

# Rocblas Deverlopers Guide

Xiangmin Jiao

December 11, 2003

## Contents

<a href="#">1 Rocblas Compound Index</a>	1
<a href="#">2 Rocblas File Index</a>	1
<a href="#">3 Rocblas Class Documentation</a>	2
<a href="#">4 Rocblas File Documentation</a>	11

## 1 Rocblas Compound Index

### 1.1 Rocblas Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Rocblas</a>	2
<a href="#">Rocblas::assn&lt; T &gt;</a>	10
<a href="#">Rocblas::limit1v&lt; T &gt;</a>	??
<a href="#">Rocblas::maxv&lt; T &gt;</a>	??
<a href="#">Rocblas::minv&lt; T &gt;</a>	??
<a href="#">Rocblas::nega&lt; T &gt;</a>	11
<a href="#">Rocblas::sumv&lt; T &gt;</a>	??
<a href="#">Rocblas::swapp&lt; T &gt;</a>	11

## 2 Rocblas File Index

### 2.1 Rocblas File List

Here is a list of all files with brief descriptions:

<a href="#">Rocblas.C</a>	11
<a href="#">Rocblas.h</a>	13

## 3 Rocblas Class Documentation

### 3.1 Rocblas Class Reference

```
#include <Rocblas.h>
```

#### Static Public Member Functions

- void **init** (const std::string &name)  
*Creates window for Rocblas and registers functions.*
- void **finalize** (const std::string &name)  
*Delete window for Rocblas.*
- void **add** (const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for addition.*
- void **sub** (const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for subtraction.*
- void **mul** (const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for multiplication.*
- void **limit1** (const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for limit1.*
- void **div** (const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for division.*
- void **add\_scalar** (const Attribute \*x, const void \*y, Attribute \*z, int swap=0)  
*Operation wrapper for addition with y as a scalar pointer.*
- void **sub\_scalar** (const Attribute \*x, const void \*y, Attribute \*z, int swap=0)  
*Operation wrapper for subtraction with y as a scalar pointer.*
- void **mul\_scalar** (const Attribute \*x, const void \*y, Attribute \*z, int swap=0)  
*Operation wrapper for multiplication with y as a scalar pointer.*
- void **div\_scalar** (const Attribute \*x, const void \*y, Attribute \*z, int swap=0)  
*Operation wrapper for division with y as a scalar pointer.*

- void `dot` (const Attribute \*x, const Attribute \*y, Attribute \*z, const Attribute \*mults=NULL)  
*Wrapper for dot product.*
- void `dot_scalar` (const Attribute \*x, const Attribute \*y, void \*z, const Attribute \*mults=NULL)  
*Wrapper for 2-norm with z as a scalar pointer.*
- void `dot_MPI` (const Attribute \*x, const Attribute \*y, Attribute \*z, const int \*comm, const Attribute \*mults=NULL)  
*Wrapper for dot product.*
- void `dot_scalar_MPI` (const Attribute \*x, const Attribute \*y, void \*z, const int \*comm, const Attribute \*mults=NULL)  
*Wrapper for 2-norm with z as a scalar pointer.*
- void `nrm2` (const Attribute \*x, Attribute \*y, const Attribute \*mults=NULL)  
*Wrapper for 2-norm.*
- void `nrm2_scalar` (const Attribute \*x, void \*y, const Attribute \*mults=NULL)  
*Wrapper for 2-norm with y as a scalar pointer.*
- void `nrm2_MPI` (const Attribute \*x, Attribute \*y, const int \*comm, const Attribute \*mults=NULL)  
*Wrapper for 2-norm with MPI.*
- void `nrm2_scalar_MPI` (const Attribute \*x, void \*y, const int \*comm, const Attribute \*mults=NULL)  
*Wrapper for 2-norm with y as a scalar pointer with MPI.*
- void `swap` (Attribute \*x, Attribute \*y)  
*Wrapper for swap.*
- void `copy` (const Attribute \*x, Attribute \*y)  
*Wrapper for copy.*
- void `copy_scalar` (const void \*x, Attribute \*y)  
*Operation wrapper for copy (x is a scalar pointer).*
- void `neg` (const Attribute \*x, Attribute \*y)  
*Wrapper for neg (y=-x).*
- void `max_MPI` (const Attribute \*x, Attribute \*y, int \*comm)

*Wrapper for max.*

- void [max\\_scalar\\_MPI](#) (const Attribute \*x, void \*y, int \*comm)  
*Operation wrapper for max (y is a scalar pointer).*
- void [min\\_MPI](#) (const Attribute \*x, Attribute \*y, int \*comm)  
*Wrapper for min.*
- void [min\\_scalar\\_MPI](#) (const Attribute \*x, void \*y, int \*comm)  
*Operation wrapper for min (y is a scalar pointer).*
- void [sum\\_MPI](#) (const Attribute \*x, Attribute \*y, int \*comm)  
*Wrapper for sum.*
- void [sum\\_scalar\\_MPI](#) (const Attribute \*x, void \*y, int \*comm)  
*Operation wrapper for sum (y is a scalar pointer).*
- void [axpy](#) (const Attribute \*a, const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for  $z = a * x + y$ .*
- void [axpy\\_scalar](#) (const void \*a, const Attribute \*x, const Attribute \*y, Attribute \*z)  
*Operation wrapper for  $z = a * x + y$  (a is a scalar pointer).*

### Protected Types

- enum {  
    [BLAS\\_VOID](#), [BLAS\\_SCALAR](#), [BLAS\\_VEC](#), [BLAS\\_SCNE](#),  
    [BLAS\\_VECNE](#) }

### Static Protected Member Functions

- template<class FuncType, int ytype> void [calc](#) (Attribute \*z, const Attribute \*x, const void \*yin, FuncType opp, bool swap=false)  
*Performs the operation:  $z = x \text{ op } y$ .*
- template<class data\_type, int ztype> void [calcDot](#) (void \*zout, const Attribute \*x, const Attribute \*y, const int \*comm=NULL, const Attribute \*mults=NULL)  
*Performs the operation:  $z = \langle x, y \rangle$ .*

- `template<class FuncType, int ytype> void gen2arg (Attribute *z, void *yin, FuncType opp)`  
*Performs the operation  $opp(x, y)$ .*
- `template<class data_type, int atype> void axpy\_gen (const void *a, const Attribute *x, const Attribute *y, Attribute *z)`  
*Performs the operation:  $z = a*x + y$ .*
- `template<class FuncType> void calcChoose (const Attribute *x, const Attribute *y, Attribute *z, FuncType opp)`  
*Chooses which calc function to call based on type of y.*
- `template<int attr_type> int get\_stride (const Attribute *attr)`
- `template<class data_type, int attr_type, bool is_staggered> data_type & getref (data_type *base, const int r, const int c, const int nc)`
- `template<class data_type, int attr_type, bool is_staggered> const data_type & getref (const data_type *base, const int r, const int c, const int nc)`
- `template<class OPint, class OPdbl, int OPMPI> void reduce\_MPI (const Attribute *x, Attribute *z, int *comm, int, double)`
- `template<class OPint, class OPdbl, int OPMPI> void reduce\_scalar\_MPI (const Attribute *x, void *y, int *comm, int, double)`

### 3.1.1 Member Enumeration Documentation

#### 3.1.1.1 anonymous enum [protected]

Enumeration values:

**BLAS\_VOID**  
**BLAS\_SCALAR**  
**BLAS\_VEC**  
**BLAS\_SCNE**  
**BLAS\_VECNE**

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `void Rocblas::add (const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for addition.

**3.1.2.2** `void Rocblas::add_scalar (const Attribute * x, const void * y, Attribute * z, int swap = 0)` [static]

Operation wrapper for addition with y as a scalar pointer.

**3.1.2.3** `void Rocblas::axy (const Attribute * a, const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for  $z = a * x + y$ .

**3.1.2.4** `template<class data_type, int atype> void Rocblas::axy_gen (const void * a, const Attribute * x, const Attribute * y, Attribute * z)` [static, protected]

Performs the operation:  $z = a * x + y$ .

**3.1.2.5** `void Rocblas::axy_scalar (const void * a, const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for  $z = a * x + y$  (a is a scalar pointer).

**3.1.2.6** `template<class FuncType, int ytype> void Rocblas::calc (Attribute * z, const Attribute * x, const void * yin, FuncType opp, bool swap = false)` [static, protected]

Performs the operation:  $z = x \text{ op } y$ .

**3.1.2.7** `template<class FuncType> void Rocblas::calcChoose (const Attribute * x, const Attribute * y, Attribute * z, FuncType opp)` [static, protected]

Chooses which calc function to call based on type of y.

**3.1.2.8** `template<class data_type, int ytype> void Rocblas::calcDot (void * zout, const Attribute * x, const Attribute * y, const int * comm = NULL, const Attribute * mults = NULL)` [static, protected]

Performs the operation:  $z = \langle x, y \rangle$ .

**3.1.2.9** `void Rocblas::copy (const Attribute * x, Attribute * y)` [static]

Wrapper for copy.

**3.1.2.10** `void Rocblas::copy_scalar (const void * x, Attribute * y)` [static]

Operation wrapper for copy (x is a scalar pointer).

**3.1.2.11** `void Rocblas::div (const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for division.

**3.1.2.12** `void Rocblas::div_scalar (const Attribute * x, const void * y, Attribute * z, int swap = 0)` [static]

Operation wrapper for division with y as a scalar pointer.

**3.1.2.13** `void Rocblas::dot (const Attribute * x, const Attribute * y, Attribute * z, const Attribute * mults = NULL)` [static]

Wrapper for dot product.

**3.1.2.14** `void Rocblas::dot_MPI (const Attribute * x, const Attribute * y, Attribute * z, const int * comm, const Attribute * mults = NULL)` [static]

Wrapper for dot product.

**3.1.2.15** `void Rocblas::dot_scalar (const Attribute * x, const Attribute * y, void * z, const Attribute * mults = NULL)` [static]

Wrapper for 2-norm with z as a scalar pointer.

**3.1.2.16** `void Rocblas::dot_scalar_MPI (const Attribute * x, const Attribute * y, void * z, const int * comm, const Attribute * mults = NULL)` [static]

Wrapper for 2-norm with z as a scalar pointer.

**3.1.2.17** `void Rocblas::finalize (const std::string & name)` [static]

Delete window for Rocblas.

**3.1.2.18** `template<class FuncType, int ytype> void Rocblas::gen2arg (Attribute * z, void * yin, FuncType opp)` [static, protected]

Performs the operation opp(x, y).

**3.1.2.19** `template<int attr_type> int Rocblas::get_stride (const Attribute * attr)` [inline, static, protected]



**3.1.2.20** `template<class data_type, int attr_type, bool is_staggered> const data_type & Rocblas::getref (const data_type * base, const int r, const int c, const int nc)` [inline, static, protected]

**3.1.2.21** `template<class data_type, int attr_type, bool is_staggered> data_type & Rocblas::getref (data_type * base, const int r, const int c, const int nc)` [inline, static, protected]

**3.1.2.22** `void Rocblas::init (const std::string & name)` [static]

Creates window for Rocblas and registers functions.

**3.1.2.23** `void Rocblas::limit1 (const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for limit1.

**3.1.2.24** `void Rocblas::max_MPI (const Attribute * x, Attribute * y, int * comm)` [static]

Wrapper for max.

**3.1.2.25** `void Rocblas::max_scalar_MPI (const Attribute * x, void * y, int * comm)` [static]

Operation wrapper for max (y is a scalar pointer).

**3.1.2.26** `void Rocblas::min_MPI (const Attribute * x, Attribute * y, int * comm)` [static]

Wrapper for min.

**3.1.2.27** `void Rocblas::min_scalar_MPI (const Attribute * x, void * y, int * comm)` [static]

Operation wrapper for min (y is a scalar pointer).

**3.1.2.28** `void Rocblas::mul (const Attribute * x, const Attribute * y, Attribute * z)` [static]

Operation wrapper for multiplication.

**3.1.2.29** void Rocblas::mul\_scalar (const Attribute \* x, const void \* y, Attribute \* z, int swap = 0) [static]

Operation wrapper for multiplication with y as a scalar pointer.

**3.1.2.30** void Rocblas::neg (const Attribute \* x, Attribute \* y) [static]

Wrapper for neg (y=-x).

**3.1.2.31** void Rocblas::nrm2 (const Attribute \* x, Attribute \* y, const Attribute \* mults = NULL) [static]

Wrapper for 2-norm.

**3.1.2.32** void Rocblas::nrm2\_MPI (const Attribute \* x, Attribute \* y, const int \* comm, const Attribute \* mults = NULL) [static]

Wrapper for 2-norm with MPI.

**3.1.2.33** void Rocblas::nrm2\_scalar (const Attribute \* x, void \* y, const Attribute \* mults = NULL) [static]

Wrapper for 2-norm with y as a scalar pointer.

**3.1.2.34** void Rocblas::nrm2\_scalar\_MPI (const Attribute \* x, void \* y, const int \* comm, const Attribute \* mults = NULL) [static]

Wrapper for 2-norm with y as a scalar pointer with MPI.

**3.1.2.35** template<class OPint, class OPdbl, int OPMPI> void Rocblas::reduce\_MPI (const Attribute \* x, Attribute \* z, int \* comm, int, double) [inline, static, protected]

**3.1.2.36** template<class OPint, class OPdbl, int OPMPI> void Rocblas::reduce\_scalar\_MPI (const Attribute \* x, void \* y, int \* comm, int, double) [inline, static, protected]

**3.1.2.37** void Rocblas::sub (const Attribute \* x, const Attribute \* y, Attribute \* z) [static]

Operation wrapper for subtraction.

**3.1.2.38** void Rocblas::sub\_scalar (const Attribute \* x, const void \* y, Attribute \* z, int swap = 0) [static]

Operation wrapper for subtraction with y as a scalar pointer.

**3.1.2.39** void Rocblas::sum\_MPI (const Attribute \* x, Attribute \* y, int \* comm) [static]

Wrapper for sum.

**3.1.2.40** void Rocblas::sum\_scalar\_MPI (const Attribute \* x, void \* y, int \* comm) [static]

Operation wrapper for sum (y is a scalar pointer).

**3.1.2.41** void Rocblas::swap (Attribute \* x, Attribute \* y) [static]

Wrapper for swap.

The documentation for this class was generated from the following files:

- [Rocblas.h](#)
- [Rocblas.C](#)

## 3.2 Rocblas::asn< T > Struct Template Reference

### Public Member Functions

- void [operator\(\)](#) (T &x, const T &y)

template<class T> struct Rocblas::asn< T >

### 3.2.1 Member Function Documentation

**3.2.1.1** template<class T> void Rocblas::asn< T >::operator() (T &x, const T &y) [inline]

The documentation for this struct was generated from the following file:

- [Rocblas.C](#)

### 3.3 Rocblas::nega< T > Struct Template Reference

#### Public Member Functions

- void [operator\(\)](#) (T &x, const T &y)

```
template<class T> struct Rocblas::nega< T >
```

#### 3.3.1 Member Function Documentation

**3.3.1.1** `template<class T> void Rocblas::nega< T >::operator\(\) (T & x, const T & y) [inline]`

The documentation for this struct was generated from the following file:

- [Rocblas.C](#)

### 3.4 Rocblas::swapp< T > Struct Template Reference

#### Public Member Functions

- void [operator\(\)](#) (T &x, T &y)

```
template<class T> struct Rocblas::swapp< T >
```

#### 3.4.1 Member Function Documentation

**3.4.1.1** `template<class T> void Rocblas::swapp< T >::operator\(\) (T & x, T & y) [inline]`

The documentation for this struct was generated from the following file:

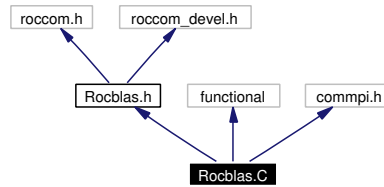
- [Rocblas.C](#)

## 4 Rocblas File Documentation

### 4.1 Rocblas.C File Reference

```
#include "Rocblas.h"  
#include <functional>  
#include "commpi.h"
```

Include dependency graph for Rocblas.C:



## Compounds

- struct [Rocblas::assn](#)
- struct [Rocblas::limit1v](#)
- struct [Rocblas::maxv](#)
- struct [Rocblas::minv](#)
- struct [Rocblas::nega](#)
- struct [Rocblas::sumv](#)
- struct [Rocblas::swapp](#)

## Defines

- #define [ROCBLAS\\_LOAD\\_MODULE](#) COM\_F\_FUNC( rocblas\_load\_module)
- #define [ROCBLAS\\_UNLOAD\\_MODULE](#) COM\_F\_FUNC( rocblas\_unload\_module)

## Functions

- void [Rocblas\\_load\\_module](#) (const char \*name)  
*Calls Rocblas initialization function.*
- void [Rocblas\\_unload\\_module](#) (const char \*name)
- void [ROCBLAS\\_LOAD\\_MODULE](#) (const char \*name, int length)
- void [ROCBLAS\\_UNLOAD\\_MODULE](#) (const char \*name, int length)

### 4.1.1 Detailed Description

Implementation of [Rocblas](#).

### 4.1.2 Define Documentation

**4.1.2.1 #define ROCBLAS\_LOAD\_MODULE COM\_F\_FUNC( rocblas\_load\_module)**

**4.1.2.2 #define ROCBLAS\_UNLOAD\_MODULE COM\_F\_FUNC( rocblas\_unload\_module)**

### 4.1.3 Function Documentation

**4.1.3.1 void ROCBLAS\_LOAD\_MODULE (const char \* name, int length)**

**4.1.3.2 void Rocblas\_load\_module (const char \* name)**

Calls [Rocblas](#) initialization function.

**4.1.3.3 void ROCBLAS\_UNLOAD\_MODULE (const char \* name, int length)**

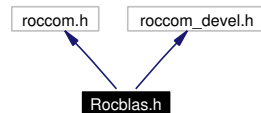
**4.1.3.4 void Rocblas\_unload\_module (const char \* name)**

## 4.2 Rocblas.h File Reference

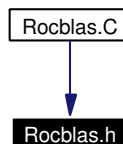
```
#include "roccom.h"
```

```
#include "roccom_devel.h"
```

Include dependency graph for Rocblas.h:



This graph shows which files directly or indirectly include this file:



### Compounds

- class [Rocblas](#)

### Functions

- void [Rocblas\\_load\\_module](#) (const char \*name)  
*Calls Rocblas initialization function.*
- void [Rocblas\\_unload\\_module](#) (const char \*name)

#### 4.2.1 Detailed Description

Definition for [Rocblas](#) API.

#### 4.2.2 Function Documentation

##### 4.2.2.1 void Rocblas\_load\_module (const char \* name)

Calls [Rocblas](#) initialization function.

##### 4.2.2.2 void Rocblas\_unload\_module (const char \* name)

## Index

add  
    Rocblas, 5

add\_scalar  
    Rocblas, 5

axpy  
    Rocblas, 6

axpy\_gen  
    Rocblas, 6

axpy\_scalar  
    Rocblas, 6

BLAS\_SCALAR  
    Rocblas, 5

BLAS\_SCNE  
    Rocblas, 5

BLAS\_VEC  
    Rocblas, 5

BLAS\_VECNE  
    Rocblas, 5

BLAS\_VOID  
    Rocblas, 5

calc  
    Rocblas, 6

calcChoose  
    Rocblas, 6

calcDot  
    Rocblas, 6

copy  
    Rocblas, 6

copy\_scalar  
    Rocblas, 6

div  
    Rocblas, 7

div\_scalar  
    Rocblas, 7

dot  
    Rocblas, 7

dot\_MPI  
    Rocblas, 7

dot\_scalar  
    Rocblas, 7

dot\_scalar\_MPI  
    Rocblas, 7

finalize  
    Rocblas, 7

gen2arg  
    Rocblas, 7

get\_stride  
    Rocblas, 7

getref  
    Rocblas, 7, 8

init  
    Rocblas, 8

limit1  
    Rocblas, 8

max\_MPI  
    Rocblas, 8

max\_scalar\_MPI  
    Rocblas, 8

min\_MPI  
    Rocblas, 8

min\_scalar\_MPI  
    Rocblas, 8

mul  
    Rocblas, 8

mul\_scalar  
    Rocblas, 8

neg  
    Rocblas, 9

nrm2  
    Rocblas, 9

nrm2\_MPI  
    Rocblas, 9

nrm2\_scalar  
    Rocblas, 9

nrm2\_scalar\_MPI  
    Rocblas, 9

operator()



- Rocblas::assn, 10
- Rocblas::nega, 11
- Rocblas::swapp, 11
- reduce\_MPI
  - Rocblas, 9
- reduce\_scalar\_MPI
  - Rocblas, 9
- Rocblas, 2
  - add, 5
  - add\_scalar, 5
  - axpy, 6
  - axpy\_gen, 6
  - axpy\_scalar, 6
  - BLAS\_SCALAR, 5
  - BLAS\_SCNE, 5
  - BLAS\_VEC, 5
  - BLAS\_VECNE, 5
  - BLAS\_VOID, 5
  - calc, 6
  - calcChoose, 6
  - calcDot, 6
  - copy, 6
  - copy\_scalar, 6
  - div, 7
  - div\_scalar, 7
  - dot, 7
  - dot\_MPI, 7
  - dot\_scalar, 7
  - dot\_scalar\_MPI, 7
  - finalize, 7
  - gen2arg, 7
  - get\_stride, 7
  - getref, 7, 8
  - init, 8
  - limit1, 8
  - max\_MPI, 8
  - max\_scalar\_MPI, 8
  - min\_MPI, 8
  - min\_scalar\_MPI, 8
  - mul, 8
  - mul\_scalar, 8
  - neg, 9
  - nrm2, 9
  - nrm2\_MPI, 9
  - nrm2\_scalar, 9
  - nrm2\_scalar\_MPI, 9
  - reduce\_MPI, 9
  - reduce\_scalar\_MPI, 9
  - sub, 9
  - sub\_scalar, 9
  - sum\_MPI, 10
  - sum\_scalar\_MPI, 10
  - swap, 10
- Rocblas.C, 11
  - ROCLAS\_LOAD\_MODULE, 13
  - Rocblas\_load\_module, 13
  - ROCLAS\_UNLOAD\_MODULE, 13
  - Rocblas\_unload\_module, 13
- Rocblas.h, 13
  - Rocblas\_load\_module, 14
  - Rocblas\_unload\_module, 14
- Rocblas::assn, 10
  - operator(), 10
- Rocblas::nega, 11
  - operator(), 11
- Rocblas::swapp, 11
  - operator(), 11
- ROCLAS\_LOAD\_MODULE
  - Rocblas.C, 13
- Rocblas\_load\_module
  - Rocblas.C, 13
  - Rocblas.h, 14
- ROCLAS\_UNLOAD\_MODULE
  - Rocblas.C, 13
- Rocblas\_unload\_module
  - Rocblas.C, 13
  - Rocblas.h, 14
- sub
  - Rocblas, 9
- sub\_scalar
  - Rocblas, 9
- sum\_MPI
  - Rocblas, 10
- sum\_scalar\_MPI
  - Rocblas, 10
- swap
  - Rocblas, 10