

UTF-8 Conversion

On this page:

- [Converting to UTF-8](#)
- [Converting from UTF-8](#)

Converting to UTF-8

If you have [installed a string converter](#), the Ice run time calls the converter's `toUTF8` function whenever it needs to convert a native string into UTF-8 representation for transmission. The `sourceStart` and `sourceEnd` pointers point at the first byte and one-beyond-the-last byte of the source string, respectively. The implementation of `toUTF8` must return a pointer to the first unused byte following the converted string.

Your implementation of `toUTF8` must allocate the returned string by calling the `getMoreBytes` member function of the `UTF8Buffer` class that is passed as the third argument. (`getMoreBytes` throws a `std::bad_alloc` if it cannot allocate enough memory.) The `firstUnused` parameter must point at the first unused byte of the allocated memory region. You can make several calls to `getMoreBytes` to incrementally allocate memory for the converted string. If you do, `getMoreBytes` may relocate the buffer in memory. (If it does, it copies the part of the string that was converted so far into the new memory region.) The function returns a pointer to the first unused byte of the (possibly relocated) memory.

Conversion with `toUTF8` during marshaling can fail because `getMoreBytes` can cause the message size to exceed `Ice.MessageSizeMax`. In this case, you should let the `MemoryLimitException` thrown by `getMoreBytes` propagate to the caller.

Conversion can also fail because the encoding of the source string is internally incorrect. In that case, you should throw a `StringConversionFailed` exception from `toUTF8`.

After it has marshaled the returned string into an internal marshaling buffer, the Ice run time deallocates the string.

Converting from UTF-8

During unmarshaling, the Ice run time calls the `fromUTF8` member function on the corresponding string converter. The function converts a UTF-8 string into its native form as a `std::string`. (The string into which the function must place the converted characters is passed to `fromUTF8` as the `target` parameter.)

See Also

- [Installing String Converters](#)