

**BSTRERROR(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<a href="#">1</a>	<a href="#">SYNOPSIS</a>	<a href="#">1</a>
<a href="#">2</a>	<a href="#">DESCRIPTION</a>	<a href="#">2</a>
<a href="#">3</a>	<a href="#">RETURN VALUE</a>	<a href="#">3</a>
<a href="#">4</a>	<a href="#">ERRORS</a>	<a href="#">4</a>
<a href="#">5</a>	<a href="#">BUGS</a>	<a href="#">5</a>
<a href="#">6</a>	<a href="#">SEE ALSO</a>	<a href="#">6</a>
<a href="#">7</a>	<a href="#">COPYING</a>	<a href="#">7</a>

## Chapter 1

# SYNOPSIS

```
#include <ubf.h>
```

```
char * Bsterror (int err);
```

Link with *-lubf -lnstd -lm -lpthread*

## Chapter 2

# DESCRIPTION

Function returns string description of given error code. Enduro/X also logs additional information for last error state, and that information is returned in string format together with description of *err*. Typically when error occurs, then programmer calls **Bsterror(Berror)**, thus function will closely describe what was happened. However if programmer calls **Bsterror()** with other error code, then message will contain the description of *err* plus last UBF call's logged error info. Then message and **Berror** infos are stored in thread local storage (TLS).

## Chapter 3

# RETURN VALUE

String containing formatted error code description plus last call's error logged info. For example:

```
...
/* call the service now! */
char username[100];
BFLDLLEN len=sizeof(username);
if (SUCCEED!=Bget(p_ub, USER_NAME, 0, buf, &len))
{
    ret=FAIL;
    /* print the result */
    fprintf(stderr, "%s\n", Bstrerror(Berror));
}
...
```

## Chapter 4

# ERRORS

Function does not return any errors.

## Chapter 5

# BUGS

Report bugs to [madars.vitolins@gmail.com](mailto:madars.vitolins@gmail.com)



## Chapter 6

## SEE ALSO

**Berrno(3)** \***B\_error(3)**

## **Chapter 7**

# **COPYING**

© ATR Baltic, SIA