

**NAME**

**bswap16**, **bswap32**, **bswap64**, **be16toh**, **be32toh**, **be64toh**, **htobe16**, **htobe32**, **htobe64**, **htole16**, **htole32**, **htole64**, **le16toh**, **le32toh**, **le64toh**, **be16enc**, **be16dec**, **be32enc**, **be32dec**, **be64enc**, **be64dec**, **le16enc**, **le16dec**, **le32enc**, **le32dec**, **le64enc**, **le64dec** - byte order operations

**SYNOPSIS**

```
#include <sys/endian.h>
```

```
uint16_t
```

```
bswap16(uint16_t int16);
```

```
uint32_t
```

```
bswap32(uint32_t int32);
```

```
uint64_t
```

```
bswap64(uint64_t int64);
```

```
uint16_t
```

```
be16toh(uint16_t big16);
```

```
uint32_t
```

```
be32toh(uint32_t big32);
```

```
uint64_t
```

```
be64toh(uint64_t big64);
```

```
uint16_t
```

```
htobe16(uint16_t host16);
```

```
uint32_t
```

```
htobe32(uint32_t host32);
```

```
uint64_t
```

```
htobe64(uint64_t host64);
```

```
uint16_t
```

```
htole16(uint16_t host16);
```

```
uint32_t
```

```
htole32(uint32_t host32);
```

```
uint64_t  
htole64(uint64_t host64);
```

```
uint16_t  
le16toh(uint16_t little16);
```

```
uint32_t  
le32toh(uint32_t little32);
```

```
uint64_t  
le64toh(uint64_t little64);
```

```
uint16_t  
be16dec(const void *);
```

```
uint32_t  
be32dec(const void *);
```

```
uint64_t  
be64dec(const void *);
```

```
uint16_t  
le16dec(const void *);
```

```
uint32_t  
le32dec(const void *);
```

```
uint64_t  
le64dec(const void *);
```

```
void  
be16enc(void *, uint16_t);
```

```
void  
be32enc(void *, uint32_t);
```

```
void  
be64enc(void *, uint64_t);
```

```
void
```

```
le16enc(void *, uint16_t);
```

*void*

```
le32enc(void *, uint32_t);
```

*void*

```
le64enc(void *, uint64_t);
```

## DESCRIPTION

The **bswap16()**, **bswap32()**, and **bswap64()** functions return a byte order swapped integer. On big endian systems, the number is converted to little endian byte order. On little endian systems, the number is converted to big endian byte order.

The **be16toh()**, **be32toh()**, and **be64toh()** functions return a big endian byte ordered integer converted to the system's native byte order. The return value will be the same as the argument on big endian systems.

The **le16toh()**, **le32toh()**, and **le64toh()** functions return a little endian byte ordered integer converted to the system's native byte order. The return value will be the same as the argument on little endian systems.

The **htobe16()**, **htobe32()**, and **htobe64()** functions return an integer in the system's native byte order converted to big endian byte order. The return value will be the same as the argument on big endian systems.

The **htole16()**, **htole32()**, and **htole64()** functions return an integer in the system's native byte order converted to little endian byte order. The return value will be the same as the argument on little endian systems.

The **be16enc()**, **be16dec()**, **be32enc()**, **be32dec()**, **be64enc()**, **be64dec()**, **le16enc()**, **le16dec()**, **le32enc()**, **le32dec()**, **le64enc()**, and **le64dec()** functions encode and decode integers to/from byte strings on any alignment in big/little endian format.

## SEE ALSO

[byteorder\(3\)](#)

## HISTORY

The **hto\*()** and **\*toh()** functions first appeared in FreeBSD 5.0, and were originally developed by the NetBSD project.

The encode/decode functions first appeared in FreeBSD 5.1.