

**CBVCHG(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="#">EXAMPLE</a>	<a href="#">5</a>
<a href="#">6</a>	<a href="#">BUGS</a>	<a href="#">6</a>
<a href="#">7</a>	<a href="#">SEE ALSO</a>	<a href="#">7</a>
<a href="#">8</a>	<a href="#">COPYING</a>	<a href="#">8</a>

## Chapter 1

# SYNOPSIS

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

```
int CBvchg(char *cstruct, char *view, char *cname, BFLDOCC occ, char *buf, BFLDLEN len, int usrtype);
```

```
Link with -lubf -lnstd -lm -lpthread
```

---

## Chapter 2

# DESCRIPTION

Function is used to set field data dynamically to VIEW buffer field. *cstruct* is pointer to C structure instance (object), *view* is name of the VIEW (must match the instance of the object), *occ* is field array occurrence, if field is not an array or first array element must be set, then use *occ=0*. *buf* is buffer where to take user data from. *len* is used for indicating the length of the bytes of the target buffer (*buf*) size in bytes. If field size in VIEW is shorter than data putting to, then **BNOSPACE** error is returned. *len* is used only for **BFLD\_CARRAY** fields. *usrtype* is user type indicating the data type of the *buf*, if types are different, the automatic data conversion is performed. And VIEW matched data is installed in the *cstruct* field offset. If current array *C\_* (count) field is less than occurrences, then *C\_* count field is adjusted accordingly to have the occurrence included in the count range.

## Chapter 3

# RETURN VALUE

On success, **CBvchg()** return zero; on error, -1 is returned, with **Berror** set to indicate the error.

## Chapter 4

# ERRORS

Note that **Bstrerror()** returns generic error message plus custom message with debug info from last function call, stored in thread local storage (TLS).

**BEINVAL** *view*, *cstruct* or *buf* is NULL. *view* or *cname* is empty string (first byte 0x00). Or occurrence is out of bounds of the array declaration.

**BBADVIEW** invalid view name (not found in view file).

**BNOCNAME** field name (*cname*) not found in view structure.

**BNOSPACE** No space in view field to install data (data larger than field size).

**BMALLOC** Failed to allocate type conversion buffer.

---

## Chapter 5

# EXAMPLE

See `atmitest/test040_typedview/vaccset.c` for sample code.



## Chapter 6

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)

# Chapter 7

## SEE ALSO

**CBvget(3) Bvnext(3) Bvsetoccur(3)**

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd