

Dexter

0.01

Generated by Doxygen 1.8.17

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 Dexter Namespace Reference	7
4.1.1 Function Documentation	7
4.1.1.1 entryWiseAdd()	7
4.1.1.2 entryWiseDiv()	8
4.1.1.3 entryWiseExp()	8
4.1.1.4 entryWiseLog()	8
4.1.1.5 entryWiseProd()	8
4.1.1.6 entryWiseSub()	8
5 Class Documentation	9
5.1 Dexter::Tensor< T, N > Class Template Reference	9
5.1.1 Member Typedef Documentation	9
5.1.1.1 dat_map	10
5.1.1.2 uarr	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 Tensor() [1/2]	10
5.1.2.2 Tensor() [2/2]	10
5.1.2.3 ~Tensor()	10
5.1.3 Member Function Documentation	10
5.1.3.1 adtoid()	11
5.1.3.2 adtoid_nocheck()	11
5.1.3.3 adtostr()	11
5.1.3.4 adtostr_nocheck()	11
5.1.3.5 at() [1/2]	11
5.1.3.6 at() [2/2]	11
5.1.3.7 at_nocheck()	12
5.1.3.8 idtoad()	12
5.1.3.9 idtoad_nocheck()	12
5.1.3.10 idtostrad()	12
5.1.3.11 idtostrad_nocheck()	12
5.1.3.12 n_elem()	12
5.1.3.13 print()	12
6 File Documentation	13

6.1 Dexter/include/dxtrTensor.hxx File Reference	13
6.2 Dexter/main.cxx File Reference	14
6.2.1 Function Documentation	14
6.2.1.1 main()	14
6.2.1.2 Test1()	14
6.3 Dexter/src/dxtrTensor.cxx File Reference	14
Index	15

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Dexter	7
----------------------------------	---

Chapter 2

Class Index

2.1 Class List

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

Dexter::Tensor< T, N >	9
--	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Dexter/ main.cxx	14
Dexter/include/ dxtrTensor.hxx	13
Dexter/src/ dxtrTensor.cxx	14

Chapter 4

Namespace Documentation

4.1 Dexter Namespace Reference

Classes

- class [Tensor](#)

Functions

- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseAdd](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)
- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseSub](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)
- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseProd](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)
- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseDiv](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)
- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseExp](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)
- `template<class T , std::size_t N>`
[Tensor](#)< T, N > [entryWiseLog](#) (const [Tensor](#)< T, N > &lhs, const [Tensor](#)< T, N > &rhs)

4.1.1 Function Documentation

4.1.1.1 [entryWiseAdd\(\)](#)

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseAdd (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

4.1.1.2 entryWiseDiv()

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseDiv (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

4.1.1.3 entryWiseExp()

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseExp (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

4.1.1.4 entryWiseLog()

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseLog (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

4.1.1.5 entryWiseProd()

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseProd (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

4.1.1.6 entryWiseSub()

```
template<class T , std::size_t N>
Tensor<T, N> Dexter::entryWiseSub (
    const Tensor< T, N > & lhs,
    const Tensor< T, N > & rhs )
```

Chapter 5

Class Documentation

5.1 Dexter::Tensor< T, N > Class Template Reference

```
#include <dxtrTensor.hxx>
```

Public Types

- typedef std::array< std::size_t, N > [uarr](#)
- typedef std::unordered_map< std::size_t, T > [dat_map](#)

Public Member Functions

- [Tensor](#) (T default_val=0)
- [Tensor](#) (const [uarr](#) &dims, T default_val=0)
- [~Tensor](#) ()=default
- std::size_t [n_elem](#) () const
- [uarr idtoad](#) (const std::size_t &) const
- [uarr idtoad_noccheck](#) (const std::size_t &) const
- std::size_t [adtoid](#) (const [uarr](#) &) const
- std::size_t [adtoid_noccheck](#) (const [uarr](#) &) const
- T & [at](#) (std::size_t id)
- T & [at_noccheck](#) (std::size_t id)
- T & [at](#) (const [uarr](#) &address)
- std::string [adtostr](#) (const [uarr](#) &) const
- std::string [adtostr_noccheck](#) (const [uarr](#) &) const
- std::string [idtostrad](#) (const std::size_t &) const
- std::string [idtostrad_noccheck](#) (const std::size_t &) const
- void [print](#) () const

5.1.1 Member Typedef Documentation

5.1.1.1 dat_map

```
template<class T , std::size_t N>
typedef std::unordered_map<std::size_t, T> Dexter::Tensor< T, N >::dat_map
```

5.1.1.2 uarr

```
template<class T , std::size_t N>
typedef std::array<std::size_t, N> Dexter::Tensor< T, N >::uarr
```

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Tensor() [1/2]

```
template<class T , std::size_t N>
Dexter::Tensor< T, N >::Tensor (
    T default_val = 0 )
```

5.1.2.2 Tensor() [2/2]

```
template<class T , std::size_t N>
Dexter::Tensor< T, N >::Tensor (
    const uarr & dims,
    T default_val = 0 )
```

5.1.2.3 ~Tensor()

```
template<class T , std::size_t N>
Dexter::Tensor< T, N >::~~Tensor ( ) [default]
```

5.1.3 Member Function Documentation

5.1.3.1 adtoid()

```
template<class T , std::size_t N>
std::size_t Dexter::Tensor< T, N >::adtoid (
    const uarr & ) const
```

5.1.3.2 adtoid_nocheck()

```
template<class T , std::size_t N>
std::size_t Dexter::Tensor< T, N >::adtoid_nocheck (
    const uarr & ) const
```

5.1.3.3 adtostr()

```
template<class T , std::size_t N>
std::string Dexter::Tensor< T, N >::adtostr (
    const uarr & ) const
```

5.1.3.4 adtostr_nocheck()

```
template<class T , std::size_t N>
std::string Dexter::Tensor< T, N >::adtostr_nocheck (
    const uarr & ) const
```

5.1.3.5 at() [1/2]

```
template<class T , std::size_t N>
T& Dexter::Tensor< T, N >::at (
    const uarr & address )
```

5.1.3.6 at() [2/2]

```
template<class T , std::size_t N>
T& Dexter::Tensor< T, N >::at (
    std::size_t id )
```

5.1.3.7 at_nocheck()

```
template<class T , std::size_t N>
T& Dexter::Tensor< T, N >::at_nocheck (
    std::size_t id )
```

5.1.3.8 idtoad()

```
template<class T , std::size_t N>
uarr Dexter::Tensor< T, N >::idtoad (
    const std::size_t & ) const
```

5.1.3.9 idtoad_nocheck()

```
template<class T , std::size_t N>
uarr Dexter::Tensor< T, N >::idtoad_nocheck (
    const std::size_t & ) const
```

5.1.3.10 idtostrad()

```
template<class T , std::size_t N>
std::string Dexter::Tensor< T, N >::idtostrad (
    const std::size_t & ) const
```

5.1.3.11 idtostrad_nocheck()

```
template<class T , std::size_t N>
std::string Dexter::Tensor< T, N >::idtostrad_nocheck (
    const std::size_t & ) const
```

5.1.3.12 n_elem()

```
template<class T , std::size_t N>
std::size_t Dexter::Tensor< T, N >::n_elem ( ) const
```

5.1.3.13 print()

```
template<class T , std::size_t N>
void Dexter::Tensor< T, N >::print ( ) const
```

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

- Dexter/include/[dxtrTensor.hxx](#)

Chapter 6

File Documentation

6.1 Dexter/include/dxtrTensor.hxx File Reference

```
#include <array>
#include <vector>
#include <unordered_map>
#include <exception>
#include <string>
#include <sstream>
#include <iostream>
#include "dxtrTensor.txx"
```

Classes

- class [Dexter::Tensor< T, N >](#)
- class [Dexter::Tensor< T, N >](#)

Namespaces

- [Dexter](#)

Functions

- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseAdd (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`
- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseSub (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`
- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseProd (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`
- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseDiv (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`
- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseExp (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`
- `template<class T, std::size_t N>`
`Tensor< T, N > Dexter::entryWiseLog (const Tensor< T, N > &lhs, const Tensor< T, N > &rhs)`

6.2 Dexter/main.cxx File Reference

```
#include <iostream>
#include <chrono>
#include "dxtrTensor.hxx"
```

Functions

- bool [Test1](#) ()
- int [main](#) ()

6.2.1 Function Documentation

6.2.1.1 main()

```
int main ( )
```

6.2.1.2 Test1()

```
bool Test1 ( )
```

6.3 Dexter/src/dxtrTensor.cxx File Reference

```
#include "dxtrTensor.hxx"
```

Index

`~Tensor`
 Dexter::Tensor< T, N >, [10](#)

`adtoid`
 Dexter::Tensor< T, N >, [10](#)

`adtoid_nocheck`
 Dexter::Tensor< T, N >, [11](#)

`adtostr`
 Dexter::Tensor< T, N >, [11](#)

`adtostr_nocheck`
 Dexter::Tensor< T, N >, [11](#)

`at`
 Dexter::Tensor< T, N >, [11](#)

`at_nocheck`
 Dexter::Tensor< T, N >, [11](#)

`dat_map`
 Dexter::Tensor< T, N >, [9](#)

`Dexter, 7`
 entryWiseAdd, [7](#)
 entryWiseDiv, [7](#)
 entryWiseExp, [8](#)
 entryWiseLog, [8](#)
 entryWiseProd, [8](#)
 entryWiseSub, [8](#)

`Dexter/include/dxtrTensor.hxx, 13`

`Dexter/main.cxx, 14`

`Dexter/src/dxtrTensor.cxx, 14`

`Dexter::Tensor< T, N >, 9`
 ~Tensor, [10](#)
 adtoid, [10](#)
 adtoid_nocheck, [11](#)
 adtostr, [11](#)
 adtostr_nocheck, [11](#)
 at, [11](#)
 at_nocheck, [11](#)
 dat_map, [9](#)
 idtoad, [12](#)
 idtoad_nocheck, [12](#)
 idtostrad, [12](#)
 idtostrad_nocheck, [12](#)
 n_elem, [12](#)
 print, [12](#)
 Tensor, [10](#)
 uarr, [10](#)

`entryWiseAdd`
 Dexter, [7](#)

`entryWiseDiv`
 Dexter, [7](#)

`entryWiseExp`
 Dexter, [8](#)

`entryWiseLog`
 Dexter, [8](#)

`entryWiseProd`
 Dexter, [8](#)

`entryWiseSub`
 Dexter, [8](#)

`idtoad`
 Dexter::Tensor< T, N >, [12](#)

`idtoad_nocheck`
 Dexter::Tensor< T, N >, [12](#)

`idtostrad`
 Dexter::Tensor< T, N >, [12](#)

`idtostrad_nocheck`
 Dexter::Tensor< T, N >, [12](#)

`main`
 main.cxx, [14](#)

`main.cxx`
 main, [14](#)
 Test1, [14](#)

`n_elem`
 Dexter::Tensor< T, N >, [12](#)

`print`
 Dexter::Tensor< T, N >, [12](#)

`Tensor`
 Dexter::Tensor< T, N >, [10](#)

`Test1`
 main.cxx, [14](#)

`uarr`
 Dexter::Tensor< T, N >, [10](#)