Passing `&state` to one of the restartable functions causes the conversion to begin in the initial conversion state. Once an `mbstate_t` variable has been altered by one of these functions, it should not be used to convert a different multibyte character sequence, nor should it be used to perform a conversion in the opposite direction. Attempting to perform either action causes undefined behavior. Using the variable after a change in the `LC_CTYPE` category of a locale also causes undefined behavior.

Single-Byte/Wide-Character Conversion Functions

```
wint_t btowc(int c);
int wctob(wint_t c);
```

The functions in this group convert single-byte characters to wide characters and vice versa.

btowc The `btowc` function returns `WEOF` if `c` is equal to `EOF` or if `c` (when cast to `unsigned char`) isn't a valid single-byte character in the initial shift state. Otherwise, `btowc` returns the wide-character representation of `c`.

wctob The `wctob` function is the opposite of `btowc`. It returns `EOF` if `c` doesn't correspond to one multibyte character in the initial shift state. Otherwise, it returns the single-byte representation of `c`.

Conversion-State Functions

```
int mbsinit(const mbstate_t *ps);
```

mbsinit This group consists of a single function, `mbsinit`, which returns a nonzero value if `ps` is a null pointer or it points to an `mbstate_t` variable that describes an initial conversion state.

Restartable Multibyte/Wide-Character Conversion Functions

```
size_t mbrlen(const char * restrict s, size_t n,
              mbstate_t * restrict ps);
size_t mbrtowc(wchar_t * restrict pwc,
               const char * restrict s, size_t n,
               mbstate_t * restrict ps);
size_t wcrtomb(char * restrict s, wchar_t wc,
               mbstate_t * restrict ps);
```

The functions in this group are restartable versions of the `mblen`, `mbtowc`, and `wctomb` functions, which belong to `<stdlib.h>` and are discussed in Section 25.2. The newer `mbrlen`, `mbrtowc`, and `wcrtomb` functions differ from their `<stdlib.h>` counterparts in several ways: