Fast CDR

Version 1.0.15

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 eprosima Namespace Reference

Namespaces

- fastcdr

4.2 eprosima::fastcdr Namespace Reference

Namespaces

- exception

Classes

- class _FastBuffer_iterator

  This class implements the iterator used to go through a FastBuffer.
- class Cdr

  This class offers an interface to serialize/deserialize some basic types using CDR protocol inside an eprosima::fastcdr::FastBuffer.
- class FastBuffer

  This class represents a stream of bytes that contains (or will contain) serialized data.
- class FastCdr

  This class offers an interface to serialize/deserialize some basic types using a modified CDR protocol inside a eprosima::FastBuffer.

4.3 eprosima::fastcdr::exception Namespace Reference

Classes

- class BadParamException

  This class is thrown as an exception when a invalid parameter was being serialized.
- class Exception

  This abstract class is used to create exceptions.
- class NotEnoughMemoryException

  This class is thrown as an exception when the buffer's internal memory reachs its size limit.

Chapter 5

Class Documentation

5.1 _FastBuffer_iterator Class Reference

This class implements the iterator used to go through a FastBuffer.

`#include <FastBuffer.h>`

Public Member Functions

- _FastBuffer_iterator ()
  Default constructor.
- _FastBuffer_iterator (char *buffer, size_t index)
  Constructor.
- void operator<< (const _FastBuffer_iterator &iterator)
  This operator changes the iterator's raw buffer.
- void operator>> (const _FastBuffer_iterator &iterator)
  This operator changes the position where the iterator points.
- template<typename _T >
  void operator<< (const _T &data)
  This operator copies a data in the raw buffer.
- template<typename _T >
  void operator>> (_T &data)
  This operator copies data from the raw buffer to a variable.
- void memcopy (const void *src, const size_t size)
  This function copies a buffer into the raw buffer.
- void rmemcopy (void *dst, const size_t size)
  This function copies from the raw buffer to a external buffer.
- void operator+= (size_t numBytes)
  This function increments the position where the iterator points.
- size_t operator- (const _FastBuffer_iterator &it) const
  This operator returns the subtraction of the current interator's position and the source iterator's position.
- _FastBuffer_iterator operator++ ()
  This function increments the iterator in one the position.
- _FastBuffer_iterator operator++ (int)
  This function increments the iterator in one the position.
- char * operator& ()
  This function returns the current position in the raw buffer.

5.1.1 Detailed Description

This class implements the iterator used to go through a FastBuffer.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 _FastBuffer_iterator() [1/2]

```
_FastBuffer_iterator ( )  [inline]
```

Default constructor.

The iterator points any position.

5.1.2.2 _FastBuffer_iterator() [2/2]

```
_FastBuffer_iterator (
            char * buffer,
            size_t index )  [inline], [explicit]
```

Constructor.

The iterator points to the indicated position.

Parameters

buffer	Pointer to the raw buffer.
index	Position of the raw buffer where the iterator will point.

5.1.3 Member Function Documentation

5.1.3.1 memcopy()

```
void memcopy (
            const void * src,
            const size_t size )  [inline]
```

This function copies a buffer into the raw buffer.

Parameters

src	The source buffer.
size	The number of bytes to be copied.

5.1.3.2 operator&()

```
char* operator& ( )  [inline]
```

This function returns the current position in the raw buffer.

Returns

The current position in the raw buffer.

5.1.3.3 operator++() [1/2]

```
_FastBuffer_iterator operator++ ( )  [inline]
```

This function increments the iterator in one the position.

Returns

The current iterator.

5.1.3.4 operator++() [2/2]

```
_FastBuffer_iterator operator++ (
            int  )  [inline]
```

This function increments the iterator in one the position.

Returns

The current iterator.

5.1.3.5 operator+=()

```
void operator+= (
            size_t numBytes )  [inline]
```

This function increments the position where the iterator points.

Parameters

numBytes	Number of bytes the iterator moves the position.

5.1.3.6 operator-()

```
size_t operator- (
            const _FastBuffer_iterator & it ) const  [inline]
```

This operator returns the subtraction of the current interator's position and the source iterator's position.

Parameters

it	Source iterator whose position is subtracted to the current iterator's position.

Returns

The result of subtract the current iterator's position and the source iterator's position.

5.1.3.7 operator<<() [1/2]

```
void operator<< (
            const _FastBuffer_iterator & iterator )  [inline]
```

This operator changes the iterator's raw buffer.

This operator makes the iterator point to the same position but in another raw buffer. The new raw buffer is the same than the source iterator's.

Parameters

iterator	The source iterator. The iterator will use the source iterator's raw buffer after this operation.

5.1.3.8 operator<<() [2/2]

```
void operator<< (
            const _T & data )  [inline]
```

This operator copies a data in the raw buffer.

The copy uses the size of the data type.

Parameters

data	Data to be copied. Cannot be NULL.

5.1.3.9 operator>>() [1/2]

```
void operator>> (
            _T & data )  [inline]
```

This operator copies data from the raw buffer to a variable.

The copy uses the size of the data type.

Parameters

data	Data to be filled.

5.1.3.10 operator>>() [2/2]

```
void operator>> (
            const _FastBuffer_iterator & iterator )  [inline]
```

This operator changes the position where the iterator points.

This operator takes the index of the source iterator, but the iterator continues using its raw buffer.

Parameters

iterator	The source iterator. The iterator will use the source's iterator index to point to its own raw buffer.

5.1.3.11 rmemcopy()

```
void rmemcopy (
            void * dst,
            const size_t size )  [inline]
```

This function copies from the raw buffer to a external buffer.

Parameters

dst	The destination buffer.
size	The size of bytes to be copied.

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

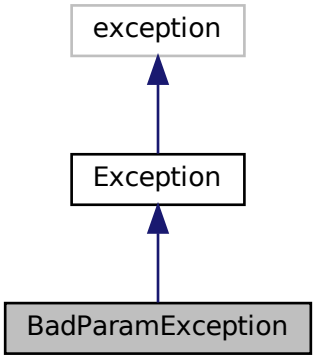
- include/fastcdr/FastBuffer.h

5.2 BadParamException Class Reference

This class is thrown as an exception when a invalid parameter was being serialized.

`#include <BadParamException.h>`

Inheritance diagram for BadParamException:



Public Member Functions

- Cdr_DllAPI BadParamException (const char ∗const &message) noexcept

  Default constructor.
- Cdr_DllAPI BadParamException (const BadParamException &ex) noexcept

  Default copy constructor.
- Cdr_DllAPI BadParamException & operator= (const BadParamException &ex) noexcept

  Assigment operation.
- virtual Cdr_DllAPI ∼BadParamException () noexcept

  Default constructor.
- virtual Cdr_DllAPI void raise () const

  This function throws the object as exception.

Static Public Attributes

- static const Cdr_DllAPI char ∗const BAD_PARAM_MESSAGE_DEFAULT

  Default message used in the library.

Additional Inherited Members

5.2.1 Detailed Description

This class is thrown as an exception when a invalid parameter was being serialized.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 BadParamException() [1/2]

```
Cdr_DllAPI BadParamException (
            const char *const & message )  [noexcept]
```

Default constructor.

Parameters

message	A error message. This message pointer is copied.

5.2.2.2 BadParamException() [2/2]

```
Cdr_DllAPI BadParamException (
            const BadParamException & ex )  [noexcept]
```

Default copy constructor.

Parameters

ex	BadParamException that will be copied.

5.2.2.3 ~BadParamException()

```
virtual Cdr_DllAPI ~BadParamException ( )  [virtual], [noexcept]
```

Default constructor.

5.2.3 Member Function Documentation

5.2.3.1 operator=()

```
Cdr_DllAPI BadParamException& operator= (
            const BadParamException & ex )  [noexcept]
```

Assigment operation.

Parameters

ex	BadParamException that will be copied.

5.2.3.2 raise()

```
virtual Cdr_DllAPI void raise ( ) const  [virtual]
```

This function throws the object as exception.

Implements Exception.

5.2.4 Member Data Documentation

5.2.4.1 BAD_PARAM_MESSAGE_DEFAULT

```
const Cdr_DllAPI char* const BAD_PARAM_MESSAGE_DEFAULT  [static]
```

Default message used in the library.

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

- include/fastcdr/exceptions/BadParamException.h

5.3 Cdr Class Reference

This class offers an interface to serialize/deserialize some basic types using CDR protocol inside an eprosima::fastcdr::FastBuffer.

```
#include <Cdr.h>
```

Classes

- class state

  This class stores the current state of a CDR serialization.

Public Types

- enum CdrType { CORBA_CDR, DDS_CDR }

  This enumeration represents the two kinds of CDR serialization supported by eprosima::fastcdr::CDR.
- enum DDSCdrPlFlag { DDS_CDR_WITHOUT_PL = 0x0, DDS_CDR_WITH_PL = 0x2 }

  This enumeration represents the two posible values of the flag that points if the content is a parameter list (only in DDS CDR).
- enum Endianness { BIG_ENDIANNESS = 0x0, LITTLE_ENDIANNESS = 0x1 }

  This enumeration represents endianness types.

Public Member Functions

- Cdr (FastBuffer &cdrBuffer, const Endianness endianness=DEFAULT_ENDIAN, const CdrType cdr↩ Type=CORBA_CDR)

  This constructor creates an eprosima::fastcdr::Cdr object that can serialize/deserialize the assigned buffer.
- Cdr & read_encapsulation ()

  This function reads the encapsulation of the CDR stream.
- Cdr & serialize_encapsulation ()

  This function writes the encapsulation of the CDR stream.
- DDSCdrPlFlag getDDSCdrPlFlag () const

  This function returns the parameter list flag when the CDR type is eprosima::fastcdr::DDS_CDR.
- void setDDSCdrPlFlag (DDSCdrPlFlag plFlag)

  This function sets the parameter list flag when the CDR type is eprosima::fastcdr::DDS_CDR.
- uint16_t getDDSCdrOptions () const

  This function returns the option flags when the CDR type is eprosima::fastcdr::DDS_CDR.
- void setDDSCdrOptions (uint16_t options)

  This function sets the option flags when the CDR type is eprosima::fastcdr::DDS_CDR.
- void changeEndianness (Endianness endianness)

  This function sets the current endianness used by the CDR type.
- Endianness endianness () const

  This function returns the current endianness used by the CDR type.
- bool jump (size_t numBytes)

  This function skips a number of bytes in the CDR stream buffer.
- void reset ()

  This function resets the current position in the buffer to the beginning.
- char * getBufferPointer ()

  This function returns the pointer to the current used buffer.
- char * getCurrentPosition ()

  This function returns the current position in the CDR stream.
- size_t getSerializedDataLength () const

  This function returns the length of the serialized data inside the stream.
- state getState ()

  This function returns the current state of the CDR serialization process.
- void setState (state &state)

  This function sets a previous state of the CDR serialization process;.
- bool moveAlignmentForward (size_t numBytes)

  This function moves the alignment forward.
- void resetAlignment ()

  This function resets the alignment to the current position in the buffer.
- Cdr & operator<< (const uint8_t octet_t)

This operator serializes an octet.

- Cdr & operator<< (const char char_t)

  This operator serializes a character.
- Cdr & operator<< (const int8_t int8)

  This operator serializes a int8_t.
- Cdr & operator<< (const uint16_t ushort_t)

  This operator serializes an unsigned short.
- Cdr & operator<< (const int16_t short_t)

  This operator serializes a short.
- Cdr & operator<< (const uint32_t ulong_t)

  This operator serializes an unsigned long.
- Cdr & operator<< (const int32_t long_t)

  This operator serializes a long.
- Cdr & operator<< (const wchar_t wchar)

  This operator serializes a wide-char.
- Cdr & operator<< (const uint64_t ulonglong_t)

  This operator serializes an unsigned long long.
- Cdr & operator<< (const int64_t longlong_t)

  This operator serializes a long long.
- Cdr & operator<< (const float float_t)

  This operator serializes a float.
- Cdr & operator<< (const double double_t)

  This operator serializes a double.
- Cdr & operator<< (const long double ldouble_t)

  This operator serializes a long double.
- Cdr & operator<< (const bool bool_t)

  This operator serializes a boolean.
- Cdr & operator<< (const char *string_t)

  This operator serializes a null-terminated c-string.
- Cdr & operator<< (char *string_t)

  This operator serializes a null-terminated c-string.
- Cdr & operator<< (const std::string &string_t)

  This operator serializes a string.
- Cdr & operator<< (const std::wstring &string_t)

  This operator serializes a wstring.
- template<class _T >
  Cdr & operator<< (const std::vector< _T > &vector_t)

  This operator template is used to serialize sequences.
- template<class _K , class _T >
  Cdr & operator<< (const std::map< _K, _T > &map_t)

  This operator template is used to serialize maps.
- template<class _T >
  Cdr & operator<< (const _T &type_t)

  This operator template is used to serialize any other non-basic type.
- Cdr & operator>> (uint8_t &octet_t)

  This operator deserializes an octet.
- Cdr & operator>> (char &char_t)

  This operator deserializes a character.
- Cdr & operator>> (int8_t &int8)

  This operator deserializes a int8_t.
- Cdr & operator>> (uint16_t &ushort_t)

This operator deserializes an unsigned short.

- Cdr & operator>> (int16_t &short_t)

  This operator deserializes a short.
- Cdr & operator>> (uint32_t &ulong_t)

  This operator deserializes an unsigned long.
- Cdr & operator>> (int32_t &long_t)

  This operator deserializes a long.
- Cdr & operator>> (wchar_t &wchar)

  This operator deserializes a wide-char.
- Cdr & operator>> (uint64_t &ulonglong_t)

  This operator deserializes a unsigned long long.
- Cdr & operator>> (int64_t &longlong_t)

  This operator deserializes a long long.
- Cdr & operator>> (float &float_t)

  This operator deserializes a float.
- Cdr & operator>> (double &double_t)

  This operator deserializes a double.
- Cdr & operator>> (long double &ldouble_t)

  This operator deserializes a long double.
- Cdr & operator>> (bool &bool_t)

  This operator deserializes a boolean.
- Cdr & operator>> (char *&string_t)

  This operator deserializes a null-terminated c-string.
- Cdr & operator>> (std::string &string_t)

  This operator deserializes a string.
- Cdr & operator>> (std::wstring &string_t)

  This operator deserializes a string.
- template<class _T >
  Cdr & operator>> (std::vector< _T > &vector_t)

  This operator template is used to deserialize sequences.
- template<class _K , class _T >
  Cdr & operator>> (std::map< _K, _T > &map_t)

  This operator template is used to deserialize maps.
- template<class _T >
  Cdr & operator>> (_T &type_t)

  This operator template is used to deserialize any other non-basic type.
- Cdr & serialize (const uint8_t octet_t)

  This function serializes an octet.
- Cdr & serialize (const uint8_t octet_t, Endianness endianness)

  This function serializes an octet with a different endianness.
- Cdr & serialize (const char char_t)

  This function serializes a character.
- Cdr & serialize (const char char_t, Endianness endianness)

  This function serializes a character with a different endianness.
- Cdr & serialize (const int8_t int8)

  This function serializes an int8_t.
- Cdr & serialize (const int8_t int8, Endianness endianness)

  This function serializes an int8_t with a different endianness.
- Cdr & serialize (const uint16_t ushort_t)

  This function serializes an unsigned short.
- Cdr & serialize (const uint16_t ushort_t, Endianness endianness)

This function serializes an unsigned short with a different endianness.

- Cdr & serialize (const int16_t short_t)

  This function serializes a short.
- Cdr & serialize (const int16_t short_t, Endianness endianness)

  This function serializes a short with a different endianness.
- Cdr & serialize (const uint32_t ulong_t)

  This function serializes an unsigned long.
- Cdr & serialize (const uint32_t ulong_t, Endianness endianness)

  This function serializes an unsigned long with a different endianness.
- Cdr & serialize (const int32_t long_t)

  This function serializes a long.
- Cdr & serialize (const int32_t long_t, Endianness endianness)

  This function serializes a long with a different endianness.
- Cdr & serialize (const wchar_t wchar)

  This function serializes a wide-char.
- Cdr & serialize (const wchar_t wchar, Endianness endianness)

  This function serializes a wide-char with a different endianness.
- Cdr & serialize (const uint64_t ulonglong_t)

  This function serializes an unsigned long long.
- Cdr & serialize (const uint64_t ulonglong_t, Endianness endianness)

  This function serializes an unsigned long long with a different endianness.
- Cdr & serialize (const int64_t longlong_t)

  This function serializes a long long.
- Cdr & serialize (const int64_t longlong_t, Endianness endianness)

  This function serializes a long long with a different endianness.
- Cdr & serialize (const float float_t)

  This function serializes a float.
- Cdr & serialize (const float float_t, Endianness endianness)

  This function serializes a float with a different endianness.
- Cdr & serialize (const double double_t)

  This function serializes a double.
- Cdr & serialize (const double double_t, Endianness endianness)

  This function serializes a double with a different endianness.
- Cdr & serialize (const long double ldouble_t)

  This function serializes a long double.
- Cdr & serialize (const long double ldouble_t, Endianness endianness)

  This function serializes a long double with a different endianness.
- Cdr & serialize (const bool bool_t)

  This function serializes a boolean.
- Cdr & serialize (const bool bool_t, Endianness endianness)

  This function serializes a boolean with a different endianness.
- Cdr & serialize (char ∗string_t)

  This function serializes a string.
- Cdr & serialize (const char ∗string_t)

  This function serializes a string.
- Cdr & serialize (const wchar_t ∗string_t)

  This function serializes a wstring.
- Cdr & serialize (const char ∗string_t, Endianness endianness)

  This function serializes a string with a different endianness.
- Cdr & serialize (const wchar_t ∗string_t, Endianness endianness)

  This function serializes a wstring with a different endianness.

- Cdr & serialize (const std::string &string_t)

  This function serializes a std::string.
- Cdr & serialize (const std::wstring &string_t)

  This function serializes a std::wstring.
- Cdr & serialize (const std::string &string_t, Endianness endianness)

  This function serializes a std::string with a different endianness.
- template<class _T >
  Cdr & serialize (const std::vector< _T > &vector_t)

  This function template serializes a sequence.
- template<class _K , class _T >
  Cdr & serialize (const std::map< _K, _T > &map_t)

  This function template serializes a map.
- template<class _T >
  Cdr & serialize (const std::vector< _T > &vector_t, Endianness endianness)

  This function template serializes a sequence with a different endianness.
- template<class _T >
  Cdr & serialize (const _T &type_t)

  This function template serializes a non-basic object.
- Cdr & serializeArray (const uint8_t *octet_t, size_t numElements)

  This function serializes an array of octets.
- Cdr & serializeArray (const uint8_t *octet_t, size_t numElements, Endianness endianness)

  This function serializes an array of octets with a different endianness.
- Cdr & serializeArray (const char *char_t, size_t numElements)

  This function serializes an array of characters.
- Cdr & serializeArray (const char *char_t, size_t numElements, Endianness endianness)

  This function serializes an array of characters with a different endianness.
- Cdr & serializeArray (const int8_t *int8, size_t numElements)

  This function serializes an array of int8_t.
- Cdr & serializeArray (const int8_t *int8, size_t numElements, Endianness endianness)

  This function serializes an array of int8_t with a different endianness.
- Cdr & serializeArray (const uint16_t *ushort_t, size_t numElements)

  This function serializes an array of unsigned shorts.
- Cdr & serializeArray (const uint16_t *ushort_t, size_t numElements, Endianness endianness)

  This function serializes an array of unsigned shorts with a different endianness.
- Cdr & serializeArray (const int16_t *short_t, size_t numElements)

  This function serializes an array of shorts.
- Cdr & serializeArray (const int16_t *short_t, size_t numElements, Endianness endianness)

  This function serializes an array of shorts with a different endianness.
- Cdr & serializeArray (const uint32_t *ulong_t, size_t numElements)

  This function serializes an array of unsigned longs.
- Cdr & serializeArray (const uint32_t *ulong_t, size_t numElements, Endianness endianness)

  This function serializes an array of unsigned longs with a different endianness.
- Cdr & serializeArray (const int32_t *long_t, size_t numElements)

  This function serializes an array of longs.
- Cdr & serializeArray (const int32_t *long_t, size_t numElements, Endianness endianness)

  This function serializes an array of longs with a different endianness.
- Cdr & serializeArray (const wchar_t *wchar, size_t numElements)

  This function serializes an array of wide-chars.
- Cdr & serializeArray (const wchar_t *wchar, size_t numElements, Endianness endianness)

  This function serializes an array of wide-chars with a different endianness.
- Cdr & serializeArray (const uint64_t *ulonglong_t, size_t numElements)

This function serializes an array of unsigned long longs.

- Cdr & serializeArray (const uint64_t *ulonglong_t, size_t numElements, Endianness endianness)

  This function serializes an array of unsigned long longs with a different endianness.
- Cdr & serializeArray (const int64_t *longlong_t, size_t numElements)

  This function serializes an array of long longs.
- Cdr & serializeArray (const int64_t *longlong_t, size_t numElements, Endianness endianness)

  This function serializes an array of long longs with a different endianness.
- Cdr & serializeArray (const float *float_t, size_t numElements)

  This function serializes an array of floats.
- Cdr & serializeArray (const float *float_t, size_t numElements, Endianness endianness)

  This function serializes an array of floats with a different endianness.
- Cdr & serializeArray (const double *double_t, size_t numElements)

  This function serializes an array of doubles.
- Cdr & serializeArray (const double *double_t, size_t numElements, Endianness endianness)

  This function serializes an array of doubles with a different endianness.
- Cdr & serializeArray (const long double *ldouble_t, size_t numElements)

  This function serializes an array of long doubles.
- Cdr & serializeArray (const long double *ldouble_t, size_t numElements, Endianness endianness)

  This function serializes an array of long doubles with a different endianness.
- Cdr & serializeArray (const bool *bool_t, size_t numElements)

  This function serializes an array of booleans.
- Cdr & serializeArray (const bool *bool_t, size_t numElements, Endianness endianness)

  This function serializes an array of booleans with a different endianness.
- Cdr & serializeArray (const std::string *string_t, size_t numElements)

  This function serializes an array of strings.
- Cdr & serializeArray (const std::wstring *string_t, size_t numElements)

  This function serializes an array of wide-strings.
- Cdr & serializeArray (const std::string *string_t, size_t numElements, Endianness endianness)

  This function serializes an array of strings with a different endianness.
- Cdr & serializeArray (const std::wstring *string_t, size_t numElements, Endianness endianness)

  This function serializes an array of wide-strings with a different endianness.
- template<class _T >
  Cdr & serializeArray (const std::vector< _T > *vector_t, size_t numElements)

  This function template serializes an array of sequences of objects.
- template<class _T >
  Cdr & serializeArray (const _T *type_t, size_t numElements)

  This function template serializes an array of non-basic objects.
- template<class _T >
  Cdr & serializeArray (const _T *type_t, size_t numElements, Endianness endianness)

  This function template serializes an array of non-basic objects with a different endianness.
- template<class _T >
  Cdr & serializeSequence (const _T *sequence_t, size_t numElements)

  This function template serializes a raw sequence.
- template<class _T >
  Cdr & serializeSequence (const _T *sequence_t, size_t numElements, Endianness endianness)

  This function template serializes a raw sequence with a different endianness.
- Cdr & deserialize (uint8_t &octet_t)

  This function deserializes an octet.
- Cdr & deserialize (uint8_t &octet_t, Endianness endianness)

  This function deserializes an octet with a different endianness.
- Cdr & deserialize (char &char_t)

This function deserializes a character.

- Cdr & deserialize (char &char_t, Endianness endianness)

  This function deserializes a character with a different endianness.
- Cdr & deserialize (int8_t &int8)

  This function deserializes an int8_t.
- Cdr & deserialize (int8_t &int8, Endianness endianness)

  This function deserializes an int8_t with a different endianness.
- Cdr & deserialize (uint16_t &ushort_t)

  This function deserializes an unsigned short.
- Cdr & deserialize (uint16_t &ushort_t, Endianness endianness)

  This function deserializes an unsigned short with a different endianness.
- Cdr & deserialize (int16_t &short_t)

  This function deserializes a short.
- Cdr & deserialize (int16_t &short_t, Endianness endianness)

  This function deserializes a short with a different endianness.
- Cdr & deserialize (uint32_t &ulong_t)

  This function deserializes an unsigned long.
- Cdr & deserialize (uint32_t &ulong_t, Endianness endianness)

  This function deserializes an unsigned long with a different endianness.
- Cdr & deserialize (int32_t &long_t)

  This function deserializes a long.
- Cdr & deserialize (int32_t &long_t, Endianness endianness)

  This function deserializes a long with a different endianness.
- Cdr & deserialize (wchar_t &wchar)

  This function deserializes a wide-char.
- Cdr & deserialize (wchar_t &wchar, Endianness endianness)

  This function deserializes a wide-char with a different endianness.
- Cdr & deserialize (uint64_t &ulonglong_t)

  This function deserializes an unsigned long long.
- Cdr & deserialize (uint64_t &ulonglong_t, Endianness endianness)

  This function deserializes an unsigned long long with a different endianness.
- Cdr & deserialize (int64_t &longlong_t)

  This function deserializes a long long.
- Cdr & deserialize (int64_t &longlong_t, Endianness endianness)

  This function deserializes a long long with a different endianness.
- Cdr & deserialize (float &float_t)

  This function deserializes a float.
- Cdr & deserialize (float &float_t, Endianness endianness)

  This function deserializes a float with a different endianness.
- Cdr & deserialize (double &double_t)

  This function deserializes a double.
- Cdr & deserialize (double &double_t, Endianness endianness)

  This function deserializes a double with a different endianness.
- Cdr & deserialize (long double &ldouble_t)

  This function deserializes a long double.
- Cdr & deserialize (long double &ldouble_t, Endianness endianness)

  This function deserializes a long double with a different endianness.
- Cdr & deserialize (bool &bool_t)

  This function deserializes a boolean.
- Cdr & deserialize (bool &bool_t, Endianness endianness)

  This function deserializes a boolean with a different endianness.

- Cdr & deserialize (char *&string_t)

  This function deserializes a string.
- Cdr & deserialize (wchar_t *&string_t)

  This function deserializes a wide string.
- Cdr & deserialize (char *&string_t, Endianness endianness)

  This function deserializes a string with a different endianness.
- Cdr & deserialize (wchar_t *&string_t, Endianness endianness)

  This function deserializes a wide string with a different endianness.
- Cdr & deserialize (std::string &string_t)

  This function deserializes a std::string.
- Cdr & deserialize (std::wstring &string_t)

  This function deserializes a std::string.
- Cdr & deserialize (std::string &string_t, Endianness endianness)

  This function deserializes a string with a different endianness.
- Cdr & deserialize (std::wstring &string_t, Endianness endianness)

  This function deserializes a string with a different endianness.
- template<class _T >
  Cdr & deserialize (std::vector< _T > &vector_t)

  This function template deserializes a sequence.
- template<class _K , class _T >
  Cdr & deserialize (std::map< _K, _T > &map_t)

  This function template deserializes a map.
- template<class _T >
  Cdr & deserialize (std::vector< _T > &vector_t, Endianness endianness)

  This function template deserializes a sequence with a different endianness.
- template<class _T >
  Cdr & deserialize (_T &type_t)

  This function template deserializes a non-basic object.
- Cdr & deserializeArray (uint8_t *octet_t, size_t numElements)

  This function deserializes an array of octets.
- Cdr & deserializeArray (uint8_t *octet_t, size_t numElements, Endianness endianness)

  This function deserializes an array of octets with a different endianness.
- Cdr & deserializeArray (char *char_t, size_t numElements)

  This function deserializes an array of characters.
- Cdr & deserializeArray (char *char_t, size_t numElements, Endianness endianness)

  This function deserializes an array of characters with a different endianness.
- Cdr & deserializeArray (int8_t *int8, size_t numElements)

  This function deserializes an array of int8_t.
- Cdr & deserializeArray (int8_t *int8, size_t numElements, Endianness endianness)

  This function deserializes an array of int8_t with a different endianness.
- Cdr & deserializeArray (uint16_t *ushort_t, size_t numElements)

  This function deserializes an array of unsigned shorts.
- Cdr & deserializeArray (uint16_t *ushort_t, size_t numElements, Endianness endianness)

  This function deserializes an array of unsigned shorts with a different endianness.
- Cdr & deserializeArray (int16_t *short_t, size_t numElements)

  This function deserializes an array of shorts.
- Cdr & deserializeArray (int16_t *short_t, size_t numElements, Endianness endianness)

  This function deserializes an array of shorts with a different endianness.
- Cdr & deserializeArray (uint32_t *ulong_t, size_t numElements)

  This function deserializes an array of unsigned longs.
- Cdr & deserializeArray (uint32_t *ulong_t, size_t numElements, Endianness endianness)

This function deserializes an array of unsigned longs with a different endianness.

- Cdr & deserializeArray (int32_t *long_t, size_t numElements)

  This function deserializes an array of longs.
- Cdr & deserializeArray (int32_t *long_t, size_t numElements, Endianness endianness)

  This function deserializes an array of longs with a different endianness.
- Cdr & deserializeArray (wchar_t *wchar, size_t numElements)

  This function deserializes an array of wide-chars.
- Cdr & deserializeArray (wchar_t *wchar, size_t numElements, Endianness endianness)

  This function deserializes an array of wide-chars with a different endianness.
- Cdr & deserializeArray (uint64_t *ulonglong_t, size_t numElements)

  This function deserializes an array of unsigned long longs.
- Cdr & deserializeArray (uint64_t *ulonglong_t, size_t numElements, Endianness endianness)

  This function deserializes an array of unsigned long longs with a different endianness.
- Cdr & deserializeArray (int64_t *longlong_t, size_t numElements)

  This function deserializes an array of long longs.
- Cdr & deserializeArray (int64_t *longlong_t, size_t numElements, Endianness endianness)

  This function deserializes an array of long longs with a different endianness.
- Cdr & deserializeArray (float *float_t, size_t numElements)

  This function deserializes an array of floats.
- Cdr & deserializeArray (float *float_t, size_t numElements, Endianness endianness)

  This function deserializes an array of floats with a different endianness.
- Cdr & deserializeArray (double *double_t, size_t numElements)

  This function deserializes an array of doubles.
- Cdr & deserializeArray (double *double_t, size_t numElements, Endianness endianness)

  This function deserializes an array of doubles with a different endianness.
- Cdr & deserializeArray (long double *ldouble_t, size_t numElements)

  This function deserializes an array of long doubles.
- Cdr & deserializeArray (long double *ldouble_t, size_t numElements, Endianness endianness)

  This function deserializes an array of long doubles with a different endianness.
- Cdr & deserializeArray (bool *bool_t, size_t numElements)

  This function deserializes an array of booleans.
- Cdr & deserializeArray (bool *bool_t, size_t numElements, Endianness endianness)

  This function deserializes an array of booleans with a different endianness.
- Cdr & deserializeArray (std::string *string_t, size_t numElements)

  This function deserializes an array of strings.
- Cdr & deserializeArray (std::wstring *string_t, size_t numElements)

  This function deserializes an array of wide-strings.
- Cdr & deserializeArray (std::string *string_t, size_t numElements, Endianness endianness)

  This function deserializes an array of strings with a different endianness.
- Cdr & deserializeArray (std::wstring *string_t, size_t numElements, Endianness endianness)

  This function deserializes an array of wide-strings with a different endianness.
- template<class _T >
  Cdr & deserializeArray (std::vector< _T > *vector_t, size_t numElements)

  This function deserializes an array of sequences of objects.
- template<class _T >
  Cdr & deserializeArray (_T *type_t, size_t numElements)

  This function template deserializes an array of non-basic objects.
- template<class _T >
  Cdr & deserializeArray (_T *type_t, size_t numElements, Endianness endianness)

  This function template deserializes an array of non-basic objects with a different endianness.

- template<class _T >
  Cdr & deserializeSequence (_T *&sequence_t, size_t &numElements)

  This function template deserializes a raw sequence.
- template<class _T >
  Cdr & deserializeSequence (_T *&sequence_t, size_t &numElements, Endianness endianness)

  This function template deserializes a raw sequence with a different endianness.

Static Public Member Functions

- static size_t alignment (size_t current_alignment, size_t dataSize)

  Get the number of bytes needed to align a position to certain data size.

Static Public Attributes

- static const Endianness DEFAULT_ENDIAN

  Default endiness in the system.

5.3.1 Detailed Description

This class offers an interface to serialize/deserialize some basic types using CDR protocol inside an eprosima::fastcdr::FastBuffer.

5.3.2 Member Enumeration Documentation

5.3.2.1 CdrType

`enum CdrType`

This enumeration represents the two kinds of CDR serialization supported by eprosima::fastcdr::CDR.

Enumerator

CORBA_CDR	Common CORBA CDR serialization.
DDS_CDR	DDS CDR serialization.

5.3.2.2 DDSCdrPlFlag

`enum DDSCdrPlFlag`

This enumeration represents the two posible values of the flag that points if the content is a parameter list (only in DDS CDR).

Enumerator

DDS_CDR_WITHOUT_PL	Specifies that the content is not a parameter list.
DDS_CDR_WITH_PL	Specifies that the content is a parameter list.

5.3.2.3 Endianness

```
enum Endianness
```

This enumeration represents endianness types.

Enumerator

BIG_ENDIANNESS	Big endianness.
LITTLE_ENDIANNESS	Little endianness.

5.3.3 Constructor & Destructor Documentation

5.3.3.1 Cdr()

```
Cdr (
            FastBuffer & cdrBuffer,
            const Endianness endianness = DEFAULT_ENDIAN,
            const CdrType cdrType = CORBA_CDR )
```

This constructor creates an eprosima::fastcdr::Cdr object that can serialize/deserialize the assigned buffer.

Parameters

cdrBuffer	A reference to the buffer that contains (or will contain) the CDR representation.
endianness	The initial endianness that will be used. The default value is the endianness of the system.
cdrType	Represents the type of CDR that will be used in serialization/deserialization. The default value is CORBA CDR.

5.3.4 Member Function Documentation

5.3.4.1 alignment()

```
static size_t alignment (
            size_t current_alignment,
```

```
            size_t dataSize )  [inline], [static]
```

Get the number of bytes needed to align a position to certain data size.

Parameters

current_alignment	Position to be aligned.
dataSize	Size of next data to process (should be power of two).

Returns

Number of required alignment bytes.

5.3.4.2 changeEndianness()

```
void changeEndianness (
            Endianness endianness )
```

This function sets the current endianness used by the CDR type.

Parameters

endianness	The new endianness value.

5.3.4.3 deserialize() [1/40]

```
Cdr& deserialize (
            _T & type_t )  [inline]
```

This function template deserializes a non-basic object.

Parameters

type_t	The variable that will store the object read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.4 deserialize() [2/40]

```
Cdr& deserialize (
            bool & bool_t )
```

This function deserializes a boolean.

Parameters

bool_t	The variable that will store the boolean read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.3.4.5 deserialize() [3/40]

```
Cdr& deserialize (
            bool & bool_t,
            Endianness endianness )  [inline]
```

This function deserializes a boolean with a different endianness.

Parameters

bool_t	The variable that will store the boolean read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.3.4.6 deserialize() [4/40]

```
Cdr& deserialize (
            char & char_t )
```

This function deserializes a character.

Parameters

char_t	The variable that will store the character read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.7 deserialize() [5/40]

```
Cdr& deserialize (
            char & char_t,
            Endianness endianness )  [inline]
```

This function deserializes a character with a different endianness.

Parameters

char_t	The variable that will store the character read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.8 deserialize() [6/40]

```
Cdr& deserialize (
            char *& string_t )
```

This function deserializes a string.

This function allocates memory to store the string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string_t	The pointer that will point to the string read from the buffer. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.9 deserialize() [7/40]

```
Cdr& deserialize (
            char *& string_t,
            Endianness endianness )
```

This function deserializes a string with a different endianness.

This function allocates memory to store the string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string_t	The pointer that will point to the string read from the buffer.
endianness	Endianness that will be used in the deserialization of this value. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.10 deserialize() [8/40]

```
Cdr& deserialize (
            double & double_t )
```

This function deserializes a double.

Parameters

double_t	The variable that will store the double read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.11 deserialize() [9/40]

```
Cdr& deserialize (
            double & double_t,
            Endianness endianness )
```

This function deserializes a double with a different endianness.

Parameters

double_t	The variable that will store the double read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.12 deserialize() [10/40]

```
Cdr& deserialize (
            float & float_t )
```

This function deserializes a float.

Parameters

float_t	The variable that will store the float read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.13 deserialize() [11/40]

```
Cdr& deserialize (
            float & float_t,
            Endianness endianness )
```

This function deserializes a float with a different endianness.

Parameters

float_t	The variable that will store the float read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.14 deserialize() [12/40]

```
Cdr& deserialize (
            int16_t & short_t )
```

This function deserializes a short.

Parameters

short_t	The variable that will store the short read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.15 deserialize() [13/40]

```
Cdr& deserialize (
            int16_t & short_t,
            Endianness endianness )
```

This function deserializes a short with a different endianness.

Parameters

short_t	The variable that will store the short read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.16 deserialize() [14/40]

```
Cdr& deserialize (
            int32_t & long_t )
```

This function deserializes a long.

Parameters

long_t	The variable that will store the long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.17 deserialize() [15/40]

```
Cdr& deserialize (
            int32_t & long_t,
            Endianness endianness )
```

This function deserializes a long with a different endianness.

Parameters

long_t	The variable that will store the long read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.18 deserialize() [16/40]

```
Cdr& deserialize (
            int64_t & longlong_t )
```

This function deserializes a long long.

Parameters

longlong_t	The variable that will store the long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.19 deserialize() [17/40]

```
Cdr& deserialize (
            int64_t & longlong_t,
            Endianness endianness )
```

This function deserializes a long long with a different endianness.

Parameters

longlong_t	The variable that will store the long long read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.20 deserialize() [18/40]

```
Cdr& deserialize (
            int8_t & int8 )  [inline]
```

This function deserializes an int8_t.

Parameters

int8	The variable that will store the int8_t read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.21 deserialize() [19/40]

```
Cdr& deserialize (
            int8_t & int8,
            Endianness endianness )  [inline]
```

This function deserializes an int8_t with a different endianness.

Parameters

int8	The variable that will store the int8_t read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.22 deserialize() [20/40]

```
Cdr& deserialize (
            long double & ldouble_t )
```

This function deserializes a long double.

Parameters

ldouble_t	The variable that will store the long double read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

Note

Due to internal representation differences, WIN32 and ∗NIX like systems are not compatible.

5.3.4.23 deserialize() [21/40]

```
Cdr& deserialize (
            long double & ldouble_t,
            Endianness endianness )
```

This function deserializes a long double with a different endianness.

Parameters

ldouble_t	The variable that will store the long double read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

Note

Due to internal representation differences, WIN32 and ∗NIX like systems are not compatible.

5.3.4.24 deserialize() [22/40]

```
Cdr& deserialize (
            std::map< _K, _T > & map_t )  [inline]
```

This function template deserializes a map.

Parameters

map_t	The variable that will store the map read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.25 deserialize() [23/40]

```
Cdr& deserialize (
            std::string & string_t )  [inline]
```

This function deserializes a std::string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.26 deserialize() [24/40]

```
Cdr& deserialize (
            std::string & string_t,
            Endianness endianness )  [inline]
```

This function deserializes a string with a different endianness.

Parameters

string_t	The variable that will store the string read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.27 deserialize() [25/40]

```
Cdr& deserialize (
            std::vector< _T > & vector_t )  [inline]
```

This function template deserializes a sequence.

Parameters

vector_t	The variable that will store the sequence read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.28 deserialize() [26/40]

```
Cdr& deserialize (
            std::vector< _T > & vector_t,
            Endianness endianness )  [inline]
```

This function template deserializes a sequence with a different endianness.

Parameters

vector_t	The variable that will store the sequence read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.29 deserialize() [27/40]

```
Cdr& deserialize (
            std::wstring & string_t )  [inline]
```

This function deserializes a std::string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.30 deserialize() [28/40]

```
Cdr& deserialize (
            std::wstring & string_t,
            Endianness endianness )  [inline]
```

This function deserializes a string with a different endianness.

Parameters

string_t	The variable that will store the string read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.31 deserialize() [29/40]

```
Cdr& deserialize (
            uint16_t & ushort_t )  [inline]
```

This function deserializes an unsigned short.

Parameters

ushort_t	The variable that will store the unsigned short read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.32 deserialize() [30/40]

```
Cdr& deserialize (
            uint16_t & ushort_t,
            Endianness endianness )  [inline]
```

This function deserializes an unsigned short with a different endianness.

Parameters

ushort_t	The variable that will store the unsigned short read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.33 deserialize() [31/40]

```
Cdr& deserialize (
            uint32_t & ulong_t )  [inline]
```

This function deserializes an unsigned long.

Parameters

ulong_t	The variable that will store the unsigned long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.34 deserialize() [32/40]

```
Cdr& deserialize (
            uint32_t & ulong_t,
            Endianness endianness )  [inline]
```

This function deserializes an unsigned long with a different endianness.

Parameters

ulong_t	The variable that will store the unsigned long read from the buffer..
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.35 deserialize() [33/40]

```
Cdr& deserialize (
            uint64_t & ulonglong_t )  [inline]
```

This function deserializes an unsigned long long.

Parameters

ulonglong_t	The variable that will store the unsigned long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.36 deserialize() [34/40]

```
Cdr& deserialize (
            uint64_t & ulonglong_t,
            Endianness endianness )  [inline]
```

This function deserializes an unsigned long long with a different endianness.

Parameters

ulonglong_t	The variable that will store the unsigned long long read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.37 deserialize() [35/40]

```
Cdr& deserialize (
            uint8_t & octet_t )  [inline]
```

This function deserializes an octet.

Parameters

octet_t	The variable that will store the octet read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.38 deserialize() [36/40]

```
Cdr& deserialize (
            uint8_t & octet_t,
            Endianness endianness )  [inline]
```

This function deserializes an octet with a different endianness.

Parameters

octet_t	The variable that will store the octet read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.39 deserialize() [37/40]

```
Cdr& deserialize (
            wchar_t & wchar )  [inline]
```

This function deserializes a wide-char.

Parameters

wchar	The variable that will store the wide-char read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.40 deserialize() [38/40]

```
Cdr& deserialize (
            wchar_t & wchar,
            Endianness endianness )  [inline]
```

This function deserializes a wide-char with a different endianness.

Parameters

wchar	The variable that will store the wide-char read from the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.41 deserialize() [39/40]

```
Cdr& deserialize (
            wchar_t *& string_t )
```

This function deserializes a wide string.

This function allocates memory to store the wide string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string_t	The pointer that will point to the wide string read from the buffer. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.42 deserialize() [40/40]

```
Cdr& deserialize (
            wchar_t *& string_t,
            Endianness endianness )
```

This function deserializes a wide string with a different endianness.

This function allocates memory to store the wide string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string_t	The pointer that will point to the wide string read from the buffer.
endianness	Endianness that will be used in the deserialization of this value. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.43 deserializeArray() [1/35]

```
Cdr& deserializeArray (
            _T * type_t,
            size_t numElements )  [inline]
```

This function template deserializes an array of non-basic objects.

Parameters

type_t	The variable that will store the array of objects read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.44 deserializeArray() [2/35]

```
Cdr& deserializeArray (
            _T * type_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function template deserializes an array of non-basic objects with a different endianness.

Parameters

type_t	The variable that will store the array of objects read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.45 deserializeArray() [3/35]

```
Cdr& deserializeArray (
            bool * bool_t,
            size_t numElements )
```

This function deserializes an array of booleans.

Parameters

bool_t	The variable that will store the array of booleans read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.46 deserializeArray() [4/35]

```
Cdr& deserializeArray (
            bool * bool_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of booleans with a different endianness.

Parameters

bool_t	The variable that will store the array of booleans read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.47 deserializeArray() [5/35]

```
Cdr& deserializeArray (
            char * char_t,
            size_t numElements )
```

This function deserializes an array of characters.

Parameters

char_t	The variable that will store the array of characters read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.48 deserializeArray() [6/35]

```
Cdr& deserializeArray (
            char * char_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of characters with a different endianness.

Parameters

char_t	The variable that will store the array of characters read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.49 deserializeArray() [7/35]

```
Cdr& deserializeArray (
            double * double_t,
            size_t numElements )
```

This function deserializes an array of doubles.

Parameters

double_t	The variable that will store the array of doubles read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.50 deserializeArray() [8/35]

```
Cdr& deserializeArray (
            double * double_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of doubles with a different endianness.

Parameters

double_t	The variable that will store the array of doubles read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.51 deserializeArray() [9/35]

```
Cdr& deserializeArray (
            float * float_t,
            size_t numElements )
```

This function deserializes an array of floats.

Parameters

float_t	The variable that will store the array of floats read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.52 deserializeArray() [10/35]

```
Cdr& deserializeArray (
            float * float_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of floats with a different endianness.

Parameters

float_t	The variable that will store the array of floats read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.53 deserializeArray() [11/35]

```
Cdr& deserializeArray (
            int16_t * short_t,
            size_t numElements )
```

This function deserializes an array of shorts.

Parameters

short_t	The variable that will store the array of shorts read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.54 deserializeArray() [12/35]

```
Cdr& deserializeArray (
            int16_t * short_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of shorts with a different endianness.

Parameters

short_t	The variable that will store the array of shorts read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.55 deserializeArray() [13/35]

```
Cdr& deserializeArray (
            int32_t * long_t,
            size_t numElements )
```

This function deserializes an array of longs.

Parameters

long_t	The variable that will store the array of longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.56 deserializeArray() [14/35]

```
Cdr& deserializeArray (
            int32_t * long_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of longs with a different endianness.

Parameters

long_t	The variable that will store the array of longs read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.57 deserializeArray() [15/35]

```
Cdr& deserializeArray (
            int64_t * longlong_t,
            size_t numElements )
```

This function deserializes an array of long longs.

Parameters

longlong_t	The variable that will store the array of long longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.58 deserializeArray() [16/35]

```
Cdr& deserializeArray (
            int64_t * longlong_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of long longs with a different endianness.

Parameters

longlong_t	The variable that will store the array of long longs read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.59 deserializeArray() [17/35]

```
Cdr& deserializeArray (
            int8_t * int8,
            size_t numElements )  [inline]
```

This function deserializes an array of int8_t.

Parameters

int8	The variable that will store the array of int8_t read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.60 deserializeArray() [18/35]

```
Cdr& deserializeArray (
            int8_t * int8,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of int8_t with a different endianness.

Parameters

int8	The variable that will store the array of int8_t read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.61 deserializeArray() [19/35]

```
Cdr& deserializeArray (
            long double * ldouble_t,
            size_t numElements )
```

This function deserializes an array of long doubles.

Parameters

ldouble_t	The variable that will store the array of long doubles read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.62 deserializeArray() [20/35]

```
Cdr& deserializeArray (
            long double * ldouble_t,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of long doubles with a different endianness.

Parameters

ldouble_t	The variable that will store the array of long doubles read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.63 deserializeArray() [21/35]

```
Cdr& deserializeArray (
            std::string * string_t,
            size_t numElements )  [inline]
```

This function deserializes an array of strings.

Parameters

string_t	The variable that will store the array of strings read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.64 deserializeArray() [22/35]

```
Cdr& deserializeArray (
            std::string * string_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of strings with a different endianness.

Parameters

string_t	The variable that will store the array of strings read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.65 deserializeArray() [23/35]

```
Cdr& deserializeArray (
            std::vector< _T > * vector_t,
            size_t numElements )  [inline]
```

This function deserializes an array of sequences of objects.

Parameters

vector_t	The variable that will store the array of sequences of objects read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.66 deserializeArray() [24/35]

```
Cdr& deserializeArray (
            std::wstring * string_t,
            size_t numElements )  [inline]
```

This function deserializes an array of wide-strings.

Parameters

string_t	The variable that will store the array of wide-strings read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.67 deserializeArray() [25/35]

```
Cdr& deserializeArray (
            std::wstring * string_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of wide-strings with a different endianness.

Parameters

string_t	The variable that will store the array of wide-strings read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.68 deserializeArray() [26/35]

```
Cdr& deserializeArray (
            uint16_t * ushort_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned shorts.

Parameters

ushort_t	The variable that will store the array of unsigned shorts read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.69 deserializeArray() [27/35]

```
Cdr& deserializeArray (
            uint16_t * ushort_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of unsigned shorts with a different endianness.

Parameters

ushort_t	The variable that will store the array of unsigned shorts read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.70 deserializeArray() [28/35]

```
Cdr& deserializeArray (
            uint32_t * ulong_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned longs.

Parameters

ulong_t	The variable that will store the array of unsigned longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.71 deserializeArray() [29/35]

```
Cdr& deserializeArray (
            uint32_t * ulong_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of unsigned longs with a different endianness.

Parameters

ulong_t	The variable that will store the array of unsigned longs read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.72 deserializeArray() [30/35]

```
Cdr& deserializeArray (
            uint64_t * ulonglong_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned long longs.

Parameters

ulonglong_t	The variable that will store the array of unsigned long longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.73 deserializeArray() [31/35]

```
Cdr& deserializeArray (
            uint64_t * ulonglong_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of unsigned long longs with a different endianness.

Parameters

ulonglong_t	The variable that will store the array of unsigned long longs read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.74 deserializeArray() [32/35]

```
Cdr& deserializeArray (
            uint8_t * octet_t,
            size_t numElements )  [inline]
```

This function deserializes an array of octets.

Parameters

octet_t	The variable that will store the array of octets read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.75 deserializeArray() [33/35]

```
Cdr& deserializeArray (
            uint8_t * octet_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function deserializes an array of octets with a different endianness.

Parameters

octet_t	The variable that will store the array of octets read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.76 deserializeArray() [34/35]

```
Cdr& deserializeArray (
            wchar_t * wchar,
            size_t numElements )
```

This function deserializes an array of wide-chars.

Parameters

wchar	The variable that will store the array of wide-chars read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.77 deserializeArray() [35/35]

```
Cdr& deserializeArray (
            wchar_t * wchar,
            size_t numElements,
            Endianness endianness )
```

This function deserializes an array of wide-chars with a different endianness.

Parameters

wchar	The variable that will store the array of wide-chars read from the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.78 deserializeSequence() [1/2]

```
Cdr& deserializeSequence (
            _T *& sequence_t,
            size_t & numElements ) [inline]
```

This function template deserializes a raw sequence.

This function allocates memory to store the sequence. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

sequence_t	The pointer that will store the sequence read from the buffer.
numElements	This variable return the number of elements of the sequence.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.79 deserializeSequence() [2/2]

```
Cdr& deserializeSequence (
            _T *& sequence_t,
```

```
        size_t & numElements,
        Endianness endianness )  [inline]
```

This function template deserializes a raw sequence with a different endianness.

This function allocates memory to store the sequence. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

sequence_t	The pointer that will store the sequence read from the buffer.
numElements	This variable return the number of elements of the sequence.
endianness	Endianness that will be used in the deserialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.80 endianness()

```
Endianness endianness ( ) const  [inline]
```

This function returns the current endianness used by the CDR type.

Returns

The endianness.

5.3.4.81 getBufferPointer()

```
char* getBufferPointer ( )
```

This function returns the pointer to the current used buffer.

Returns

Pointer to the starting position of the buffer.

5.3.4.82 getCurrentPosition()

```
char* getCurrentPosition ( )
```

This function returns the current position in the CDR stream.

Returns

Pointer to the current position in the buffer.

5.3.4.83 getDDSCdrOptions()

```
uint16_t getDDSCdrOptions ( ) const
```

This function returns the option flags when the CDR type is eprosima::fastcdr::DDS_CDR.

Returns

The option flags.

5.3.4.84 getDDSCdrPlFlag()

```
DDSCdrPlFlag getDDSCdrPlFlag ( ) const
```

This function returns the parameter list flag when the CDR type is eprosima::fastcdr::DDS_CDR.

Returns

The flag that specifies if the content is a parameter list.

5.3.4.85 getSerializedDataLength()

```
size_t getSerializedDataLength ( ) const  [inline]
```

This function returns the length of the serialized data inside the stream.

Returns

The length of the serialized data.

5.3.4.86 getState()

```
state getState ( )
```

This function returns the current state of the CDR serialization process.

Returns

The current state of the CDR serialization process.

5.3.4.87 jump()

```
bool jump (
            size_t numBytes )
```

This function skips a number of bytes in the CDR stream buffer.

Parameters

numBytes	The number of bytes that will be jumped.

Returns

True is returned when it works successfully. Otherwise, false is returned.

5.3.4.88 moveAlignmentForward()

```
bool moveAlignmentForward (
            size_t numBytes )
```

This function moves the alignment forward.

Parameters

numBytes	The number of bytes the alignment should advance.

Returns

True If alignment was moved successfully.

5.3.4.89 operator<<() [1/21]

```
Cdr& operator<< (
            char * string_t )  [inline]
```

This operator serializes a null-terminated c-string.

Parameters

string_t	Pointer to the begining of the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.90 operator<<() [2/21]

```
Cdr& operator<< (
            const _T & type_t )  [inline]
```

This operator template is used to serialize any other non-basic type.

Parameters

type_t	A reference to the object that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.91 operator<<() [3/21]

```
Cdr& operator<< (
```

```
            const bool bool_t )  [inline]
```

This operator serializes a boolean.

Parameters

bool_t	The value of the boolean that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.92 operator<<() [4/21]

```
Cdr& operator<< (
            const char * string_t )  [inline]
```

This operator serializes a null-terminated c-string.

Parameters

string_t	Pointer to the begining of the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.93 operator<<() [5/21]

```
Cdr& operator<< (
            const char char_t )  [inline]
```

This operator serializes a character.

Parameters

char_t	The value of the character that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.94 operator<<() [6/21]

```
Cdr& operator<< (
            const double double_t )  [inline]
```

This operator serializes a double.

Parameters

double_t	The value of the double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.95 operator<<() [7/21]

```
Cdr& operator<< (
            const float float_t )  [inline]
```

This operator serializes a float.

Parameters

float_t	The value of the float that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.96 operator<<() [8/21]

```
Cdr& operator<< (
            const int16_t short_t )  [inline]
```

This operator serializes a short.

Parameters

short_t	The value of the short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.97 operator<<() [9/21]

```
Cdr& operator<< (
            const int32_t long_t )  [inline]
```

This operator serializes a long.

Parameters

long_t	The value of the long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.98 operator<<() [10/21]

```
Cdr& operator<< (
            const int64_t longlong_t )  [inline]
```

This operator serializes a long long.

Parameters

longlong_t	The value of the long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.99 operator<<() [11/21]

```
Cdr& operator<< (
            const int8_t int8 )  [inline]
```

This operator serializes a int8_t.

Parameters

int8	The value of the int8_t that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.100 operator<<() [12/21]

```
Cdr& operator<< (
            const long double ldouble_t )  [inline]
```

This operator serializes a long double.

Parameters

ldouble_t	The value of the long double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.101 operator<<() [13/21]

```
Cdr& operator<< (
            const std::map< _K, _T > & map_t )  [inline]
```

This operator template is used to serialize maps.

Parameters

map_t	The map that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.102 operator<<() [14/21]

```
Cdr& operator<< (
            const std::string & string_t )  [inline]
```

This operator serializes a string.

Parameters

string_t	The string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.103 operator<<() [15/21]

```
Cdr& operator<< (
            const std::vector< _T > & vector_t )  [inline]
```

This operator template is used to serialize sequences.

Parameters

vector_t	The sequence that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.104 operator<<() [16/21]

```
Cdr& operator<< (
            const std::wstring & string_t )  [inline]
```

This operator serializes a wstring.

Parameters

string_t	The wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.105 operator<<() [17/21]

```
Cdr& operator<< (
            const uint16_t ushort_t )  [inline]
```

This operator serializes an unsigned short.

Parameters

ushort_t	The value of the unsigned short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.106 operator<<() [18/21]

```
Cdr& operator<< (
            const uint32_t ulong_t )  [inline]
```

This operator serializes an unsigned long.

Parameters

ulong_t	The value of the unsigned long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.107 operator<<() [19/21]

```
Cdr& operator<< (
            const uint64_t ulonglong_t )  [inline]
```

This operator serializes an unsigned long long.

Parameters

ulonglong_t	The value of the unsigned long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.108 operator<<() [20/21]

```
Cdr& operator<< (
            const uint8_t octet_t )  [inline]
```

This operator serializes an octet.

Parameters

octet_t	The value of the octet that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.109 operator<<() [21/21]

```
Cdr& operator<< (
            const wchar_t wchar )  [inline]
```

This operator serializes a wide-char.

Parameters

wchar	The value of the wide-char that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.110 operator>>() [1/20]

```
Cdr& operator>> (
            _T & type_t )  [inline]
```

This operator template is used to deserialize any other non-basic type.

Parameters

type_t	The variable that will store the object read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.111 operator>>() [2/20]

```
Cdr& operator>> (
            bool & bool_t )  [inline]
```

This operator deserializes a boolean.

Parameters

bool_t	The variable that will store the boolean read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.3.4.112 operator>>() [3/20]

```
Cdr& operator>> (
            char & char_t )  [inline]
```

This operator deserializes a character.

Parameters

char_t	The variable that will store the character read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.113 operator>>() [4/20]

```
Cdr& operator>> (
            char *& string_t )  [inline]
```

This operator deserializes a null-terminated c-string.

Parameters

string_t	The variable that will store the c-string read from the buffer. Please note that a newly allocated string will be returned. The caller should free the returned pointer when appropiate.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.3.4.114 operator>>() [5/20]

```
Cdr& operator>> (
            double & double_t )  [inline]
```

This operator deserializes a double.

Parameters

double_t	The variable that will store the double read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.115 operator>>() [6/20]

```
Cdr& operator>> (
            float & float_t )  [inline]
```

This operator deserializes a float.

Parameters

float_t	The variable that will store the float read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.116 operator>>() [7/20]

```
Cdr& operator>> (
            int16_t & short_t )  [inline]
```

This operator deserializes a short.

Parameters

short_t	The variable that will store the short read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.117 operator>>() [8/20]

```
Cdr& operator>> (
            int32_t & long_t )  [inline]
```

This operator deserializes a long.

Parameters

long↩ _t	The variable that will store the long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.118 operator>>() [9/20]

```
Cdr& operator>> (
            int64_t & longlong_t )  [inline]
```

This operator deserializes a long long.

Parameters

longlong↩ _t	The variable that will store the long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.119 operator>>() [10/20]

```
Cdr& operator>> (
            int8_t & int8 )  [inline]
```

This operator deserializes a int8_t.

Parameters

int8	The variable that will store the int8_t read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.120 operator>>() [11/20]

```
Cdr& operator>> (
            long double & ldouble_t )  [inline]
```

This operator deserializes a long double.

Parameters

ldouble_t	The variable that will store the long double read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.121 operator>>() [12/20]

```
Cdr& operator>> (
            std::map< _K, _T > & map_t )  [inline]
```

This operator template is used to deserialize maps.

Parameters

map_t	The variable that will store the map read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.122 operator>>() [13/20]

```
Cdr& operator>> (
            std::string & string_t )  [inline]
```

This operator deserializes a string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.123 operator>>() [14/20]

```
Cdr& operator>> (
            std::vector< _T > & vector_t )  [inline]
```

This operator template is used to deserialize sequences.

Parameters

vector_t	The variable that will store the sequence read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.124 operator>>() [15/20]

```
Cdr& operator>> (
            std::wstring & string_t )  [inline]
```

This operator deserializes a string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.125 operator>>() [16/20]

```
Cdr& operator>> (
            uint16_t & ushort_t )  [inline]
```

This operator deserializes an unsigned short.

Parameters

ushort↩ _t	The variable that will store the unsigned short read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.126 operator>>() [17/20]

```
Cdr& operator>> (
            uint32_t & ulong_t )  [inline]
```

This operator deserializes an unsigned long.

Parameters

ulong↩ _t	The variable that will store the unsigned long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.127 operator>>() [18/20]

```
Cdr& operator>> (
            uint64_t & ulonglong_t )  [inline]
```

This operator deserializes a unsigned long long.

Parameters

ulonglong_t	The variable that will store the unsigned long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.128 operator>>() [19/20]

```
Cdr& operator>> (
            uint8_t & octet_t )  [inline]
```

This operator deserializes an octet.

Parameters

octet_t	The variable that will store the octet read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.129 operator>>() [20/20]

```
Cdr& operator>> (
            wchar_t & wchar )  [inline]
```

This operator deserializes a wide-char.

Parameters

wchar	The variable that will store the wide-char read from the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.

5.3.4.130 read_encapsulation()

```
Cdr& read_encapsulation ( )
```

This function reads the encapsulation of the CDR stream.

If the CDR stream contains an encapsulation, then this function should be called before starting to deserialize.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.3.4.131 reset()

```
void reset ( )
```

This function resets the current position in the buffer to the beginning.

5.3.4.132 resetAlignment()

```
void resetAlignment ( )  [inline]
```

This function resets the alignment to the current position in the buffer.

5.3.4.133 serialize() [1/40]

```
Cdr& serialize (
            char * string_t )  [inline]
```

This function serializes a string.

Parameters

string_t	The pointer to the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.134 serialize() [2/40]

```
Cdr& serialize (
            const _T & type_t )  [inline]
```

This function template serializes a non-basic object.

Parameters

type_t	The object that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.135 serialize() [3/40]

```
Cdr& serialize (
```

```
            const bool bool_t )
```

This function serializes a boolean.

Parameters

bool_t	The value of the boolean that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.136 serialize() [4/40]

```
Cdr& serialize (
            const bool bool_t,
            Endianness endianness )  [inline]
```

This function serializes a boolean with a different endianness.

Parameters

bool_t	The value of the boolean that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.137 serialize() [5/40]

```
Cdr& serialize (
            const char * string_t )
```

This function serializes a string.

Parameters

string_t	The pointer to the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.138 serialize() [6/40]

```
Cdr& serialize (
            const char * string_t,
            Endianness endianness )
```

This function serializes a string with a different endianness.

Parameters

string_t	The pointer to the string that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.139 serialize() [7/40]

```
Cdr& serialize (
            const char char_t )
```

This function serializes a character.

Parameters

char_t	The value of the character that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.140 serialize() [8/40]

```
Cdr& serialize (
            const char char_t,
            Endianness endianness )  [inline]
```

This function serializes a character with a different endianness.

Parameters

char_t	The value of the character that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.141 serialize() [9/40]

```
Cdr& serialize (
            const double double_t )
```

This function serializes a double.

Parameters

double_t	The value of the double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.142 serialize() [10/40]

```
Cdr& serialize (
            const double double_t,
            Endianness endianness )
```

This function serializes a double with a different endianness.

Parameters

double_t	The value of the double that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.143 serialize() [11/40]

```
Cdr& serialize (
            const float float_t )
```

This function serializes a float.

Parameters

float_t	The value of the float that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.144 serialize() [12/40]

```
Cdr& serialize (
            const float float_t,
            Endianness endianness )
```

This function serializes a float with a different endianness.

Parameters

float_t	The value of the float that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.145 serialize() [13/40]

```
Cdr& serialize (
            const int16_t short_t )
```

This function serializes a short.

Parameters

short_t	The value of the short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.146 serialize() [14/40]

```
Cdr& serialize (
            const int16_t short_t,
            Endianness endianness )
```

This function serializes a short with a different endianness.

Parameters

short_t	The value of the short that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.147 serialize() [15/40]

```
Cdr& serialize (
            const int32_t long_t )
```

This function serializes a long.

Parameters

long_t	The value of the long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.148 serialize() [16/40]

```
Cdr& serialize (
            const int32_t long_t,
            Endianness endianness )
```

This function serializes a long with a different endianness.

Parameters

long_t	The value of the long that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.149 serialize() [17/40]

```
Cdr& serialize (
            const int64_t longlong_t )
```

This function serializes a long long.

Parameters

longlong_t	The value of the long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.150 serialize() [18/40]

```
Cdr& serialize (
            const int64_t longlong_t,
            Endianness endianness )
```

This function serializes a long long with a different endianness.

Parameters

longlong_t	The value of the long long that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.151 serialize() [19/40]

```
Cdr& serialize (
            const int8_t int8 )  [inline]
```

This function serializes an int8_t.

Parameters

int8	The value of the int8_t that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.152 serialize() [20/40]

```
Cdr& serialize (
            const int8_t int8,
            Endianness endianness )  [inline]
```

This function serializes an int8_t with a different endianness.

Parameters

int8	The value of the int8_t that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.153 serialize() [21/40]

```
Cdr& serialize (
            const long double ldouble_t )
```

This function serializes a long double.

Parameters

ldouble_t	The value of the long double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

Note

Due to internal representation differences, WIN32 and ∗NIX like systems are not compatible.

5.3.4.154 serialize() [22/40]

```
Cdr& serialize (
            const long double ldouble_t,
            Endianness endianness )
```

This function serializes a long double with a different endianness.

Parameters

ldouble_t	The value of the long double that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

Note

Due to internal representation differences, WIN32 and ∗NIX like systems are not compatible.

5.3.4.155 serialize() [23/40]

```
Cdr& serialize (
            const std::map< _K, _T > & map_t )  [inline]
```

This function template serializes a map.

Parameters

map_t	The map that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.156 serialize() [24/40]

```
Cdr& serialize (
            const std::string & string_t )  [inline]
```

This function serializes a std::string.

Parameters

string_t	The string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.157 serialize() [25/40]

```
Cdr& serialize (
```

```
            const std::string & string_t,
            Endianness endianness )  [inline]
```

This function serializes a std::string with a different endianness.

Parameters

string_t	The string that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.158 serialize() [26/40]

```
Cdr& serialize (
            const std::vector< _T > & vector_t )  [inline]
```

This function template serializes a sequence.

Parameters

vector_t	The sequence that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.159 serialize() [27/40]

```
Cdr& serialize (
            const std::vector< _T > & vector_t,
            Endianness endianness )  [inline]
```

This function template serializes a sequence with a different endianness.

Parameters

vector_t	The sequence that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.160 serialize() [28/40]

```
Cdr& serialize (
            const std::wstring & string_t )  [inline]
```

This function serializes a std::wstring.

Parameters

string_t	The wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.161 serialize() [29/40]

```
Cdr& serialize (
            const uint16_t ushort_t )  [inline]
```

This function serializes an unsigned short.

Parameters

ushort_t	The value of the unsigned short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.162 serialize() [30/40]

```
Cdr& serialize (
            const uint16_t ushort_t,
            Endianness endianness )  [inline]
```

This function serializes an unsigned short with a different endianness.

Parameters

ushort_t	The value of the unsigned short that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.163 serialize() [31/40]

```
Cdr& serialize (
            const uint32_t ulong_t )  [inline]
```

This function serializes an unsigned long.

Parameters

ulong_t	The value of the unsigned long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.164 serialize() [32/40]

```
Cdr& serialize (
            const uint32_t ulong_t,
            Endianness endianness )  [inline]
```

This function serializes an unsigned long with a different endianness.

Parameters

ulong_t	The value of the unsigned long that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.165 serialize() [33/40]

```
Cdr& serialize (
            const uint64_t ulonglong_t )  [inline]
```

This function serializes an unsigned long long.

Parameters

ulonglong↩ _t	The value of the unsigned long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.166 serialize() [34/40]

```
Cdr& serialize (
            const uint64_t ulonglong_t,
            Endianness endianness )  [inline]
```

This function serializes an unsigned long long with a different endianness.

Parameters

ulonglong↩ _t	The value of the unsigned long long that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.167 serialize() [35/40]

```
Cdr& serialize (
            const uint8_t octet_t )  [inline]
```

This function serializes an octet.

Parameters

octet_t	The value of the octet that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.168 serialize() [36/40]

```
Cdr& serialize (
            const uint8_t octet_t,
            Endianness endianness )  [inline]
```

This function serializes an octet with a different endianness.

Parameters

octet_t	The value of the octet that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.169 serialize() [37/40]

```
Cdr& serialize (
            const wchar_t * string_t )
```

This function serializes a wstring.

Parameters

string_t	The pointer to the wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.170 serialize() [38/40]

```
Cdr& serialize (
            const wchar_t * string_t,
            Endianness endianness )
```

This function serializes a wstring with a different endianness.

Parameters

string_t	The pointer to the wstring that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.171 serialize() [39/40]

```
Cdr& serialize (
            const wchar_t wchar )  [inline]
```

This function serializes a wide-char.

Parameters

wchar	The value of the wide-char that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.172 serialize() [40/40]

```
Cdr& serialize (
            const wchar_t wchar,
            Endianness endianness )  [inline]
```

This function serializes a wide-char with a different endianness.

Parameters

wchar	The value of the wide-char that will be serialized in the buffer.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.173 serialize_encapsulation()

```
Cdr& serialize_encapsulation ( )
```

This function writes the encapsulation of the CDR stream.

If the CDR stream should contain an encapsulation, then this function should be called before starting to serialize.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.174 serializeArray() [1/35]

```
Cdr& serializeArray (
            const _T * type_t,
            size_t numElements )  [inline]
```

This function template serializes an array of non-basic objects.

Parameters

type_t	The array of objects that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.175 serializeArray() [2/35]

```
Cdr& serializeArray (
            const _T * type_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function template serializes an array of non-basic objects with a different endianness.

Parameters

type_t	The array of objects that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.176 serializeArray() [3/35]

```
Cdr& serializeArray (
            const bool * bool_t,
            size_t numElements )
```

This function serializes an array of booleans.

Parameters

bool_t	The array of booleans that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.177 serializeArray() [4/35]

```
Cdr& serializeArray (
            const bool * bool_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of booleans with a different endianness.

Parameters

bool_t	The array of booleans that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.178 serializeArray() [5/35]

```
Cdr& serializeArray (
            const char * char_t,
            size_t numElements )
```

This function serializes an array of characters.

Parameters

char_t	The array of characters that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.179 serializeArray() [6/35]

```
Cdr& serializeArray (
            const char * char_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of characters with a different endianness.

Parameters

char_t	The array of characters that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.180 serializeArray() [7/35]

```
Cdr& serializeArray (
            const double * double_t,
            size_t numElements )
```

This function serializes an array of doubles.

Parameters

double_t	The array of doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.181 serializeArray() [8/35]

```
Cdr& serializeArray (
            const double * double_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of doubles with a different endianness.

Parameters

double_t	The array of doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.182 serializeArray() [9/35]

```
Cdr& serializeArray (
            const float * float_t,
            size_t numElements )
```

This function serializes an array of floats.

Parameters

float_t	The array of floats that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.183 serializeArray() [10/35]

```
Cdr& serializeArray (
            const float * float_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of floats with a different endianness.

Parameters

float_t	The array of floats that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.184 serializeArray() [11/35]

```
Cdr& serializeArray (
            const int16_t * short_t,
            size_t numElements )
```

This function serializes an array of shorts.

Parameters

short_t	The array of shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.185 serializeArray() [12/35]

```
Cdr& serializeArray (
            const int16_t * short_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of shorts with a different endianness.

Parameters

short_t	The array of shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.186 serializeArray() [13/35]

```
Cdr& serializeArray (
            const int32_t * long_t,
            size_t numElements )
```

This function serializes an array of longs.

Parameters

long_t	The array of longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.187 serializeArray() [14/35]

```
Cdr& serializeArray (
            const int32_t * long_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of longs with a different endianness.

Parameters

long_t	The array of longs that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.188 serializeArray() [15/35]

```
Cdr& serializeArray (
            const int64_t * longlong_t,
            size_t numElements )
```

This function serializes an array of long longs.

Parameters

longlong_t	The array of long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.189 serializeArray() [16/35]

```
Cdr& serializeArray (
            const int64_t * longlong_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of long longs with a different endianness.

Parameters

longlong_t	The array of long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.190 serializeArray() [17/35]

```
Cdr& serializeArray (
            const int8_t * int8,
            size_t numElements )  [inline]
```

This function serializes an array of int8_t.

Parameters

int8	The sequence of int8_t that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.191 serializeArray() [18/35]

```
Cdr& serializeArray (
            const int8_t * int8,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of int8_t with a different endianness.

Parameters

int8	The array of int8_t that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.192 serializeArray() [19/35]

```
Cdr& serializeArray (
            const long double * ldouble_t,
            size_t numElements )
```

This function serializes an array of long doubles.

Parameters

ldouble_t	The array of long doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.193 serializeArray() [20/35]

```
Cdr& serializeArray (
            const long double * ldouble_t,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of long doubles with a different endianness.

Parameters

ldouble_t	The array of long doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.194 serializeArray() [21/35]

```
Cdr& serializeArray (
            const std::string * string_t,
            size_t numElements )  [inline]
```

This function serializes an array of strings.

Parameters

string_t	The array of strings that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.195 serializeArray() [22/35]

```
Cdr& serializeArray (
            const std::string * string_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of strings with a different endianness.

Parameters

string_t	The array of strings that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.196 serializeArray() [23/35]

```
Cdr& serializeArray (
            const std::vector< _T > * vector_t,
            size_t numElements )  [inline]
```

This function template serializes an array of sequences of objects.

Parameters

vector_t	The array of sequences of objects that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.197 serializeArray() [24/35]

```
Cdr& serializeArray (
            const std::wstring * string_t,
            size_t numElements )  [inline]
```

This function serializes an array of wide-strings.

Parameters

string_t	The array of wide-strings that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.198 serializeArray() [25/35]

```
Cdr& serializeArray (
            const std::wstring * string_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of wide-strings with a different endianness.

Parameters

string_t	The array of wide-strings that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.199 serializeArray() [26/35]

```
Cdr& serializeArray (
            const uint16_t * ushort_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned shorts.

Parameters

ushort_t	The array of unsigned shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.200 serializeArray() [27/35]

```
Cdr& serializeArray (
            const uint16_t * ushort_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of unsigned shorts with a different endianness.

Parameters

ushort_t	The array of unsigned shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.201 serializeArray() [28/35]

```
Cdr& serializeArray (
            const uint32_t * ulong_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned longs.

Parameters

ulong_t	The array of unsigned longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.202 serializeArray() [29/35]

```
Cdr& serializeArray (
            const uint32_t * ulong_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of unsigned longs with a different endianness.

Parameters

ulong_t	The array of unsigned longs that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.203 serializeArray() [30/35]

```
Cdr& serializeArray (
            const uint64_t * ulonglong_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned long longs.

Parameters

ulonglong_t	The array of unsigned long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.204 serializeArray() [31/35]

```
Cdr& serializeArray (
            const uint64_t * ulonglong_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of unsigned long longs with a different endianness.

Parameters

ulonglong_t	The array of unsigned long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.205 serializeArray() [32/35]

```
Cdr& serializeArray (
            const uint8_t * octet_t,
            size_t numElements )  [inline]
```

This function serializes an array of octets.

Parameters

octet_t	The sequence of octets that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.206 serializeArray() [33/35]

```
Cdr& serializeArray (
            const uint8_t * octet_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function serializes an array of octets with a different endianness.

Parameters

octet_t	The array of octets that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.207 serializeArray() [34/35]

```
Cdr& serializeArray (
            const wchar_t * wchar,
            size_t numElements )
```

This function serializes an array of wide-chars.

Parameters

wchar	The array of wide-chars that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.208 serializeArray() [35/35]

```
Cdr& serializeArray (
            const wchar_t * wchar,
            size_t numElements,
            Endianness endianness )
```

This function serializes an array of wide-chars with a different endianness.

Parameters

wchar	The array of longs that will be serialized in the buffer.
numElements	Number of the elements in the array.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.209 serializeSequence() [1/2]

```
Cdr& serializeSequence (
            const _T * sequence_t,
            size_t numElements )  [inline]
```

This function template serializes a raw sequence.

Parameters

sequence_t	Pointer to the sequence that will be serialized in the buffer.
numElements	The number of elements contained in the sequence.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.210 serializeSequence() [2/2]

```
Cdr& serializeSequence (
            const _T * sequence_t,
            size_t numElements,
            Endianness endianness )  [inline]
```

This function template serializes a raw sequence with a different endianness.

Parameters

sequence_t	Pointer to the sequence that will be serialized in the buffer.
numElements	The number of elements contained in the sequence.
endianness	Endianness that will be used in the serialization of this value.

Returns

Reference to the eprosima::fastcdr::Cdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize a position that exceeds the internal memory size.

5.3.4.211 setDDSCdrOptions()

```
void setDDSCdrOptions (
            uint16_t options )
```

This function sets the option flags when the CDR type is eprosima::fastcdr::DDS_CDR.

Parameters

options	New value for the option flags.

5.3.4.212 setDDSCdrPlFlag()

```
void setDDSCdrPlFlag (
            DDSCdrPlFlag plFlag )
```

This function sets the parameter list flag when the CDR type is eprosima::fastcdr::DDS_CDR.

Parameters

plFlag	New value for the flag that specifies if the content is a parameter list.

5.3.4.213 setState()

```
void setState (
            state & state )
```

This function sets a previous state of the CDR serialization process;.

Parameters

state	Previous state that will be set.

5.3.5 Member Data Documentation

5.3.5.1 DEFAULT_ENDIAN

```
const Endianness DEFAULT_ENDIAN  [static]
```

Default endiness in the system.

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

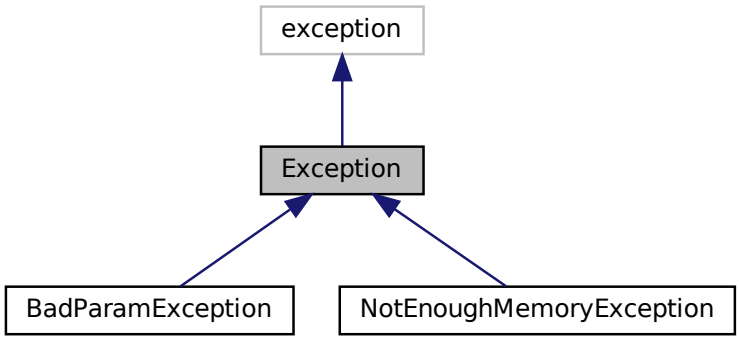
- include/fastcdr/Cdr.h

5.4 Exception Class Reference

This abstract class is used to create exceptions.

`#include <Exception.h>`

Inheritance diagram for Exception:



Public Member Functions

- virtual Cdr_DllAPI ~Exception () noexcept

  Default destructor.
- virtual Cdr_DllAPI void raise () const =0

  This function throws the object as exception.
- virtual const Cdr_DllAPI char * what () const noexcept override

  This function returns the error message.

Protected Member Functions

- Cdr_DllAPI Exception (const char *const &message) noexcept

  Default constructor.
- Cdr_DllAPI Exception (const Exception &ex) noexcept

  Default copy constructor.
- Cdr_DllAPI Exception & operator= (const Exception &ex) noexcept

  Assigment operation.

5.4.1 Detailed Description

This abstract class is used to create exceptions.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 ~Exception()

```
virtual Cdr_DllAPI ~Exception ( )  [virtual], [noexcept]
```

Default destructor.

5.4.2.2 Exception() [1/2]

```
Cdr_DllAPI Exception (
            const char *const & message )  [protected], [noexcept]
```

Default constructor.

Parameters

message	A error message. This message pointer is copied.

5.4.2.3 Exception() [2/2]

```
Cdr_DllAPI Exception (
            const Exception & ex )  [protected], [noexcept]
```

Default copy constructor.

Parameters

ex	Exception that will be copied.

5.4.3 Member Function Documentation

5.4.3.1 operator=()

```
Cdr_DllAPI Exception& operator= (
            const Exception & ex )  [protected], [noexcept]
```

Assigment operation.

Parameters

ex	Exception that will be copied.

5.4.3.2 raise()

```
virtual Cdr_DllAPI void raise ( ) const  [pure virtual]
```

This function throws the object as exception.

Implemented in NotEnoughMemoryException, and BadParamException.

5.4.3.3 what()

```
virtual const Cdr_DllAPI char* what ( ) const  [override], [virtual], [noexcept]
```

This function returns the error message.

Returns

The error message.

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

- include/fastcdr/exceptions/Exception.h

5.5 FastBuffer Class Reference

This class represents a stream of bytes that contains (or will contain) serialized data.

```
#include <FastBuffer.h>
```

Public Types

- typedef _FastBuffer_iterator iterator

Public Member Functions

- FastBuffer ()

  This constructor creates an internal stream and assigns it to the eprosima::fastcdr::FastBuffers object.
- FastBuffer (char *const buffer, const size_t bufferSize)

  This constructor assigns the user's stream of bytes to the eprosima::fastcdr::FastBuffers object.
- FastBuffer (FastBuffer &&fbuffer)

  Move constructor.
- FastBuffer & operator= (FastBuffer &&fbuffer)

  Move assignment.
- virtual ~FastBuffer ()

  Default destructor.
- char * getBuffer () const

  This function returns the stream that the eprosima::fastcdr::FastBuffers uses to serialize data.
- size_t getBufferSize () const

  This function returns the size of the allocated memory of the stream that the eprosima::fastcdr::FastBuffers uses to serialize data.
- iterator begin ()

  This function returns a iterator that points to the begining of the stream.
- iterator end ()

  This function returns a iterator that points to the end of the stream.
- bool reserve (size_t size)

  This function reserves memory for the internal raw buffer.
- bool resize (size_t minSizeInc)

  This function resizes the raw buffer.

5.5.1 Detailed Description

This class represents a stream of bytes that contains (or will contain) serialized data.

This class is used by the serializers to serialize or deserialize using their representation.

5.5.2 Member Typedef Documentation

5.5.2.1 iterator

```
typedef _FastBuffer_iterator iterator
```

5.5.3 Constructor & Destructor Documentation

5.5.3.1 FastBuffer() [1/3]

```
FastBuffer ( )
```

This constructor creates an internal stream and assigns it to the eprosima::fastcdr::FastBuffers object.

The user can obtain this internal stream using the function eprosima::fastcdr::FastBuffers::getBuffer(). Be careful because this internal stream is deleted in the destruction of the eprosima::fastcdr::FastBuffers object.

5.5.3.2 FastBuffer() [2/3]

```
FastBuffer (
            char *const buffer,
            const size_t bufferSize )
```

This constructor assigns the user's stream of bytes to the eprosima::fastcdr::FastBuffers object.

The user's stream will be used to serialize.

Parameters

buffer	The user's buffer that will be used. This buffer is not deallocated in the object's destruction. Cannot be NULL.
bufferSize	The length of user's buffer.

5.5.3.3 FastBuffer() [3/3]

```
FastBuffer (
            FastBuffer && fbuffer )  [inline]
```

Move constructor.

5.5.3.4 ~FastBuffer()

```
virtual ~FastBuffer ( )  [virtual]
```

Default destructor.

5.5.4 Member Function Documentation

5.5.4.1 begin()

```
iterator begin ( )  [inline]
```

This function returns a iterator that points to the begining of the stream.

Returns

The new iterator.

5.5.4.2 end()

```
iterator end ( )  [inline]
```

This function returns a iterator that points to the end of the stream.

Returns

The new iterator.

5.5.4.3 getBuffer()

```
char* getBuffer ( ) const  [inline]
```

This function returns the stream that the eprosima::fastcdr::FastBuffers uses to serialize data.

Returns

The stream used by eprosima::fastcdr::FastBuffers to serialize data.

5.5.4.4 getBufferSize()

```
size_t getBufferSize ( ) const  [inline]
```

This function returns the size of the allocated memory of the stream that the eprosima::fastcdr::FastBuffers uses to serialize data.

Returns

The size of the allocated memory of the stream used by the eprosima::fastcdr::FastBuffers to serialize data.

5.5.4.5 operator=()

```
FastBuffer& operator= (
            FastBuffer && fbuffer )  [inline]
```

Move assignment.

5.5.4.6 reserve()

```
bool reserve (
            size_t size )
```

This function reserves memory for the internal raw buffer.

It will only do so if the buffer is not yet allocated and is not externally set.

Parameters

size	The size of the memory to be allocated.

Returns

True if the allocation suceeded. False if the raw buffer was set externally or is already allocated.

5.5.4.7 resize()

```
bool resize (
            size_t minSizeInc )
```

This function resizes the raw buffer.

It will call the user's defined function for this purpose.

Parameters

minSizeInc	The minimun growth expected of the current raw buffer.

Returns

True if the operation works. False if it does not.

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

- include/fastcdr/FastBuffer.h

5.6 FastCdr Class Reference

This class offers an interface to serialize/deserialize some basic types using a modified CDR protocol inside a eprosima::FastBuffer.

```
#include <FastCdr.h>
```

Classes

- class state

  This class stores the current state of a CDR serialization.

Public Member Functions

- FastCdr (FastBuffer &cdrBuffer)

  This constructor creates a eprosima::fastcdr::FastCdr object that can serialize/deserialize the assigned buffer.
- bool jump (size_t numBytes)

  This function skips a number of bytes in the CDR stream buffer.
- void reset ()

  This function resets the current position in the buffer to the begining.
- char * getCurrentPosition ()

  This function returns the current position in the CDR stream.
- size_t getSerializedDataLength () const

  This function returns the length of the serialized data inside the stream.
- FastCdr::state getState ()

  This function returns the current state of the CDR stream.
- void setState (FastCdr::state &state)

  This function sets a previous state of the CDR stream;.
- FastCdr & operator<< (const uint8_t octet_t)

  This operator serializes an octet.
- FastCdr & operator<< (const char char_t)

  This operator serializes a character.
- FastCdr & operator<< (const int8_t int8)

  This operator serializes a int8_t.
- FastCdr & operator<< (const uint16_t ushort_t)

  This operator serializes an unsigned short.
- FastCdr & operator<< (const int16_t short_t)

  This operator serializes a short.
- FastCdr & operator<< (const uint32_t ulong_t)

  This operator serializes an unsigned long.
- FastCdr & operator<< (const int32_t long_t)

  This operator serializes a long.
- FastCdr & operator<< (const wchar_t wchar)

  This operator serializes a wide-char.
- FastCdr & operator<< (const uint64_t ulonglong_t)

  This operator serializes an unsigned long long.
- FastCdr & operator<< (const int64_t longlong_t)

  This operator serializes a long long.
- FastCdr & operator<< (const float float_t)

  This operator serializes a float.
- FastCdr & operator<< (const double double_t)

  This operator serializes a ldouble.
- FastCdr & operator<< (const long double ldouble_t)

  This operator serializes a long double.
- FastCdr & operator<< (const bool bool_t)

  This operator serializes a boolean.
- FastCdr & operator<< (const char *string_t)

  This operator serializes a null-terminated string.
- FastCdr & operator<< (const wchar_t *string_t)

  This operator serializes a null-terminated wide-string.
- FastCdr & operator<< (const std::string &string_t)

  This operator serializes a string.
- FastCdr & operator<< (const std::wstring &string_t)

This operator serializes a wstring.

- template<class _T >
  FastCdr & operator<< (const std::vector< _T > &vector_t)

  This operator template is used to serialize sequences.
- template<class _T >
  FastCdr & operator<< (const _T &type_t)

  This operator template is used to serialize non-basic types.
- FastCdr & operator>> (uint8_t &octet_t)

  This operator deserializes an octet.
- FastCdr & operator>> (char &char_t)

  This operator deserializes a character.
- FastCdr & operator>> (int8_t &int8)

  This operator deserializes an int8_t.
- FastCdr & operator>> (uint16_t &ushort_t)

  This operator deserializes an unsigned short.
- FastCdr & operator>> (int16_t &short_t)

  This operator deserializes a short.
- FastCdr & operator>> (uint32_t &ulong_t)

  This operator deserializes an unsigned long.
- FastCdr & operator>> (int32_t &long_t)

  This operator deserializes a long.
- FastCdr & operator>> (wchar_t &wchar)

  This operator deserializes a wide-char.
- FastCdr & operator>> (uint64_t &ulonglong_t)

  This operator deserializes an unsigned long long.
- FastCdr & operator>> (int64_t &longlong_t)

  This operator deserializes a long long.
- FastCdr & operator>> (float &float_t)

  This operator deserializes a float.
- FastCdr & operator>> (double &double_t)

  This operator deserializes a double.
- FastCdr & operator>> (long double &ldouble_t)

  This operator deserializes a long double.
- FastCdr & operator>> (bool &bool_t)

  This operator deserializes a boolean.
- FastCdr & operator>> (char *&string_t)

  This operator deserializes a null-terminated c-string.
- FastCdr & operator>> (std::string &string_t)

  This operator deserializes a string.
- FastCdr & operator>> (std::wstring &string_t)

  This operator deserializes a wstring.
- template<class _T >
  FastCdr & operator>> (std::vector< _T > &vector_t)

  This operator template is used to deserialize sequences.
- template<class _T >
  FastCdr & operator>> (_T &type_t)

  This operator template is used to deserialize non-basic types.
- FastCdr & serialize (const uint8_t octet_t)

  This function serializes an octet.
- FastCdr & serialize (const char char_t)

  This function serializes a character.

- FastCdr & serialize (const int8_t int8)

  This function serializes an int8_t.
- FastCdr & serialize (const uint16_t ushort_t)

  This function serializes an unsigned short.
- FastCdr & serialize (const int16_t short_t)

  This function serializes a short.
- FastCdr & serialize (const uint32_t ulong_t)

  This function serializes an unsigned long.
- FastCdr & serialize (const int32_t long_t)

  This function serializes a long.
- FastCdr & serialize (const wchar_t wchar)

  This function serializes a wide-char.
- FastCdr & serialize (const uint64_t ulonglong_t)

  This function serializes an unsigned long long.
- FastCdr & serialize (const int64_t longlong_t)

  This function serializes a long long.
- FastCdr & serialize (const float float_t)

  This function serializes a float.
- FastCdr & serialize (const double double_t)

  This function serializes a double.
- FastCdr & serialize (const long double ldouble_t)

  This function serializes a long double.
- FastCdr & serialize (const bool bool_t)

  This function serializes a boolean.
- FastCdr & serialize (const char *string_t)

  This function serializes a string.
- FastCdr & serialize (const wchar_t *string_t)

  This function serializes a wstring.
- FastCdr & serialize (const std::string &string_t)

  This function serializes a std::string.
- FastCdr & serialize (const std::wstring &string_t)

  This function serializes a std::wstring.
- template<class _T >
  FastCdr & serialize (const std::vector< _T > &vector_t)

  This function template serializes a sequence.
- template<class _T >
  FastCdr & serialize (const _T &type_t)

  This function template serializes a non-basic type.
- FastCdr & serializeArray (const uint8_t *octet_t, size_t numElements)

  This function serializes an array of octets.
- FastCdr & serializeArray (const char *char_t, size_t numElements)

  This function serializes an array of characters.
- FastCdr & serializeArray (const int8_t *int8, size_t numElements)

  This function serializes an array of int8_t.
- FastCdr & serializeArray (const uint16_t *ushort_t, size_t numElements)

  This function serializes an array of unsigned shorts.
- FastCdr & serializeArray (const int16_t *short_t, size_t numElements)

  This function serializes an array of shorts.
- FastCdr & serializeArray (const uint32_t *ulong_t, size_t numElements)

  This function serializes an array of unsigned longs.
- FastCdr & serializeArray (const int32_t *long_t, size_t numElements)

This function serializes an array of longs.

- FastCdr & serializeArray (const wchar_t *wchar, size_t numElements)

  This function serializes an array of wide-chars.
- FastCdr & serializeArray (const uint64_t *ulonglong_t, size_t numElements)

  This function serializes an array of unsigned long longs.
- FastCdr & serializeArray (const int64_t *longlong_t, size_t numElements)

  This function serializes an array of long longs.
- FastCdr & serializeArray (const float *float_t, size_t numElements)

  This function serializes an array of floats.
- FastCdr & serializeArray (const double *double_t, size_t numElements)

  This function serializes an array of doubles.
- FastCdr & serializeArray (const long double *ldouble_t, size_t numElements)

  This function serializes an array of long doubles.
- FastCdr & serializeArray (const bool *bool_t, size_t numElements)

  This function serializes an array of booleans.
- FastCdr & serializeArray (const std::string *string_t, size_t numElements)

  This function serializes an array of strings.
- FastCdr & serializeArray (const std::wstring *string_t, size_t numElements)

  This function serializes an array of wstrings.
- template<class _T >
  FastCdr & serializeArray (const std::vector< _T > *vector_t, size_t numElements)

  This function template serializes an array of sequences.
- template<class _T >
  FastCdr & serializeArray (const _T *type_t, size_t numElements)

  This function template serializes an array of non-basic type objects.
- template<class _T >
  FastCdr & serializeSequence (const _T *sequence_t, size_t numElements)

  This function template serializes a raw sequence.
- FastCdr & deserialize (uint8_t &octet_t)

  This function deserializes an octet.
- FastCdr & deserialize (char &char_t)

  This function deserializes a character.
- FastCdr & deserialize (int8_t &int8)

  This function deserializes an int8_t.
- FastCdr & deserialize (uint16_t &ushort_t)

  This function deserializes an unsigned short.
- FastCdr & deserialize (int16_t &short_t)

  This function deserializes a short.
- FastCdr & deserialize (uint32_t &ulong_t)

  This function deserializes an unsigned long.
- FastCdr & deserialize (int32_t &long_t)

  This function deserializes a long.
- FastCdr & deserialize (wchar_t &wchar)

  This function deserializes a wide-char.
- FastCdr & deserialize (uint64_t &ulonglong_t)

  This function deserializes an unsigned long long.
- FastCdr & deserialize (int64_t &longlong_t)

  This function deserializes a long long.
- FastCdr & deserialize (float &float_t)

  This function deserializes a float.
- FastCdr & deserialize (double &double_t)

This function deserializes a double.

- FastCdr & deserialize (long double &ldouble_t)

  This function deserializes a long double.
- FastCdr & deserialize (bool &bool_t)

  This function deserializes a boolean.
- FastCdr & deserialize (char *&string_t)

  This function deserializes a string.
- FastCdr & deserialize (wchar_t *&string_t)

  This function deserializes a wide string.
- FastCdr & deserialize (std::string &string_t)

  This function deserializes a std::string.
- FastCdr & deserialize (std::wstring &string_t)

  This function deserializes a std::wstring.
- template<class _T >
  FastCdr & deserialize (std::vector< _T > &vector_t)

  This function template deserializes a sequence.
- template<class _T >
  FastCdr & deserialize (_T &type_t)

  This function template deserializes a non-basic type object.
- FastCdr & deserializeArray (uint8_t *octet_t, size_t numElements)

  This function deserializes an array of octets.
- FastCdr & deserializeArray (char *char_t, size_t numElements)

  This function deserializes an array of characters.
- FastCdr & deserializeArray (int8_t *int8, size_t numElements)

  This function deserializes an array of int8_t.
- FastCdr & deserializeArray (uint16_t *ushort_t, size_t numElements)

  This function deserializes an array of unsigned shorts.
- FastCdr & deserializeArray (int16_t *short_t, size_t numElements)

  This function deserializes an array of shorts.
- FastCdr & deserializeArray (uint32_t *ulong_t, size_t numElements)

  This function deserializes an array of unsigned longs.
- FastCdr & deserializeArray (int32_t *long_t, size_t numElements)

  This function deserializes an array of longs.
- FastCdr & deserializeArray (wchar_t *wchar, size_t numElements)

  This function deserializes an array of wide-chars.
- FastCdr & deserializeArray (uint64_t *ulonglong_t, size_t numElements)

  This function deserializes an array of unsigned long longs.
- FastCdr & deserializeArray (int64_t *longlong_t, size_t numElements)

  This function deserializes an array of long longs.
- FastCdr & deserializeArray (float *float_t, size_t numElements)

  This function deserializes an array of floats.
- FastCdr & deserializeArray (double *double_t, size_t numElements)

  This function deserializes an array of doubles.
- FastCdr & deserializeArray (long double *ldouble_t, size_t numElements)

  This function deserializes an array of long doubles.
- FastCdr & deserializeArray (bool *bool_t, size_t numElements)

  This function deserializes an array of booleans.
- FastCdr & deserializeArray (std::string *string_t, size_t numElements)

  This function deserializes an array of strings.
- FastCdr & deserializeArray (std::wstring *string_t, size_t numElements)

  This function deserializes an array of wide-strings.

- template<class _T >
  FastCdr & deserializeArray (std::vector< _T > *vector_t, size_t numElements)

  This function template deserializes an array of sequences.
- template<class _T >
  FastCdr & deserializeArray (_T *type_t, size_t numElements)

  This function template deserializes an array of non-basic type objects.
- template<class _T >
  FastCdr & deserializeSequence (_T *&sequence_t, size_t &numElements)

  This function template deserializes a raw sequence.

5.6.1 Detailed Description

This class offers an interface to serialize/deserialize some basic types using a modified CDR protocol inside a eprosima::FastBuffer.

This modified CDR protocol provides a serialization mechanism much faster than common CDR protocol, because it doesn't use alignment.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 FastCdr()

```
FastCdr (
            FastBuffer & cdrBuffer )
```

This constructor creates a eprosima::fastcdr::FastCdr object that can serialize/deserialize the assigned buffer.

Parameters

cdrBuffer	A reference to the buffer that contains (or will contain) the CDR representation.

5.6.3 Member Function Documentation

5.6.3.1 deserialize() [1/20]

```
FastCdr& deserialize (
            _T & type_t )  [inline]
```

This function template deserializes a non-basic type object.

Parameters

type_t	The variable that will store the object read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.2 deserialize() [2/20]

```
FastCdr& deserialize (
            bool & bool_t )
```

This function deserializes a boolean.

Parameters

bool_t	The variable that will store the boolean read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize in an invalid value.

5.6.3.3 deserialize() [3/20]

```
FastCdr& deserialize (
            char & char_t )  [inline]
```

This function deserializes a character.

Parameters

char_t	The variable that will store the character read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.4 deserialize() [4/20]

```
FastCdr& deserialize (
            char *& string_t )
```

This function deserializes a string.

This function allocates memory to store the string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string_t	The pointer that will point to the string read from the buffer. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.5 deserialize() [5/20]

```
FastCdr& deserialize (
            double & double_t )  [inline]
```

This function deserializes a double.

Parameters

double_t	The variable that will store the double read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.6 deserialize() [6/20]

```
FastCdr& deserialize (
            float & float_t )  [inline]
```

This function deserializes a float.

Parameters

float_t	The variable that will store the float read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.7 deserialize() [7/20]

```
FastCdr& deserialize (
            int16_t & short_t )  [inline]
```

This function deserializes a short.

Parameters

short_t	The variable that will store the short read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.8 deserialize() [8/20]

```
FastCdr& deserialize (
            int32_t & long_t )  [inline]
```

This function deserializes a long.

Parameters

long_t	The variable that will store the long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.9 deserialize() [9/20]

```
FastCdr& deserialize (
            int64_t & longlong_t )  [inline]
```

This function deserializes a long long.

Parameters

longlong_t	The variable that will store the long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.10 deserialize() [10/20]

```
FastCdr& deserialize (
            int8_t & int8 )  [inline]
```

This function deserializes an int8_t.

Parameters

int8	The variable that will store the int8_t read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.11 deserialize() [11/20]

```
FastCdr& deserialize (
            long double & ldouble_t )  [inline]
```

This function deserializes a long double.

Parameters

ldouble_t	The variable that will store the long double read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.12 deserialize() [12/20]

```
FastCdr& deserialize (
            std::string & string_t )  [inline]
```

This function deserializes a std::string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.13 deserialize() [13/20]

```
FastCdr& deserialize (
            std::vector< _T > & vector_t )  [inline]
```

This function template deserializes a sequence.

Parameters

vector_t	The variable that will store the sequence read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.14 deserialize() [14/20]

```
FastCdr& deserialize (
            std::wstring & string_t )  [inline]
```

This function deserializes a std::wstring.

Parameters

string_t	The variable that will store the wstring read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.15 deserialize() [15/20]

```
FastCdr& deserialize (
            uint16_t & ushort_t )  [inline]
```

This function deserializes an unsigned short.

Parameters

ushort_t	The variable that will store the unsigned short read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.16 deserialize() [16/20]

```
FastCdr& deserialize (
            uint32_t & ulong_t )  [inline]
```

This function deserializes an unsigned long.

Parameters

ulong_t	The variable that will store the unsigned long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.17 deserialize() [17/20]

```
FastCdr& deserialize (
            uint64_t & ulonglong_t )  [inline]
```

This function deserializes an unsigned long long.

Parameters

ulonglong_t	The variable that will store the unsigned long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.18 deserialize() [18/20]

```
FastCdr& deserialize (
            uint8_t & octet_t )  [inline]
```

This function deserializes an octet.

Parameters

octet_t	The variable that will store the octet read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.19 deserialize() [19/20]

```
FastCdr& deserialize (
            wchar_t & wchar )  [inline]
```

This function deserializes a wide-char.

Parameters

wchar	The variable that will store the wide-char read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.20 deserialize() [20/20]

```
FastCdr& deserialize (
            wchar_t *& string_t )
```

This function deserializes a wide string.

This function allocates memory to store the wide string. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

string↩ _t	The pointer that will point to the wide string read from the buffer. The user will have to free the allocated memory using free()

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.21 deserializeArray() [1/18]

```
FastCdr& deserializeArray (
            _T * type_t,
            size_t numElements )  [inline]
```

This function template deserializes an array of non-basic type objects.

Parameters

type_t	The variable that will store the array of objects read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.22 deserializeArray() [2/18]

```
FastCdr& deserializeArray (
            bool * bool_t,
            size_t numElements )
```

This function deserializes an array of booleans.

Parameters

bool_t	The variable that will store the array of booleans read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.23 deserializeArray() [3/18]

```
FastCdr& deserializeArray (
            char * char_t,
            size_t numElements )
```

This function deserializes an array of characters.

Parameters

char_t	The variable that will store the array of characters read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.24 deserializeArray() [4/18]

```
FastCdr& deserializeArray (
            double * double_t,
            size_t numElements )
```

This function deserializes an array of doubles.

Parameters

double_t	The variable that will store the array of doubles read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.25 deserializeArray() [5/18]

```
FastCdr& deserializeArray (
            float * float_t,
            size_t numElements )
```

This function deserializes an array of floats.

Parameters

float_t	The variable that will store the array of floats read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.26 deserializeArray() [6/18]

```
FastCdr& deserializeArray (
            int16_t * short_t,
            size_t numElements )
```

This function deserializes an array of shorts.

Parameters

short_t	The variable that will store the array of shorts read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.27 deserializeArray() [7/18]

```
FastCdr& deserializeArray (
            int32_t * long_t,
            size_t numElements )
```

This function deserializes an array of longs.

Parameters

long_t	The variable that will store the array of longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.28 deserializeArray() [8/18]

```
FastCdr& deserializeArray (
            int64_t * longlong_t,
            size_t numElements )
```

This function deserializes an array of long longs.

Parameters

longlong_t	The variable that will store the array of long longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.29 deserializeArray() [9/18]

```
FastCdr& deserializeArray (
            int8_t * int8,
            size_t numElements )  [inline]
```

This function deserializes an array of int8_t.

Parameters

int8	The variable that will store the array of int8_t read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.30 deserializeArray() [10/18]

```
FastCdr& deserializeArray (
            long double * ldouble_t,
            size_t numElements )
```

This function deserializes an array of long doubles.

Parameters

ldouble_t	The variable that will store the array of long doubles read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.31 deserializeArray() [11/18]

```
FastCdr& deserializeArray (
            std::string * string_t,
            size_t numElements )  [inline]
```

This function deserializes an array of strings.

Parameters

string_t	The variable that will store the array of strings read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.32 deserializeArray() [12/18]

```
FastCdr& deserializeArray (
            std::vector< _T > * vector_t,
            size_t numElements )  [inline]
```

This function template deserializes an array of sequences.

Parameters

vector_t	The variable that will store the array of sequences read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.33 deserializeArray() [13/18]

```
FastCdr& deserializeArray (
            std::wstring * string_t,
            size_t numElements )  [inline]
```

This function deserializes an array of wide-strings.

Parameters

string_t	The variable that will store the array of strings read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.34 deserializeArray() [14/18]

```
FastCdr& deserializeArray (
            uint16_t * ushort_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned shorts.

Parameters

ushort_t	The variable that will store the array of unsigned shorts read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.35 deserializeArray() [15/18]

```
FastCdr& deserializeArray (
            uint32_t * ulong_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned longs.

Parameters

ulong_t	The variable that will store the array of unsigned longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.36 deserializeArray() [16/18]

```
FastCdr& deserializeArray (
            uint64_t * ulonglong_t,
            size_t numElements )  [inline]
```

This function deserializes an array of unsigned long longs.

Parameters

ulonglong_t	The variable that will store the array of unsigned long longs read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.37 deserializeArray() [17/18]

```
FastCdr& deserializeArray (
            uint8_t * octet_t,
            size_t numElements )  [inline]
```

This function deserializes an array of octets.

Parameters

octet_t	The variable that will store the array of octets read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.38 deserializeArray() [18/18]

```
FastCdr& deserializeArray (
            wchar_t * wchar,
            size_t numElements )
```

This function deserializes an array of wide-chars.

Parameters

wchar	The variable that will store the array of wide-chars read from the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.39 deserializeSequence()

```
FastCdr& deserializeSequence (
            _T *& sequence_t,
            size_t & numElements )  [inline]
```

This function template deserializes a raw sequence.

This function allocates memory to store the sequence. The user pointer will be set to point this allocated memory. The user will have to free this allocated memory using free()

Parameters

sequence_t	The pointer that will store the sequence read from the buffer.
numElements	This variable return the number of elements of the sequence.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.40 getCurrentPosition()

```
char* getCurrentPosition ( )
```

This function returns the current position in the CDR stream.

Returns

Pointer to the current position in the buffer.

5.6.3.41 getSerializedDataLength()

```
size_t getSerializedDataLength ( ) const  [inline]
```

This function returns the length of the serialized data inside the stream.

Returns

The length of the serialized data.

5.6.3.42 getState()

```
FastCdr::state getState ( )
```

This function returns the current state of the CDR stream.

Returns

The current state of the buffer.

5.6.3.43 jump()

```
bool jump (
            size_t numBytes )
```

This function skips a number of bytes in the CDR stream buffer.

Parameters

numBytes	The number of bytes that will be jumped.

Returns

True is returned when the jump operation works successfully. Otherwise, false is returned.

5.6.3.44 operator<<() [1/20]

```
FastCdr& operator<< (
            const _T & type_t )  [inline]
```

This operator template is used to serialize non-basic types.

Parameters

type_t	The object that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.45 operator<<() [2/20]

```
FastCdr& operator<< (
            const bool bool_t )  [inline]
```

This operator serializes a boolean.

Parameters

bool_t	The value of the boolean that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.46 operator<<() [3/20]

```
FastCdr& operator<< (
            const char * string_t )  [inline]
```

This operator serializes a null-terminated string.

Parameters

string_t	The value of the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.47 operator<<() [4/20]

```
FastCdr& operator<< (
            const char char_t )  [inline]
```

This operator serializes a character.

Parameters

char_t	The value of the character that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.48 operator<<() [5/20]

```
FastCdr& operator<< (
            const double double_t )  [inline]
```

This operator serializes a ldouble.

Parameters

double_t	The value of the double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.49 operator<<() [6/20]

```
FastCdr& operator<< (
            const float float_t )  [inline]
```

This operator serializes a float.

Parameters

float_t	The value of the float that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.50 operator<<() [7/20]

```
FastCdr& operator<< (
            const int16_t short_t )  [inline]
```

This operator serializes a short.

Parameters

short_t	The value of the short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.51 operator<<() [8/20]

```
FastCdr& operator<< (
            const int32_t long_t )  [inline]
```

This operator serializes a long.

Parameters

long_t	The value of the long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.52 operator<<() [9/20]

```
FastCdr& operator<< (
            const int64_t longlong_t )  [inline]
```

This operator serializes a long long.

Parameters

longlong_t	The value of the long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.53 operator<<() [10/20]

```
FastCdr& operator<< (
            const int8_t int8 )  [inline]
```

This operator serializes a int8_t.

Parameters

int8	The value of the int8_t that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.54 operator<<() [11/20]

```
FastCdr& operator<< (
            const long double ldouble_t )  [inline]
```

This operator serializes a long double.

Parameters

ldouble_t	The value of the long double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.55 operator<<() [12/20]

```
FastCdr& operator<< (
            const std::string & string_t )  [inline]
```

This operator serializes a string.

Parameters

string_t	The string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.56 operator<<() [13/20]

```
FastCdr& operator<< (
            const std::vector< _T > & vector_t )  [inline]
```

This operator template is used to serialize sequences.

Parameters

vector_t	The sequence that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.57 operator<<() [14/20]

```
FastCdr& operator<< (
            const std::wstring & string_t )  [inline]
```

This operator serializes a wstring.

Parameters

string_t	The wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.58 operator<<() [15/20]

```
FastCdr& operator<< (
            const uint16_t ushort_t )  [inline]
```

This operator serializes an unsigned short.

Parameters

ushort_t	The value of the unsigned short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.59 operator<<() [16/20]

```
FastCdr& operator<< (
            const uint32_t ulong_t )  [inline]
```

This operator serializes an unsigned long.

Parameters

ulong_t	The value of the unsigned long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.60 operator<<() [17/20]

```
FastCdr& operator<< (
            const uint64_t ulonglong_t )  [inline]
```

This operator serializes an unsigned long long.

Parameters

ulonglong_t	The value of the unsigned long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.61 operator<<() [18/20]

```
FastCdr& operator<< (
            const uint8_t octet_t )  [inline]
```

This operator serializes an octet.

Parameters

octet_t	The value of the octet that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.62 operator<<() [19/20]

```
FastCdr& operator<< (
            const wchar_t * string_t )  [inline]
```

This operator serializes a null-terminated wide-string.

Parameters

string_t	The value of the wide-string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.63 operator<<() [20/20]

```
FastCdr& operator<< (
            const wchar_t wchar )  [inline]
```

This operator serializes a wide-char.

Parameters

wchar	The value of the wide-char that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.64 operator>>() [1/19]

```
FastCdr& operator>> (
            _T & type_t )  [inline]
```

This operator template is used to deserialize non-basic types.

Parameters

type_t	The variable that will store the object read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.65 operator>>() [2/19]

```
FastCdr& operator>> (
            bool & bool_t )  [inline]
```

This operator deserializes a boolean.

Parameters

bool_t	The variable that will store the boolean read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize in an invalid value.

5.6.3.66 operator>>() [3/19]

```
FastCdr& operator>> (
            char & char_t )  [inline]
```

This operator deserializes a character.

Parameters

char_t	The variable that will store the character read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.67 operator>>() [4/19]

```
FastCdr& operator>> (
            char *& string_t )  [inline]
```

This operator deserializes a null-terminated c-string.

Parameters

string_t	The variable that will store the c-string read from the buffer. Please note that a newly allocated string will be returned. The caller should free the returned pointer when appropiate.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize a position that exceeds the internal memory size.
exception::BadParamException	This exception is thrown when trying to deserialize an invalid value.

5.6.3.68 operator>>() [5/19]

```
FastCdr& operator>> (
            double & double_t )  [inline]
```

This operator deserializes a double.

Parameters

double_t	The variable that will store the double read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.69 operator>>() [6/19]

```
FastCdr& operator>> (
            float & float_t )  [inline]
```

This operator deserializes a float.

Parameters

float_t	The variable that will store the float read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.70 operator>>() [7/19]

```
FastCdr& operator>> (
            int16_t & short_t )  [inline]
```

This operator deserializes a short.

Parameters

short_t	The variable that will store the short read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.71 operator>>() [8/19]

```
FastCdr& operator>> (
            int32_t & long_t )  [inline]
```

This operator deserializes a long.

Parameters

long_t	The variable that will store the long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.72 operator>>() [9/19]

```
FastCdr& operator>> (
            int64_t & longlong_t )  [inline]
```

This operator deserializes a long long.

Parameters

longlong_t	The variable that will store the long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.73 operator>>() [10/19]

```
FastCdr& operator>> (
            int8_t & int8 )  [inline]
```

This operator deserializes an int8_t.

Parameters

int8	The variable that will store the int8_t read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.74 operator>>() [11/19]

```
FastCdr& operator>> (
            long double & ldouble_t )  [inline]
```

This operator deserializes a long double.

Parameters

ldouble_t	The variable that will store the long double read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.75 operator>>() [12/19]

```
FastCdr& operator>> (
            std::string & string_t )  [inline]
```

This operator deserializes a string.

Parameters

string_t	The variable that will store the string read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.76 operator>>() [13/19]

```
FastCdr& operator>> (
            std::vector< _T > & vector_t )  [inline]
```

This operator template is used to deserialize sequences.

Parameters

vector_t	The variable that will store the sequence read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.77 operator>>() [14/19]

```
FastCdr& operator>> (
            std::wstring & string_t )  [inline]
```

This operator deserializes a wstring.

Parameters

string_t	The variable that will store the wstring read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.78 operator>>() [15/19]

```
FastCdr& operator>> (
            uint16_t & ushort_t )  [inline]
```

This operator deserializes an unsigned short.

Parameters

ushort_t	The variable that will store the unsigned short read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.79 operator>>() [16/19]

```
FastCdr& operator>> (
            uint32_t & ulong_t )  [inline]
```

This operator deserializes an unsigned long.

Parameters

ulong_t	The variable that will store the unsigned long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.80 operator>>() [17/19]

```
FastCdr& operator>> (
            uint64_t & ulonglong_t )  [inline]
```

This operator deserializes an unsigned long long.

Parameters

ulonglong_t	The variable that will store the unsigned long long read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.81 operator>>() [18/19]

```
FastCdr& operator>> (
            uint8_t & octet_t )  [inline]
```

This operator deserializes an octet.

Parameters

octet_t	The variable that will store the octet read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.82 operator>>() [19/19]

```
FastCdr& operator>> (
            wchar_t & wchar ) [inline]
```

This operator deserializes a wide-char.

Parameters

wchar	The variable that will store the wide-char read from the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to deserialize in a position that exceeds the internal memory size.

5.6.3.83 reset()

```
void reset ( )
```

This function resets the current position in the buffer to the begining.

5.6.3.84 serialize() [1/20]

```
FastCdr& serialize (
            const _T & type_t ) [inline]
```

This function template serializes a non-basic type.

Parameters

type_t	The object that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.85 serialize() [2/20]

```
FastCdr& serialize (
            const bool bool_t )
```

This function serializes a boolean.

Parameters

bool_t	The value of the boolean that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.86 serialize() [3/20]

```
FastCdr& serialize (
            const char * string_t )
```

This function serializes a string.

Parameters

string_t	The pointer to the string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.87 serialize() [4/20]

```
FastCdr& serialize (
            const char char_t )  [inline]
```

This function serializes a character.

Parameters

char_t	The value of the character that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.88 serialize() [5/20]

```
FastCdr& serialize (
            const double double_t )  [inline]
```

This function serializes a double.

Parameters

double_t	The value of the double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.89 serialize() [6/20]

```
FastCdr& serialize (
            const float float_t )  [inline]
```

This function serializes a float.

Parameters

float_t	The value of the float that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.90 serialize() [7/20]

```
FastCdr& serialize (
            const int16_t short_t )  [inline]
```

This function serializes a short.

Parameters

short_t	The value of the short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.91 serialize() [8/20]

```
FastCdr& serialize (
            const int32_t long_t )  [inline]
```

This function serializes a long.

Parameters

long_t	The value of the long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.92 serialize() [9/20]

```
FastCdr& serialize (
            const int64_t longlong_t )  [inline]
```

This function serializes a long long.

Parameters

longlong_t	The value of the long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.93 serialize() [10/20]

```
FastCdr& serialize (
            const int8_t int8 )  [inline]
```

This function serializes an int8_t.

Parameters

int8	The value of the int8_t that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.94 serialize() [11/20]

```
FastCdr& serialize (
            const long double ldouble_t )  [inline]
```

This function serializes a long double.

Parameters

ldouble_t	The value of the long double that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.95 serialize() [12/20]

```
FastCdr& serialize (
            const std::string & string_t )  [inline]
```

This function serializes a std::string.

Parameters

string_t	The string that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.96 serialize() [13/20]

```
FastCdr& serialize (
            const std::vector< _T > & vector_t )  [inline]
```

This function template serializes a sequence.

Parameters

vector_t	The sequence that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.97 serialize() [14/20]

```
FastCdr& serialize (
            const std::wstring & string_t )  [inline]
```

This function serializes a std::wstring.

Parameters

string_t	The wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.98 serialize() [15/20]

```
FastCdr& serialize (
            const uint16_t ushort_t )  [inline]
```

This function serializes an unsigned short.

Parameters

ushort_t	The value of the unsigned short that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.99 serialize() [16/20]

```
FastCdr& serialize (
            const uint32_t ulong_t )  [inline]
```

This function serializes an unsigned long.

Parameters

ulong_t	The value of the unsigned long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.100 serialize() [17/20]

```
FastCdr& serialize (
            const uint64_t ulonglong_t )  [inline]
```

This function serializes an unsigned long long.

Parameters

ulonglong_t	The value of the unsigned long long that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.101 serialize() [18/20]

```
FastCdr& serialize (
            const uint8_t octet_t )  [inline]
```

This function serializes an octet.

Parameters

octet_t	The value of the octet that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.102 serialize() [19/20]

```
FastCdr& serialize (
            const wchar_t * string_t )
```

This function serializes a wstring.

Parameters

string_t	The pointer to the wstring that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.103 serialize() [20/20]

```
FastCdr& serialize (
            const wchar_t wchar )  [inline]
```

This function serializes a wide-char.

Parameters

wchar	The value of the wide-char that will be serialized in the buffer.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.104 serializeArray() [1/18]

```
FastCdr& serializeArray (
            const _T * type_t,
            size_t numElements )  [inline]
```

This function template serializes an array of non-basic type objects.

Parameters

type_t	The array of objects that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.105 serializeArray() [2/18]

```
FastCdr& serializeArray (
            const bool * bool_t,
            size_t numElements )
```

This function serializes an array of booleans.

Parameters

bool_t	The array of booleans that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.106 serializeArray() [3/18]

```
FastCdr& serializeArray (
            const char * char_t,
            size_t numElements )
```

This function serializes an array of characters.

Parameters

char_t	The array of characters that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.107 serializeArray() [4/18]

```
FastCdr& serializeArray (
            const double * double_t,
            size_t numElements )
```

This function serializes an array of doubles.

Parameters

double_t	The array of doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.108 serializeArray() [5/18]

```
FastCdr& serializeArray (
            const float * float_t,
            size_t numElements )
```

This function serializes an array of floats.

Parameters

float_t	The array of floats that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.109 serializeArray() [6/18]

```
FastCdr& serializeArray (
            const int16_t * short_t,
            size_t numElements )
```

This function serializes an array of shorts.

Parameters

short_t	The array of shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.110 serializeArray() [7/18]

```
FastCdr& serializeArray (
            const int32_t * long_t,
            size_t numElements )
```

This function serializes an array of longs.

Parameters

long_t	The array of longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.111 serializeArray() [8/18]

```
FastCdr& serializeArray (
            const int64_t * longlong_t,
            size_t numElements )
```

This function serializes an array of long longs.

Parameters

longlong_t	The array of long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.112 serializeArray() [9/18]

```
FastCdr& serializeArray (
            const int8_t * int8,
            size_t numElements )  [inline]
```

This function serializes an array of int8_t.

Parameters

int8	The sequence of int8_t that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.113 serializeArray() [10/18]

```
FastCdr& serializeArray (
            const long double * ldouble_t,
            size_t numElements )
```

This function serializes an array of long doubles.

Parameters

ldouble_t	The array of long doubles that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.114 serializeArray() [11/18]

```
FastCdr& serializeArray (
            const std::string * string_t,
            size_t numElements )  [inline]
```

This function serializes an array of strings.

Parameters

string_t	The array of strings that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.115 serializeArray() [12/18]

```
FastCdr& serializeArray (
            const std::vector< _T > * vector_t,
            size_t numElements )  [inline]
```

This function template serializes an array of sequences.

Parameters

vector_t	The array of sequences that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.116 serializeArray() [13/18]

```
FastCdr& serializeArray (
            const std::wstring * string_t,
            size_t numElements )  [inline]
```

This function serializes an array of wstrings.

Parameters

string_t	The array of wstrings that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.117 serializeArray() [14/18]

```
FastCdr& serializeArray (
            const uint16_t * ushort_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned shorts.

Parameters

ushort_t	The array of unsigned shorts that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.118 serializeArray() [15/18]

```
FastCdr& serializeArray (
            const uint32_t * ulong_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned longs.

Parameters

ulong_t	The array of unsigned longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.119 serializeArray() [16/18]

```
FastCdr& serializeArray (
            const uint64_t * ulonglong_t,
            size_t numElements )  [inline]
```

This function serializes an array of unsigned long longs.

Parameters

ulonglong_t	The array of unsigned long longs that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.120 serializeArray() [17/18]

```
FastCdr& serializeArray (
            const uint8_t * octet_t,
            size_t numElements )  [inline]
```

This function serializes an array of octets.

Parameters

octet_t	The sequence of octets that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.121 serializeArray() [18/18]

```
FastCdr& serializeArray (
            const wchar_t * wchar,
            size_t numElements )
```

This function serializes an array of wide-chars.

Parameters

wchar	The array of wide-chars that will be serialized in the buffer.
numElements	Number of the elements in the array.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.122 serializeSequence()

```
FastCdr& serializeSequence (
            const _T * sequence_t,
            size_t numElements )  [inline]
```

This function template serializes a raw sequence.

Parameters

sequence_t	Pointer to the sequence that will be serialized in the buffer.
numElements	The number of elements contained in the sequence.

Returns

Reference to the eprosima::fastcdr::FastCdr object.

Exceptions

exception::NotEnoughMemoryException	This exception is thrown when trying to serialize in a position that exceeds the internal memory size.

5.6.3.123 setState()

```
void setState (
            FastCdr::state & state )
```

This function sets a previous state of the CDR stream;.

Parameters

state	Previous state that will be set again.

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

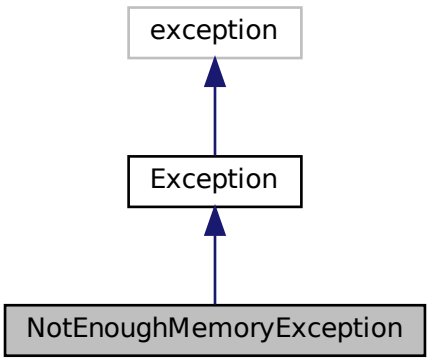
- include/fastcdr/FastCdr.h

5.7 NotEnoughMemoryException Class Reference

This class is thrown as an exception when the buffer's internal memory reachs its size limit.

`#include <NotEnoughMemoryException.h>`

Inheritance diagram for NotEnoughMemoryException:



Public Member Functions

- Cdr_DllAPI NotEnoughMemoryException (const char ∗const &message) noexcept

  Default constructor.
- Cdr_DllAPI NotEnoughMemoryException (const NotEnoughMemoryException &ex) noexcept

  Default copy constructor.
- Cdr_DllAPI NotEnoughMemoryException & operator= (const NotEnoughMemoryException &ex) noexcept

  Assigment operation.
- virtual Cdr_DllAPI ∼NotEnoughMemoryException () noexcept

  Default constructor.
- virtual Cdr_DllAPI void raise () const override

  This function throws the object as exception.

Static Public Attributes

- static const Cdr_DllAPI char ∗const NOT_ENOUGH_MEMORY_MESSAGE_DEFAULT

  Default message used in the library.

Additional Inherited Members

5.7.1 Detailed Description

This class is thrown as an exception when the buffer's internal memory reachs its size limit.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 NotEnoughMemoryException() [1/2]

```
Cdr_DllAPI NotEnoughMemoryException (
            const char *const & message )  [noexcept]
```

Default constructor.

Parameters

message	A error message. This message pointer is copied.

5.7.2.2 NotEnoughMemoryException() [2/2]

```
Cdr_DllAPI NotEnoughMemoryException (
            const NotEnoughMemoryException & ex )  [noexcept]
```

Default copy constructor.

Parameters

ex	NotEnoughMemoryException that will be copied.

5.7.2.3 ~NotEnoughMemoryException()

```
virtual Cdr_DllAPI ~NotEnoughMemoryException ( )  [virtual], [noexcept]
```

Default constructor.

5.7.3 Member Function Documentation

5.7.3.1 operator=()

```
Cdr_DllAPI NotEnoughMemoryException& operator= (
            const NotEnoughMemoryException & ex )  [noexcept]
```

Assigment operation.

Parameters

ex	NotEnoughMemoryException that will be copied.

5.7.3.2 raise()

```
virtual Cdr_DllAPI void raise ( ) const  [override], [virtual]
```

This function throws the object as exception.

Implements Exception.

5.7.4 Member Data Documentation

5.7.4.1 NOT_ENOUGH_MEMORY_MESSAGE_DEFAULT

```
const Cdr_DllAPI char* const NOT_ENOUGH_MEMORY_MESSAGE_DEFAULT  [static]
```

Default message used in the library.

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

- include/fastcdr/exceptions/NotEnoughMemoryException.h

5.8 FastCdr::state Class Reference

This class stores the current state of a CDR serialization.

```
#include <FastCdr.h>
```

Public Member Functions

- state (const FastCdr &fastcdr)
  Default constructor.
- state (const state &)
  Copy constructor.

Friends

- class FastCdr

5.8.1 Detailed Description

This class stores the current state of a CDR serialization.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 state() [1/2]

```
state (
            const FastCdr & fastcdr )
```

Default constructor.

5.8.2.2 state() [2/2]

```
state (
            const state &  )
```

Copy constructor.

5.8.3 Friends And Related Function Documentation

5.8.3.1 FastCdr

```
friend class FastCdr  [friend]
```

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

- include/fastcdr/FastCdr.h

5.9 Cdr::state Class Reference

This class stores the current state of a CDR serialization.

```
#include <Cdr.h>
```

Public Member Functions

- state (const Cdr &cdr)
  Default constructor.
- state (const state &)
  Copy constructor.

Friends

- class Cdr

5.9.1 Detailed Description

This class stores the current state of a CDR serialization.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 state() [1/2]

```
state (
            const Cdr & cdr )
```

Default constructor.

5.9.2.2 state() [2/2]

```
state (
            const state &  )
```

Copy constructor.

5.9.3 Friends And Related Function Documentation

5.9.3.1 Cdr

```
friend class Cdr  [friend]
```

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

- include/fastcdr/Cdr.h

Index