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_UPPERBOUND | float | inf | Upper bound of the ReLU activation function |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_THRESHOLD | float | 0.0f | Lower threshold of the ReLU activation function |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_GELU | int 0: **false**, **true** otherwise | false | GeLU activation function |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_ABS | int 0: **false**, **true** otherwise | false | ABS activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU | int 0: **false**, **true** otherwise | false | LeakyReLU activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU\_ALPHA | float | 1.0f | Alpha value of the LeakyReLU activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_SIGMOID | int 0: **false**, **true** otherwise | false | Sigmoid activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH | int 0: **false**, **true** otherwise | false | Tanh activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_ALPHA | float | 1.0f | Alpha value of the Tanh activation function (ROC only) |
| HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_BETA | float | 1.0f | Beta value of the Tanh activation function (ROC only) |

where the *ReLU* activation function is defined as:

ReLU(v) =

The algorithm enumerator is used in the hipsparseLtMatmulDescSetAttribute() and hipsparseLtMatmulDescGetAttribute() functions.

**hipsparseLtMatmulAlg\_t**

The enumerator specifies the algorithm for matrix-matrix multiplication

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_MATMUL\_ALG\_DEFAULT | Default algorithm |

The algorithm enumerator is used in the hipsparseLtMatmulAlgSelectionInit() function.

**hipsparseLtMatmulAlgAttribute\_t**

The enumerator specifies the matrix multiplication algorithm attributes

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_ID | Algorithm ID (set and query) |
| HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_MAX\_ID | Algorithm ID limit (query only) |
| HIPSPARSELT\_MATMUL\_SEARCH\_ITERATIONS | Number of iterations (kernel launches per algorithm) for hipsparseLtMatmulSearch(), default=10 |

The algorithm attribute enumerator is used in the hipsparseLtMatmulAlgGetAttribute() and hipsparseLtMatmulAlgSetAttribute() functions.

**hipsparseLtPruneAlg\_t**

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

| **Value** | **Description** |
| --- | --- |
| HIPSPARSELT\_PRUNE\_SPMMA\_TILE | **-** half, bfloat16, int8: 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    **-** float, tf32: Zero-out two values in a 2x2 tile to maximize the L1-norm of the resulting tile, under the constraint of selecting exactly one element for each row and column (CUDA only) |
| HIPSPARSELT\_PRUNE\_SPMMA\_STRIP | **-** half, bfloat16, int8  **-** float8, bfloat8(ROC only)  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  **-** float, tf32: Zero-out one value in a 1x2 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 (CUDA only) |

The pruning algorithm is used in the hipsparseLtSpMMAPrune() function.

# hipSPARSELt Functions

## Library Management Functions

### hipsparseLtInit

hipsparseLtStatus\_t

hipsparseLtInit(hipsparseLtHandle\_t\* handle)

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

The hipsparselt library context is tied to the current ROCm/CUDA device. To use the library on multiple devices, one hipsparselt 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 | hipsparselt library handle |

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtDestroy

hipsparseLtStatus\_t

hipsparseLtDestroy(**const** hipsparseLtHandle\_t\* handle)

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

Calling any hipsparselt function which uses hipsparseLtHandle\_t after hipsparseLtDestroy() will return an error.

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

See hipsparseLtStatus\_t for the description of the return status.

## Matrix Descriptor Functions

### hipsparseLtDenseDescriptorInit

hipsparseLtStatus\_t

hipsparseLtDenseDescriptorInit(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatDescriptor\_t\* matDescr,

int64\_t rows,

int64\_t cols,

int64\_t ld,

uint32\_t alignment,

hipsparseLtDatatype\_t valueType,

hipsparseLtOrder\_t 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 | hipsparselt 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 (CUDA only) |
| [valueType](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-data-types) | Host | IN | Data type of the matrix | HIPSPARSELT\_R\_32F (CUDA only),  HIPSPARSELT\_R\_16F,  HIPSPARSELT\_R\_16BF,  HIPSPARSELT\_R\_8I,  HIPSPARSELT\_R\_8F (ROC only),  HIPSPARSELT\_R\_8BF (ROC only) |
| [order](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-layout) | Host | IN | Memory layout | HIPSPARSELT\_ORDER\_COLUMN,  HIPSPARSELT\_ORDER\_ROW (CUDA only) |

**Constrains**:

* **ROC Backend:**
  + row, col must ≥ 8
  + For matrix B = K x N, K must be a multiple of 8
* **CUDA Backend:**
* rows, cols, and ld must be a multiple of
  + 16 if valueType is HIPSPARSELT\_R\_8I
  + 8 if valueType is HIPSPARSELT\_R\_16F or HIPSPARSELT\_R\_16BF
  + 4 if valueType is HIPSPARSELT\_R\_32F
* The total size of the matrix cannot exceed:
  + 232 - 1elements for HIPSPARSELT\_R\_8I
  + 231 - 1 elements for HIPSPARSELT\_R\_16F or HIPSPARSELT\_R\_16BF
  + 230 - 1 elements for HIPSPARSELT\_R\_32F

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtStructuredDescriptorInit

hipsparseLtStatus\_t

hipsparseLtStructuredDescriptorInit(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatDescriptor\_t\* matDescr,

int64\_t rows,

int64\_t cols,

int64\_t ld,

uint32\_t alignment,

hipsparseLtDatatype\_t valueType,

hipsparseLtOrder\_t order,

hipsparseLtSparsity\_t 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 | hipsparselt 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 (CUDA only) |
| [valueType](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-data-types) | Host | IN | Data type of the matrix | HIPSPARSELT\_R\_32F (CUDA only),  HIPSPARSELT\_R\_16F,  HIPSPARSELT\_R\_16BF,  HIPSPARSELT\_R\_8I,  HIPSPARSELT\_R\_8F (ROC only),  HIPSPARSELT\_R\_8BF (ROC only) |
| [order](https://docs.nvidia.com/cuda/cusparse/index.html#cusparse-generic-enum-layout) | Host | IN | Memory layout | HIPSPARSELT\_ORDER\_COLUMN,  HIPSPARSELT\_ORDER\_ROW (CUDA only) |
| [sparsity](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltsparsity-t) | Host | IN | Matrix sparsity ratio | HIPSPARSELT\_SPARSITY\_50\_PERCENT |

**Constrains**:

* **ROC Backend:**
* row, col must ≥ 8
* For op = HIPSPARSELT\_OPERATION\_NON\_TRANSPOSE
  + col must be the multiplication of 8
* For op = HIPSPARSELT\_OPERATION\_TRANSPOSE
  + row must be the multiplication of 8
* **CUDA Backend:**
* rows, cols, and ld must be a multiple of
  + 32 if valueType is HIPSPARSELT\_R\_8I
  + 8 if valueType is HIPSPARSELT\_R\_16F or HIPSPARSELT\_R\_16BF
  + 4 if valueType is HIPSPARSELT\_R\_32F
* The total size of the matrix cannot exceed:
  + 232 - 1elements for HIPSPARSELT\_R\_8I
  + 231 - 1 elements for HIPSPARSELT\_R\_16F or HIPSPARSELT\_R\_16BF
  + 230 - 1 elements for HIPSPARSELT\_R\_32F

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatDescriptorDestroy

hipsparseLtStatus\_t

hipsparseLtMatDescriptorDestroy(**const** hipsparseLtMatDescriptor\_t\* 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatDescSetAttribute

hipsparseLtStatus\_t

hipsparseLtMatDescSetAttribute(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatDescriptor\_t\* matmulDescr,

hipsparseLtMatDescAttribute\_t 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 | hipsparselt 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 | HIPSPARSELT\_MAT\_NUM\_BATCHES,  HIPSPARSELT\_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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatDescGetAttribute

hipsparseLtStatus\_t

hipsparseLtMatDescGetAttribute(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatDescriptor\_t\* matmulDes

hipsparseLtMatDescAttribute\_t matAttrib

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 | hipsparselt 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 | HIPSPARSELT\_MAT\_NUM\_BATCHES, HIPSPARSELT\_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 hipsparseLtStatus\_t for the description of the return status.

## Matmul Descriptor Functions

### hipsparseLtMatmulDescriptorInit

hipsparseLtStatus\_t

hipsparseLtMatmulDescriptorInit(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulDescriptor\_t\* matmulDescr,

hipsparseLtOperation\_t opA,

hipsparseLtOperation\_t opB,

**const** hipsparseLtMatDescriptor\_t\* matA,

**const** hipsparseLtMatDescriptor\_t\* matB,

**const** hipsparseLtMatDescriptor\_t\* matC,

**const** hipsparseLtMatDescriptor\_t\* matD,

hipsparseLtComputetype\_t 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 | hipsparselt 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 | HIPSPARSELT\_OPERATION\_NON\_TRANSPOSE, HIPSPARSELT\_OPERATION\_TRANSPOSE |
| [opB](https://docs.nvidia.com/cuda/cusparse/index.html#cusparseOperation_t) | Host | IN | Operation applied to the matrix B | HIPSPARSELT\_OPERATION\_NON\_TRANSPOSE, HIPSPARSELT\_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 | HIPSPARSELT\_COMPUTE\_32F8F (ROC only),  HIPSPARSELT\_COMPUTE\_32I,  HIPSPARSELT\_COMPUTE\_16F (CUDA only),  HIPSPARSELT\_COMPUTE\_TF32(CUDA only),  HIPSPARSELT\_COMPUTE\_TF32\_FAST (CUDA only) |

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:**

* **ROC Backend:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| HIPSPARSELT\_R\_16F | HIPSPARSELT\_R\_16F | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_16BF | HIPSPARSELT\_R\_16BF | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_8I | HIPSPARSELT\_R\_8I | HIPSPARSELT\_COMPUTE\_32I |
| HIPSPARSELT\_R\_8F | HIPSPARSELT\_R\_8F | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_8BF | HIPSPARSELT\_R\_8BF | HIPSPARSELT\_COMPUTE\_32F |

* **CUDA Backend:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| HIPSPARSELT\_R\_16F | HIPSPARSELT\_R\_16F | HIPSPARSELT\_COMPUTE\_16F |
| HIPSPARSELT\_R\_16BF | HIPSPARSELT\_R\_16BF | HIPSPARSELT\_COMPUTE\_16F |
| HIPSPARSELT\_R\_8I | HIPSPARSELT\_R\_8I | HIPSPARSELT\_COMPUTE\_32I |
| HIPSPARSELT\_R\_32F | HIPSPARSELT\_R\_32F | HIPSPARSELT\_COMPUTE\_TF32\_FAST |
| HIPSPARSELT\_R\_32F | HIPSPARSELT\_R\_32F | HIPSPARSELT\_COMPUTE\_TF32 |

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulDescSetAttribute

hipsparseLtStatus\_t

hipsparseLtMatmulDescSetAttribute(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulDescriptor\_t\* matmulDescr,

hipsparseLtMatmulDescAttribute\_t 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 | hipsparselt 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 | HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU, HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_UPPERBOUND,  HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_THRESHOLD, HIPSPARSELT\_MATMUL\_ACTIVATION\_GELU,  HIPSPARSELT\_MATMUL\_BIAS\_POINTER, HIPSPARSELT\_MATMUL\_BIAS\_STRIDE  ROC Only:  HIPSPARSELT\_MATMUL\_ACTIVATION\_ABS, HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU, HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU\_ALPHA, HIPSPARSELT\_MATMUL\_ACTIVATION\_SIGMOID, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_ALPHA, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_BETA |
| 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulDescGetAttribute

hipsparseLtStatus\_t

hipsparseLtMatmulDescGetAttribute(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulDescriptor\_t\* matmulDescr,

hipsparseLtMatmulDescAttribute\_t 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 | hipsparselt 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 | HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU, HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_UPPERBOUND,  HIPSPARSELT\_MATMUL\_ACTIVATION\_RELU\_THRESHOLD, HIPSPARSELT\_MATMUL\_ACTIVATION\_GELU,  HIPSPARSELT\_MATMUL\_BIAS\_POINTER, HIPSPARSELT\_MATMUL\_BIAS\_STRIDE  ROC Only:  HIPSPARSELT\_MATMUL\_ACTIVATION\_ABS, HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU, HIPSPARSELT\_MATMUL\_ACTIVATION\_LEAKYRELU\_ALPHA, HIPSPARSELT\_MATMUL\_ACTIVATION\_SIGMOID, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_ALPHA, HIPSPARSELT\_MATMUL\_ACTIVATION\_TANH\_BETA |
| 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 hipsparseLtStatus\_t for the description of the return status.

## Matmul Algorithm Functions

### hipsparseLtMatmulAlgSelectionInit

hipsparseLtStatus\_t

hipsparseLtMatmulAlgSelectionInit(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulAlgSelection\_t\* algSelection,

const hipsparseLtMatmulDescriptor\_t\* matmulDescr,

hipsparseLtMatmulAlg\_t 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 | hipsparselt 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 | HIPSPARSELT\_MATMUL\_ALG\_DEFAULT |

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulAlgSetAttribute

hipsparseLtStatus\_t

hipsparseLtMatmulAlgSetAttribute(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulAlgSelection\_t\* algSelect

hipsparseLtMatmulAlgAttribute\_t 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 | hipsparselt 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 | HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_ID, HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_MAX\_ID, HIPSPARSELT\_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 [hipsparseLtStatus\_t](https://docs.nvidia.com/cuda/cusparse/index.html" \l "cusparseStatus_t) for the description of the return status.

### hipsparseLtMatmulAlgGetAttribute

hipsparseLtStatus\_t

hipsparseLtMatmulAlgGetAttribute(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulAlgSelection\_t\* algSelection,

hipsparseLtMatmulAlgAttribute\_t 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 | hipsparselt 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 | HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_ID, HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_MAX\_ID, HIPSPARSELT\_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 [hipsparseLtStatus\_t](https://docs.nvidia.com/cuda/cusparse/index.html" \l "cusparseStatus_t) for the description of the return status.

## Matmul Functions

### hipsparseLtMatmulGetWorkspace

hipsparseLtStatus\_t

hipsparseLtMatmulGetWorkspace(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulPlan\_t\* plan,

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 | hipsparselt library handle |
| [plan](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgselection-t) | Host | IN | Matrix multiplication plan |
| workspaceSize | Host | OUT | Workspace size in bytes |

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulPlanInit

hipsparseLtStatus\_t

hipsparseLtMatmulPlanInit(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulPlan\_t\* plan,

**const** hipsparseLtMatmulDescriptor\_t\* matmulDescr,

**const** hipsparseLtMatmulAlgSelection\_t\* algSelection,

size\_t workspaceSize)

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulPlanDestroy

hipsparseLtStatus\_t

hipsparseLtMatmulPlanDestroy(**const** hipsparseLtMatmulPlan\_t\* 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 hipsparselt function which uses hipsparseLtMatmulPlan\_t after hipsparseLtMatmulPlanDestroy() 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmul

hipsparseLtStatus\_t

hipsparseLtMatmul(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulPlan\_t\* 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 | hipsparselt 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:**

* **ROC Backend:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| HIPSPARSELT\_R\_16F | HIPSPARSELT\_R\_16F | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_16BF | HIPSPARSELT\_R\_16BF | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_8I | HIPSPARSELT\_R\_8I | HIPSPARSELT\_COMPUTE\_32I |
| HIPSPARSELT\_R\_8F | HIPSPARSELT\_R\_8F | HIPSPARSELT\_COMPUTE\_32F |
| HIPSPARSELT\_R\_8BF | HIPSPARSELT\_R\_8BF | HIPSPARSELT\_COMPUTE\_32F |

* **CUDA Backend:**

| **Input** | **Output** | **Compute** |
| --- | --- | --- |
| HIPSPARSELT\_R\_16F | HIPSPARSELT\_R\_16F | HIPSPARSELT\_COMPUTE\_16F |
| HIPSPARSELT\_R\_16BF | HIPSPARSELT\_R\_16BF | HIPSPARSELT\_COMPUTE\_16F |
| HIPSPARSELT\_R\_8I | HIPSPARSELT\_R\_8I | HIPSPARSELT\_COMPUTE\_32I |
| HIPSPARSELT\_R\_32F | HIPSPARSELT\_R\_32F | HIPSPARSELT\_COMPUTE\_TF32\_FAST |
| HIPSPARSELT\_R\_32F | HIPSPARSELT\_R\_32F | HIPSPARSELT\_COMPUTE\_TF32 |

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

* For op = HIPSPARSELT\_OPERATION\_NON\_TRANSPOSE
  + HIPSPARSELT\_R\_16F, HIPSPARSELT\_R\_16BF, HIPSPARSELT\_R\_8I, HIPSPARSELT\_R\_8F, HIPSPARSELT\_R\_8BF each row must have at least two zero values every four elements
  + HIPSPARSELT\_R\_32F each row must have at least one zero values every two elements
* For op = HIPSPARSELT\_OPERATION\_TRANSPOSE
  + HIPSPARSELT\_R\_16F, HIPSPARSELT\_R\_16BF, HIPSPARSELT\_R\_8I, HIPSPARSELT\_R\_8F, HIPSPARSELT\_R\_8BF each column must have at least two zero values every four elements
    - HIPSPARSELT\_R\_32F each column must have at least one zero values every two elements

The correctness of the pruning result (matrix A) can be check with the function hipsparseLtSpMMAPruneCheck().

**Properties**

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

See hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtMatmulSearch

hipsparseLtStatus\_t

hipsparseLtMatmulSearch(**const** hipsparseLtHandle\_t\* handle,

hipsparseLtMatmulPlan\_t\* 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 hipsparseLtMatmull().

* The function is NOT asynchronous with respect to streams[0] (blocking call)
* The number of iterations for the evaluation can be set by using hipsparseLtMatmulAlgSetAttribute() with [HIPSPARSELT\_MATMUL\_SEARCH\_ITERATIONS](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t).
* The selected algorithm id can be retrieved by using hipsparseLtMatmulAlgGetAttribute() with [HIPSPARSELT\_MATMUL\_ALG\_CONFIG\_ID](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparseltmatmulalgattribute-t).

## Helper Functions

### hipsparseLtSpMMAPrune

hipsparseLtStatus\_t

hipsparseLtSpMMAPrune(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulDescriptor\_t\* matmulDe

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

void\* d\_out,

hipsparseLtPruneAlg\_t 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 | hipsparselt 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 | HIPSPARSELT\_PRUNE\_SPMMA\_TILE, HIPSPARSELT\_PRUNE\_SPMMA\_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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMAPrune2

hipsparseLtStatus\_t

hipsparseLtSpMMAPrune2(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatDescriptor\_t\* sparseMat

int isSparseA

hipsparseLtOperation\_t op,

const void\* d\_in,

void\* d\_out,

hipsparseLtPruneAlg\_t 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 | hipsparselt 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 | hipsparselt\_prune\_smfmac\_tile, hipsparselt\_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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMAPruneCheck

hipsparseLtStatus\_t

hipsparseLtSpMMAPruneCheck(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulDescriptor\_t\* matmulD

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

int\* 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 | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMAPruneCheck2

hipsparseLtStatus\_t

hipsparseLtSpMMAPruneCheck2(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatDescriptor\_t\* sparseMa

int isSparse

hipsparseLtOperation\_t 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 | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMACompressedSize

hipsparseLtStatus\_t

hipsparseLtSpMMACompressedSize(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulPlan\_t\* plan,

size\_t\* compressedSize)

The function provides the size of the compressed matrix to be allocated before calling hipsparseLtSpMMACompress().

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMACompressedSize2

hipsparseLtStatus\_t

hipsparseLtSpMMACompressedSize2(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatDescriptor\_t\* sparseMatDescr,

size\_t\* compressedSize)

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

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMACompress

hipsparseLtStatus\_t

hipsparseLtSpMMACompress(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulPlan\_t\* plan,

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

void\* d\_compressed,

hipsparseLtSpMMACompress(**const** hipsparseLtHandle\_t\* handle,

**const** hipsparseLtMatmulPlan\_t\* plan,

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 hipsparseLtMatmull() function.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.

### hipsparseLtSpMMACompress2

hipsparseLtStatus\_t

hipsparseLtSpMMACompress2(const hipsparseLtHandle\_t\* handle,

const hipsparseLtMatDescriptor\_t\* sparseMatDescr

int isSparseA,

hipsparseLtOperation\_t 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 hipsparselt\_matmul() function.

| **Parameter** | **Memory** | **In/Out** | **Description** |
| --- | --- | --- | --- |
| [handle](https://docs.nvidia.com/cuda/cusparselt/types.html#cusparselthandle-t) | Host | IN | hipsparselt 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 hipsparseLtStatus\_t for the description of the return status.