

Table 15.2: Interoperability between Fortran and C types

| Fortran type | Named constant from the ISO_C_BINDING module (kind type parameter if value is positive) | C type |
|--------------|--|------------------------------|
| INTEGER | C_INT | int |
| | C_SHORT | short int |
| | C_LONG | long int |
| | C_LONG_LONG | long long int |
| | C_SIGNED_CHAR | signed char unsigned char |
| | C_SIZE_T | size_t |
| | C_INT8_T | int8_t |
| | C_INT16_T | int16_t |
| | C_INT32_T | int32_t |
| | C_INT64_T | int64_t |
| | C_INT_LEAST8_T | int_least8_t |
| | C_INT_LEAST16_T | int_least16_t |
| | C_INT_LEAST32_T | int_least32_t |
| | C_INT_LEAST64_T | int_least64_t |
| | C_INT_FAST8_T | int_fast8_t |
| | C_INT_FAST16_T | int_fast16_t |
| | C_INT_FAST32_T | int_fast32_t |
| | C_INT_FAST64_T | int_fast64_t |
| | C_INTMAX_T | intmax_t |
| | C_INTPTR_T | intptr_t |
| REAL | C_FLOAT | float |
| | C_DOUBLE | double |
| | C_LONG_DOUBLE | long double |
| COMPLEX | C_FLOAT_COMPLEX | float _Complex |
| | C_DOUBLE_COMPLEX | double _Complex |
| | C_LONG_DOUBLE_COMPLEX | long double _Complex |
| LOGICAL | C_BOOL | _Bool |
| CHARACTER | C_CHAR | char |

NOTE 15.7

For example, the type integer with a [kind type parameter](#) of C_SHORT is [interoperable](#) with the C type short or any C type derived (via typedef) from short.

NOTE 15.8

ISO/IEC 9899:1999 specifies that the representations for nonnegative signed integers are the same as the corresponding values of unsigned integers. Because Fortran does not provide direct support for unsigned kinds of integers, the ISO_C_BINDING module does not make accessible [named constants](#) for their [kind type parameter](#) values. A user can use the signed kinds of integers to interoperate with the unsigned types and all their qualified versions as well. This has the potentially surprising side effect that the C type unsigned char is [interoperable](#) with the type integer with a [kind type parameter](#) of C_SIGNED_CHAR.