

The functions in this group are wide-character versions of the miscellaneous string functions found in <string.h> and described in Section 23.6. The <wchar.h> functions have arguments of type `wchar_t *` instead of `char *`, but their behavior is mostly the same as the <string.h> functions. Table 25.18 shows the correspondence between the <string.h> functions and their wide-character counterparts.

Table 25.18
Miscellaneous Wide-String
Functions and Their
<string.h> Equivalents

<i><wchar.h> Function</i>	<i><string.h> Equivalent</i>
<code>wcslen</code>	<code>strlen</code>
<code>wmemset</code>	<code>memset</code>

Wide-Character Time-Conversion Functions

```
size_t wcsftime(wchar_t * restrict s, size_t maxsize,
                const wchar_t * restrict format,
                const struct tm * restrict timeptr);
```

wcsftime The `wcsftime` function is the wide-character version of `strftime`, which belongs to the <time.h> header and is described in Section 26.3.

Extended Multibyte/Wide-Character Conversion Utilities

We'll now examine <wchar.h> functions that perform conversions between multibyte characters and wide characters. Five of these functions (`mbstrlen`, `mbrtowc`, `wcrtomb`, `mbsrtowcs`, and `wcsrtombs`) correspond to the multibyte/wide-character and multibyte/wide-string conversion functions declared in <stdlib.h>. The <wchar.h> functions have an additional parameter, a pointer to a variable of type `mbstate_t`. This variable keeps track of the state of the conversion of a multibyte character sequence to a wide-character sequence (or vice versa), based on the current locale. As a result, the <wchar.h> functions are “restartable”; by passing a pointer to an `mbstate_t` variable modified by a previous function call, we can “restart” the function using the conversion state from that call. One advantage of this arrangement is that it allows two functions to share the same conversion state. For example, calls of `mbrtowc` and `mbsrtowcs` that are used to process a single multibyte character string could share an `mbstate_t` variable.

The conversion state stored in an `mbstate_t` variable consists of the current shift state plus the current position within a multibyte character. Setting the bytes of an `mbstate_t` variable to zero puts it in the initial conversion state, signifying that no multibyte character is yet in progress and that the initial shift state is in effect:

```
mbstate_t state;
...
memset(&state, '\0', sizeof(state));
```