mbstate_t variable used during the call is left in the initial conversion state. wcrtomb returns the number of bytes that it stores, including shift sequences. If wc isn't a valid wide character, the function returns –1 and stores EILSEQ in errno.

Restartable Multibyte/Wide-String Conversion Functions

```
size_t mbsrtowcs(wchar_t * restrict dst,
                 const char ** restrict src,
                 size_t len,
                 mbstate_t * restrict ps);
size_t wcsrtombs(char * restrict dst,
                 const wchar_t ** restrict src,
                 size_t len,
                 mbstate_t * restrict ps);
```

mbsrtowcs
wcsrtombs

The mbsrtowcs and wcsrtombs functions are restartable versions of mbstowcs and wcstombs, which belong to <stdlib.h> and are discussed in Section 25.2. mbsrtowcs and wcsrtombs are the same as their <stdlib.h> counterparts, except for the following differences:

- mbsrtowcs and wcsrtombs have an additional parameter named ps. When one of these functions is called, the corresponding argument should point to a variable of type mbstate_t; the function will store the state of the conversion in this variable. If the argument corresponding to ps is a null pointer, the function will use an internal variable to store the conversion state. (At the beginning of program execution, this variable is set to the initial conversion state.) Both functions update the state as the conversion proceeds. If the conversion stops because a null character is reached, the mbstate_t variable will be left in the initial conversion state.
- The src parameter, which represents the array containing characters to be converted (the source array), is a pointer to a pointer for mbsrtowcs and wcsrtombs. (In the older mbstowcs and wcstombs functions, the corresponding parameter was simply a pointer.) This change allows mbsrtowcs and wcsrtombs to keep track of where the conversion stopped. The pointer to which src points is set to null if the conversion stopped because a null character was reached. Otherwise, this pointer is set to point just past the last source character converted.
- The dst parameter may be a null pointer, in which case the converted characters aren't stored and the pointer to which src points isn't modified.
- When either function encounters an invalid character in the source array, it stores EILSEQ in errno (in addition to returning –1, as the older functions do).