

OS/390



# C/C++ Reference Summary



OS/390



# C/C++ Reference Summary

#### **Fourth Edition, September 1998**

This edition applies to Version 2 Release 6 of OS/390 C/C++ (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions or other updated documentation. Make sure that you use the correct edition for the level of the program listed above. Also, ensure that you apply all necessary PTFs for the program.

Technical changes in the text since the last release of this book are indicated by a vertical line (|) to the left of the change.

Order publications through your IBM representative or the IBM branch office serving your location. Publications are not stocked at the address below. Note that the OS/390 C/C++ publications are available through the OS/390 Library page at: <http://www.s390.ibm.com/os390/bkserv>.

IBM welcomes your comments. You can send your comments electronically to the network ID listed below. Be sure to include your entire network address if you wish a reply.

Internet: [torrcf@ca.ibm.com](mailto:torrcf@ca.ibm.com)  
IBMLink: [toribm\(torrcf\)](#)  
IBM/PROFS: [torolab4\(torrcf\)](#)  
IBMMAIL: [ibmmail\(caibmwt9\)](#)

To send your comments by facsimile (attention: RCF coordinator) use the following FAX numbers:

United States and Canada: 416-448-6161  
Other Countries: (+1)-416-448-6161

Alternatively, you can mail your comments directly to:

IBM Canada Ltd. Laboratory  
Information Development  
2G/345/1150/TOR  
1150 Eglinton Avenue East  
North York, Ontario, Canada. M3C 1H7

If you send comments, include the title and order number of this book, and the page number or topic related to your comment. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1996, 1998. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> . . . . .	<b>v</b>	Hardware Exceptions and Signals . . . . .	13
<b>OS/390 V2R6.0 C/C++ Reference</b>		Compiler Return Codes. . . . .	13
<b>Summary</b> . . . . .	<b>1</b>	Compiler Options. . . . .	14
Purpose of This Book . . . . .	1	Available under C and C++ Compile and IPA	
Where to Find More Information . . . . .	1	Link . . . . .	14
OS/390 C/C++ Lexical Elements . . . . .	1	Available under C Compilation and IPA Link	15
Character Set . . . . .	1	Available under C++ Compile . . . . .	16
Trigraphs . . . . .	2	Pragma Directives . . . . .	17
Digraphs (C++ Only) . . . . .	2	Supported by Both C and C++ . . . . .	17
Keywords . . . . .	2	Supported by C Only . . . . .	20
Additional Keywords (C++ Only) . . . . .	3	Supported By C++ Only . . . . .	22
Escape Sequences. . . . .	3	Library Functions. . . . .	25
Storage Classes . . . . .	3	Utilities . . . . .	85
Predefined Types . . . . .	3	CC Utility . . . . .	85
Derived Types . . . . .	4	CMOD Utility . . . . .	85
Type Qualifiers . . . . .	4	CXX Utility . . . . .	85
Operator Precedence. . . . .	4	CXXMOD Utility . . . . .	85
Redirection Symbols. . . . .	5	CXXBIND Utility . . . . .	86
printf(), fprintf(), & sprintf() Conversion		C/C++ Compilation Utilities under the OS/390	
Specifications . . . . .	6	UNIX Shell . . . . .	87
printf(), fprintf(), & sprintf() Type Conversion		Object Library Utility . . . . .	87
Specifiers . . . . .	6	CPLINK Utility . . . . .	87
printf(), fprintf(), sprintf() Flag Characters . . . . .	7	DLLRNAME Utility . . . . .	88
scanf(), fscanf(), & sscanf() Conversion		CXXFILT Utility . . . . .	89
Specifications . . . . .	9	Locale Utility (localedef) . . . . .	89
scanf(), fscanf() & sscanf() Conversion		iconv Utility . . . . .	89
Specifiers . . . . .	9	genxlt Utility . . . . .	89
The ____amrc Structure. . . . .	11	DSECT Conversion Utility . . . . .	90



---

## Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY, 10594, USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication documents *intended* Programming Interfaces that allow the customer to write OS/390 C/C++ programs.

Any interfaces, including service component interfaces, that are not documented in the OS/390 C/C++ publications are not formal interfaces. You should not build any dependencies on these interfaces, as IBM can change or remove interfaces at any time, without notice.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.





---

# OS/390 V2R6.0 C/C++ Reference Summary

---

## Purpose of This Book

This book contains a summary of frequently used information for OS/390 C/C++. The following table lists the other OS/390 C/C++ books which contain more detailed information on the topics covered in this book.

## Where to Find More Information

Subject	Book Key
Character Set	LR
Trigraphs	LR
Digraphs	LR
Keywords	LR
Escape Sequences	LR
Storage Classes	LR
Predefined Types	LR
Derived Types	LR
Type Qualifiers	LR
Operator Precedence	LR
Redirection Symbols	PG
printf(), fprintf(), & sprintf() Conversion Specifications	LIBR
scanf(), fscanf() & sscanf() Conversion Specifications	LIBR
The __amrc Structure	LIBR
Hardware Exceptions and Signals	PG
Compiler Return Codes	UG
Compiler Options	UG
Pragma Directives	LR
Library Functions	LIBR
Utilities	UG

**LR**     *OS/390 C/C++ Language Reference*

**PG**     *OS/390 C/C++ Programming Guide*

**UG**     *OS/390 C/C++ User's Guide*

**LIBR**   *OS/390 C/C++ Run-Time Library Reference*

---

## OS/390 C/C++ Lexical Elements

### Character Set

a b c d e f g h i j k l m n o p q r s t u v w x y z  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
0 1 2 3 4 5 6 7 8 9

! " # % & ' ( ) \* + , - . / :  
 ; < = > ? [ \ ] \_ { | } ~ ^

The space character.

The control characters representing horizontal tab, vertical tab, form feed, and end of string.

Valid EBCDIC multibyte sequences for strings, character constants and comments.

## Trigraphs

A trigraph is a sequence of three characters. You can use trigraphs in your source program to enter symbols that are not available on your keyboard.

Character	Trigraph
#	??=
[	??(
]	??)
{	??<
}	??>
\	??/
	??!
~	??-
^	??'

## Digraphs (C++ Only)

A digraph is a sequence of two characters. You can use digraphs in your source program to enter C++ symbols that are not available on your keyboard.

Character	Digraph
{	<%
}	%>
[	<:
]	:>
#	%:
##	%.%:

## Keywords

asm <sup>2</sup>	double	new <sup>2</sup>	switch
auto	else	operator <sup>2</sup>	template <sup>2</sup>
break	enum	_Packed <sup>1</sup>	this <sup>2</sup>
case	_Export <sup>2</sup>	private <sup>2</sup>	throw <sup>2</sup>
catch <sup>2</sup>	extern	protected <sup>2</sup>	typedef
__cdecl <sup>2</sup>	float	public <sup>2</sup>	try <sup>2</sup>
char	for	register	union
class <sup>2</sup>	friend <sup>2</sup>	return	unsigned
const	goto	short	virtual <sup>2</sup>
continue	if	signed	void
default	inline <sup>2</sup>	sizeof	volatile
delete <sup>2</sup>	int	static	wchar_t <sup>2</sup>
do	long	struct	while

Future versions of the C++ compiler may reserve the following keywords, so you should avoid using them in your applications:

*Table 1. C++ Keywords (Future)*

bool	mutable	true
const_cast	namespace	typeid
dynamic_cast	reinterpret_cast	typename
explicit	static_cast	using
false		

## Additional Keywords (C++ Only)

Characters produced	Keyword
&	bitand
&&	and
	bitor
	or
^	xor
~	compl
&=	and_eq
=	or_eq
^=	xor_eq
!	not
!=	not_eq

These additional keywords can be used in your C++ source program to enter symbols that are not available on your keyboard. They are activated with the above meanings by using the DIGRAPH compiler option.

## Escape Sequences

Escape sequence	Character represented
\a	Alarm (bell)
\b	Backspace
\f	Form feed (new page)
\n	New-line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\'	Single quotation
\"	Double quotation
\?	Question mark
\\	Backslash

## Storage Classes

auto	register	static	extern
------	----------	--------	--------

## Predefined Types

The following types are predefined:

- void
- char
- signed char
- unsigned char

1. C only.

2. C++ only.

- short, signed short, short int, signed short int
- unsigned short, unsigned short int
- int, signed, signed int
- unsigned, unsigned int
- long, signed long, long int, signed long int
- unsigned long, unsigned long int
- long long, signed long long, long long int, signed long long int
- unsigned long long, unsigned long long int
- float
- double
- long double
- decimal (n,p)<sup>3</sup>

## Derived Types

The following categories of derived types may be constructed by the user:

- array
- enum
- function
- pointer
- struct
- typedef
- union
- class <sup>4</sup>

## Type Qualifiers

- const
- volatile
- `_Packed` <sup>3</sup>

## Operator Precedence

	Associativity	Operators	Digraphs
Primary scope resolution <sup>4</sup>	left to right	::	
Primary	left to right	() [] . ->	
Unary	right to left	++ -- - + ! ~ & * ( <i>typename</i> ) sizeof digitsof <sup>3</sup> precisionof <sup>3</sup> new <sup>4</sup> delete <sup>4</sup>	
C++ pointer to member <sup>4</sup>	left to right	.* -> *	
Multiplicative	left to right	* \ %	
Additive	left to right	+ -	
Bitwise Shift	left to right	<< >>	
Relational	left to right	< > <= >=	
Equality	left to right	== !=	
Bitwise Logical AND	left to right	&	bitand
Bitwise Exclusive OR	left to right	^	xor

3. C only.

4. C++ only.

	Associativity	Operators	Digraphs
Bitwise Inclusive OR	left to right		bitor
Logical AND	left to right	&&	and
Logical OR	left to right		or
Conditional	right to left	? :	
Assignment	right to left	= += -= *= \= <<= >>= %= &= ^=	
Comma	left to right	,	

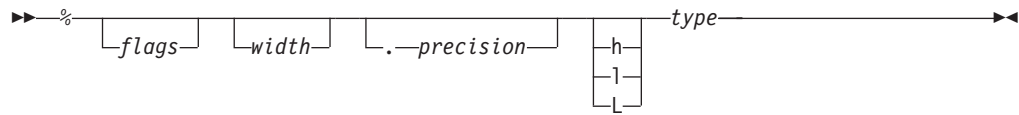
## Redirection Symbols

Symbol	Description
< <i>fn</i>	associates file specified as <i>fn</i> with stdin; reopens <i>fn</i> in mode r.
0< <i>fn</i>	associates file specified as <i>fn</i> with stdin; reopens <i>fn</i> in mode r.
> <i>fn</i>	associates file specified as <i>fn</i> with stdout; reopens <i>fn</i> in mode w.
1> <i>fn</i>	associates file specified as <i>fn</i> with stdout; reopens <i>fn</i> in mode w.
>> <i>fn</i>	associates file specified as <i>fn</i> with stdout; reopens <i>fn</i> in mode a.
2> <i>fn</i>	associates file specified as <i>fn</i> with stderr; reopens <i>fn</i> in mode w.
2>> <i>fn</i>	associates file specified as <i>fn</i> with stderr; reopens <i>fn</i> in mode a.
2>&1	associate stderr with stdout; same file and mode.
1>&2	associate stdout with stderr; same file and mode.

### Notes:

1. If you use the NOREDIR option on a #pragma runopts directive, or specify NOREDIR as a compiler option, you cannot redirect standard streams on the command line using the preceding list of symbols.
2. In the above table, 0, 1, and 2 represent, respectively, stdin, stdout, and stderr.

## printf(), fprintf(), & sprintf() Conversion Specifications



Field	Description
<b>flags</b>	Justification of output and printing of thousands' groupings, signs, blanks, decimal points, octal, and hexadecimal prefixes, and the semantics for wchar_t precision unit.
<b>width</b>	Minimum field width. You can specify width with an asterisk (*). If you specify an asterisk, the width comes from the next argument in the argument list.
<b>precision</b>	Optional precision that gives the minimum number of digits for the d, i, o, u, x and X conversions, or the number of digits after the decimal point for the e, E, and f conversions, or the maximum number of significant digits for the g and G conversions, or the maximum number of characters to be written from a string in an s conversion. You can specify precision with an asterisk (*). If you specify an asterisk, precision comes from the next argument in the argument list.
<b>h,l,L</b>	Size of argument expected: <ul style="list-style-type: none"> <li><b>h</b> A prefix, with the conversion specifiers d, i, o, u, x, and X, that specifies that the argument is a short int, or a prefix that specifies that a following n conversion specifier applies to a pointer to a short int.</li> <li><b>l</b> A prefix, with the specifiers d, i, o, u, x, and X, that specifies that the argument is a long int, or a prefix that specifies that a following n conversion specifier applies to a pointer to a long int.</li> <li><b>L</b> A prefix, with the specifiers e, E, f, g, or G, that specifies that the argument is a long double.</li> </ul>

## printf(), fprintf(), & sprintf() Type Conversion Specifiers

Character	Argument Type	Output Format
d, i	int	Signed decimal integer
u	unsigned int	Unsigned decimal integer
o	unsigned int	Unsigned octal integer
x	unsigned int	Unsigned hexadecimal integer, using "abcdef"
X	unsigned int	Unsigned hexadecimal integer, using "ABCDEF"
e	double	Floating-point signed value having the form <code>[-]d.ddde[sign]ddd</code> , where <i>d</i> is a single decimal digit, <i>ddd</i> is one or more decimal digits, <i>ddd</i> is at least two decimal digits, and <i>sign</i> is + or -
E	double	Floating-point identical to the e format except that E introduces the exponent

Character	Argument Type	Output Format
f	double	Floating-point signed value having the form <code>[-]dddd.dddd</code> , where <i>dddd</i> is one or more decimal digits
g	double	Floating-point signed value printed in f or e format, whichever is more compact for the given value and <i>precision</i> ; the e format is used only when the exponent of the value is less than -4 or greater than <i>precision</i>
G	double	Floating-point identical to the g format except that E introduces the exponent (where appropriate)
D( <i>n,p</i> )	Decimal type argument	Fixed-point value consisting of a series of one or more decimal digits, possibly containing a decimal point
c	int	int converted to an unsigned char, and written as a single character
s	Pointer to array of characters	String of characters up to but not including the null character ( <code>\0</code> ) or until <i>precision</i> is reached
n	Pointer to integer, which contains the number of characters successfully written so far to the stream or buffer	No output
p	Pointer to void	Value of the pointer, converted to a sequence of printable characters
C or lc	wchar_t character	Multibyte character
S or ls	Array of type wchar_t	Multibyte characters printed up to the first wchar_t null character ( <code>L'\0'</code> ) or until <i>precision</i> is reached
%	No argument	A % is written; the complete conversion specification is %%

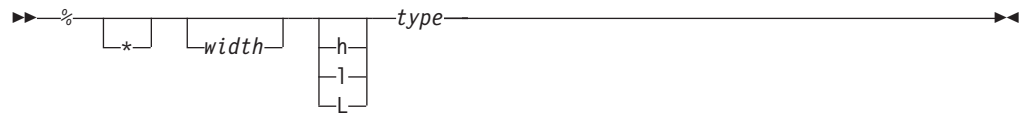
## printf(), fprintf(), sprintf() Flag Characters

Flag	Meaning
''	The integer portion of the result of a decimal conversion( <code>%i</code> , <code>%d</code> , <code>%u</code> , <code>%f</code> , <code>%g</code> , or <code>%G</code> ) will be formatted with the thousands' grouping characters.
-	Left-justifies the result within the field width.
+	Prefixes the output value with a sign (+ or -) if the signed conversion is specified.
blank (' ')	If the result of a signed conversion is a positive value or no characters, a space is prefixed to the result. If the + and the <i>blank</i> flags both appear, the <i>blank</i> flag is ignored, and a positive signed value will be output with a sign.
0	Leading zeros are used to pad to the specified field width. If the 0 and - flags both appear, the 0 flag will be ignored. For d, i, o, u, x, and X conversions, if a precision is specified, the 0 flag will be ignored.

Flag	Meaning
#	<p>When used with the o, x, or X formats, the # flag prefixes any nonzero output value with 0, 0x, or 0X, respectively.</p> <p>When used with the f, e, or E formats, the # flag forces the output value to contain a decimal point in all cases.</p> <p>When used with the g or G formats, the # flag forces the output value to contain a decimal point in all cases and prevents the truncation of trailing zeros.</p> <p>When used with the lS or S format, the # flag causes precision to be measured in multibyte characters, rather than in bytes.</p>



## scanf(), fscanf(), & sscanf() Conversion Specifications



### Field Description

- \*** An asterisk following the percent sign suppresses assignment of the next input field, which is interpreted as a field of a specified *type*. The field is scanned but not stored.
- width** Maximum number of characters to be read from input.
- h** A prefix, with the conversion specifiers `d` and `i`, that specifies that the argument is a pointer to a short int, or a prefix that specifies that a following `n` specifier applies to a pointer to a short int.  
  
A prefix, with the specifiers `o`, `a`, `x`, and `X`, that specifies that the argument is a pointer to an unsigned short int.  
  
A prefix, with the specifiers `e`, `E`, `f`, `g`, and `G`, that specifies that the argument is a pointer to a float.  
  
The `h` modifier is ignored if specified for any other *type*.
- l** A prefix, with the specifiers `d` and `i`, that specifies that the argument is a pointer to a long int, or a prefix that specifies that a following `n` specifier applies to a pointer to a long int.  
  
A prefix, with the specifiers `o`, `u`, `x`, and `X`, that specifies that the argument is a pointer to an unsigned long int.  
  
A prefix, with the specifiers `e`, `E`, `f`, `g`, and `G`, that specifies that the argument is a pointer to a double.  
  
The `l` modifier is ignored if specified for any other *type*.
- L** A prefix, with the specifiers `e`, `E`, `f`, `g`, and `G`, that specifies that the argument is a pointer to a long double.

## scanf(), fscanf() & sscanf() Conversion Specifiers

Character	Type of Input Expected	Argument Type
d	Decimal integer	Pointer to integer
o	Octal integer	Pointer to unsigned integer
x,X	Hexadecimal integer	Pointer to unsigned integer
i	Decimal, hexadecimal, or octal integer	Pointer to integer
u	Unsigned decimal integer	Pointer to unsigned integer
e,f,g	Floating-point value consisting of an optional sign (+ or -), a series of one or more decimal digits possibly containing a decimal point, and an optional exponent (e or E) followed by an integer value that may be signed	Pointer to floating type

Character	Type of Input Expected	Argument Type
E, G	Floating-point identical to the e and g formats	Pointer to floating type
D( <i>n,p</i> )	Fixed-point value consisting of an optional sign (+ or -); a series of one or more decimal digits possibly containing a decimal point	Pointer to decimal
c	Character, or sequence of characters of length specified as field width; white-space characters that are ordinarily skipped are read when c is specified	Pointer to the first element of an array of characters
s	String of characters scanned up to the first white-space character or until <i>precision</i> is reached	Pointer to the first element of an array of characters
n	No input read from stream or buffer	Pointer to integer, where the number of characters successfully read from the stream or buffer up to that point in the call to fscanf are stored
l c or C	Multibyte character	Pointer to wchar_t character
l s or S	Multibyte string	Array of type wchar_t
p	Sequence of characters converted to a pointer	Pointer to a pointer to void
[]	Strings of characters that are part of the scan set are scanned	Pointer to the first element of an array of characters
%	% matches a single %; the complete conversion specification is %%;	No conversion or assignment occurs

---

## The `__amrc` Structure

The `__amrc` structure is defined in `stdio.h`. You can use it to obtain further information when a call to `perror()` does not provide sufficient details about an I/O error. Refer to the OS/390 C/C++ Programming Guide for more information.

**Note:** `__amrc` is NULL under CICS™.

---

```
typedef struct {
    union { 1
        long int __error; 2
        struct {
            unsigned short __syscode,
                        __rc;
        } __abend; 3
        struct {
            unsigned char __fdbk_fill,
                        __rc,
                        __ftncd,
                        __fdbk;
        } __feedback; 4
        struct {
            unsigned short __svc99_info,
                        __svc99_error;
        } __alloc; 5
    } __code;
    unsigned long __RBA; 6
    unsigned int __last_op; 7
    struct {
        unsigned long __len_fill;
        unsigned long __len;
        char __str[120];
        unsigned long __parmr0;
        unsigned long __parmr1;
        unsigned long __fill2[2];
        char __str2[64];
    } __msg; 8
} __amrc_type;
```

---

- |                                  |  |
|----------------------------------|--|
| <b>1</b> <code>__code</code>     | The error or warning value from an I/O operation is in <code>__error</code> , <code>__abend</code> , <code>__feedback</code> , or <code>__alloc</code> . Examine <code>__last_op</code> to determine how to interpret the <code>__code</code> union. |
| <b>2</b> <code>__error</code>    | Contains the return code from the system macro or utility.   |
| <b>3</b> <code>__abend</code>    | Contains the abend code when <code>errno</code> is set to indicate a recoverable I/O abend. <code>__syscode</code> is the system abend code and <code>__rc</code> is the return code.  |
| <b>4</b> <code>__feedback</code> | <code>__rc</code> stores the VSAM register 15, <code>__fdbk</code> stores the VSAM error code or reason code, and <code>__RBA</code> stores the RBA after some operations.   |
| <b>5</b> <code>__alloc</code>    | Contains error from <code>fopen</code> and <code>freopen</code> calls when defining files to the system using SVC 99.  |
| <b>6</b> <code>__RBA</code>      | This is the RBA value returned by VSAM after an ESDS or KSDS record is written out.  |
| <b>7</b> <code>__last_op</code>  | This field will be one of the following values indicating the last I/O operation:  |

`__BSAM_CLOSE`

`__BSAM_CLOSE_T`

__BSAM_NOTE	__BSAM_OPEN
__BSAM_POINT	__BSAM_READ
__BSAM_WRITE	__CELMSGF_WRITE
__HSP_CREATE	__HSP_DELETE
__HSP_EXTEND	__HSP_READ
__HSP_WRITE	__IO_DEVTTYPE
__IO_INIT	__IO_RDJFCB
__SVC99_ALLOC	__SVC99_ALLOC_NEW
__TGET_READ	__TPUT_WRITE
__VSAM_CLOSE	__VSAM_ENDREQ
__VSAM_ERASE	__VSAM_GENCB
__VSAM_GET	__VSAM_MODCB
__VSAM_OPEN_ESDS	__VSAM_OPEN_ESDS_PATH
__VSAM_OPEN_FAIL	__VSAM_OPEN_KSDS
__VSAM_OPEN_KSDS_PATH	__VSAM_OPEN_RRDS
__VSAM_POINT	__VSAM_PUT
__VSAM_SHOWCB	__VSAM_TESTCB

**8** \_\_msg      May contain the system error messages from read or write operations emitted from the BSAM SYNADAF macro instruction.

---

## Hardware Exceptions and Signals

C Signal	Hardware Exception
SIGILL	<ul style="list-style-type: none"><li>• operation exception</li><li>• privileged operation exception</li><li>• execute exception</li></ul>
SIGSEGV	<ul style="list-style-type: none"><li>• protection exception</li><li>• addressing exception</li><li>• specification exception</li></ul>
SIGFPE	<ul style="list-style-type: none"><li>• data exception</li><li>• fixed-point divide exception</li><li>• decimal divide exception<sup>5</sup></li><li>• exponent overflow exception</li><li>• floating-point divide exception</li><li>• decimal overflow exception<sup>5</sup></li></ul>
masked	<ul style="list-style-type: none"><li>• fixed-point overflow exception</li><li>• exponent underflow exception</li><li>• significance exception</li></ul>

---

## Compiler Return Codes

Return Code	Type of Error Detected	Compilation Result
0	No error detected; Informational messages may have been issued.	Compilation completed. Successful execution anticipated.
4	Warning error detected.	Compilation completed. Execution may not be successful.
8	Error detected.	Compilation may have been completed. Successful execution not possible.
12	Severe error detected.	Compilation may have been completed. Successful execution not possible.
16	Terminating error detected.	Compilation terminated abnormally. Successful execution not possible.
33	A library level prior to OS/390 Version 2 Release 6 Language Environment was used.	Compilation terminated abnormally. Successful execution not possible.

---

<sup>5</sup>. C only.

## Compiler Options

The following three tables list compiler options. The first table lists compiler options available under C Compile, C++ Compile, and IPA Link. The second table lists compiler options available only for C Compile and IPA Link. The third table lists compiler options available only for C++.

### Available under C and C++ Compile and IPA Link

Compiler Options	Default
ANSiAlias   NOANSiAlias	ANS
ARCHiecture   NOARCHiecture	ARCH(0)
ARGparse   NOARGparse	ARG
CONVlit   NOCONVlit	NOCONV
CSEct   NOCSEct	NOCSE
DEFine( <i>name1</i> [= <i>def1</i> ], <i>name2</i> [= <i>def2</i> ], ...)	no definitions
EVENTs   NOEVENTs	NOEVENT
EXECops   NOEXECops	EXEC
EXPmac   NOEXPmac	NOEXP
EXPOrta11   NOEXPOrta11	NOEXP0
FLag( <i>sev</i> )   NOFLag	FL(1)
FLOAT( <i>suboptions</i> )	FLOAT( HEX, FOLD, NOMAF, NORRM, NOAFP or AFP)  For ARCH(2) compiler option, default is NOAFP. For ARCH(3) or higher, the default is AFP.
GENPch( <i>filename</i> )   NOGENPch	NOGENP
GONUMber   NOGONUMber	NOGONUM
HALT( <i>n</i> )	HALT(16)
INLRpt[( <i>filename</i> )]   NOINLRpt[( <i>filename</i> )]	NOINLR
IPA[( <i>suboptions</i> )]   NOIPA[( <i>suboptions</i> )]	NOIPA
LANGlvl( <i>language standard level</i> )	LANG (EXTENDED)
LIBansi   NOLIBansi	NOLIB
LISt[( <i>filename</i> )]   NOLIS[( <i>filename</i> )]	NOLIS
LOCale[( <i>name</i> )]   NOLOCale	NOLOC
LOngname   NOLOngname	Under C Compilation and IPA Link: NOLO  Under C++ Compilation: LO
LSEarch( <i>opt1, opt2, ...</i> )   NOLSEarch	NOLSE
MARGins( <i>m, n</i> )   NOMARGins ( <i>m, n</i> valid only under C)	F-format: MAR(1,72) V-format: NOMAR (Default under C++ is always NOMAR)
MAXMem   NOMAXMem	MAXM(2048)
MEMory   NOMEMory	MEM
NEStinc( <i>limit</i> )   NONESTinc	NONEST
OBJect   NOOBJect[( <i>filename</i> )]   OBJECT( <i>filename</i> )	OBJ

Compiler Options	Default
OE[(filename)]   NOOE	NOOE
OFFset   NOOFFset	NOOF
OPTFile[(filename)]   NOOPTFile[(filename)]	NOOPTF
OPTimize[(n)]   NOOPTimize	Under C or C++ compilation: NOOPT Under IPA Link: OPT
PLIST(OS   HOST)	PLIST(HOST)
PPonly[(suboptions)]   NOPPonly[(suboptions)]	NOPP
REDir   NOREDir	RED
ROUND(mode)	ROUND(N)
SEarch(opt1,opt2,...)   NOSEarch	NOSE
SERvice(string)   NOSERvice	NOSERV
SEquence(m,n)   NOSEquence (m,n only valid under C)	F-format: SEQ(73,80) V-format: NOSEQ (Default under C++ is always NOSEQ)
SHOWinc   NOSHOWinc	NOSHOW
SOURCE[(filename)]   NOSOURCE[(filename)]	NOSO
SPill   NOSpill	SP(128)
STArt   NOSTArt	STA
STRICT   NOSTRICT	NOSTRICT
SYSLib(pdsname-list)	no action
TARG(( )   LE   IMS) (( ) is not valid under C++)	TARG(LE)
TERMinal   NOTERMinal	TERM
TEST[(suboptions)]   NOTEST[(suboptions)] (Only HOOK is supported for C++)	Under C: NOTEST(SYM BLOCK  LINE NOPATH  HOOK) Under C++: NOTEST(HOOK)
TUNe(n)	TUNE(3)
UNDEFine(name)	no action
USEPch(filename)   NOUSEPch	NOUSEP
USERLib(pdsname-list)	no action
WSIZEOF   NOWSIZEOF	NOWSIZEOF
XRef   NOXRef	NOXR

## Available under C Compilation and IPA Link

Compiler Options	Default
AGGgregate   NOAGGgregate	NOAGG
ALIAS(name)   ALIAS()   NOALIAS <sup>6</sup>	NOALI
CHEckout(suboptions)   NOCHEckout	NOCHE
DECK   NODECK	NODECK
DLL(suboption)   NODLL(suboption)	NODLL(nocba)

Compiler Options	Default
INLine[(AUTO NOAUTO, REPORT NOREPORT, <i>threshold, limit</i> )]   NOINLine[( <i>suboptions</i> )]	NOINL(AUTO, REPORT,100,1000)
OMVS[( <i>filename</i> )]   NOOMVS	NOOMVS
RENT NORENT	NORENT
SScomm NOSScomm	NOSS
UPConv NOUPConv	NOUPC

## Available under C++ Compile

Compiler Options	Default
ATtribute[(FULL)]   NOATtribute	NOATT
DIGRaph NODIGRaph	NODIGR
EXH NOEXH	EXH
FASTTEMPINC	NOFASTT
INfo[( <i>suboptions</i> )]   NOINFO	NOIN
PORT( <i>suboptions</i> )   NOPORT( <i>suboptions</i> )	NOPORT(NOPPS)
SOM NOSOM	NOSOM
SOMGs NOSOMGs	NOSOMG
SOMRo( <i>classname1, classname2,...</i> )   NOSOMRo	NOSOMR
SOMEInit NOSOMEInit	SOMEI
SOMVolattr NOSOMVolattr	NOSOMV
SRCMsg NOSRCMsg	NOSRCM
SYSPath( <i>path1, path2,...</i> )   NOSYSPath	NOSYSP
TEMPINC[( <i>pdsname</i>   <i>hfs directory</i> )]   NOTEMPINC[( <i>pdsname</i>   <i>hfs directory</i> )]	Default for <i>pdsname</i> is TEMPINC(TEMPINC). Default for <i>hfs directory</i> is TEMPINC(./tempinc).
USERpath( <i>path1, path2,...</i> )   NOUSER	NOUSER
XSominc( <i>opt1,opt2,...</i> )   NOXSominc	NOXS



---

## Pragma Directives

The three following groups of syntax diagrams list the pragma directives supported by both C and C++, those supported only by C, and those supported only by C++.

### Supported by Both C and C++

**#pragma chars:**

▶▶ #pragma chars ( signed  
unsigned ) ▶▶

**#pragma checkout:**

▶▶ #pragma checkout ( resume  
suspend ) ▶▶

**#pragma comment:**

▶▶ #pragma comment ( | choices | ) ▶▶

**choices:**

compiler  
date  
timestamp  
copyright  
user , "token-seq" ▶▶

**#pragma convlit:**

▶▶ #pragma convlit ( resume  
suspend ) ▶▶

**#pragma csect:**

▶▶ #pragma csect ( CODE  
STATIC  
TEST , " name " ) ▶▶

**#pragma export:**

►► #pragma export ( *variable-name* | *function-name* ) ►►

#### #pragma filetag:

►► ??=pragma filetag ( - " *code-set-name* " ) ►►

#### #pragma hdrstop:

►► #pragma hdrstop ►►

#### #pragma inline:

►► #pragma <sup>(1)</sup> *inline* | *noinline* ( *identifier* ) ►►

#### Notes:

- 1 Not supported under C++

#### #pragma isolated\_call:

►► #pragma isolated\_call ( *identifier* ) ►►

#### #pragma langlvl:

►► #pragma langlvl ( <sup>(1)</sup> *option* ) ►►

#### Notes:

- 1 Under C, *option* can be -ANSI, -COMMONC, -EXTENDED, -SAA or -SAAL2; Under C++, *option* can be -ANSI, -EXTENDED or -COMPAT

#### #pragma linkage:

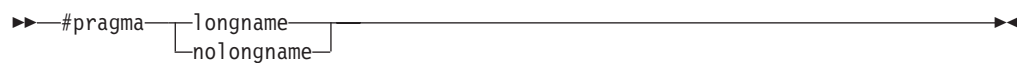
►► #pragma linkage ( *identifier* , *OS* | *FETCHABLE* | *PLI* | *COBOL* | *FORTRAN* | *RETURNCODE* ) ►►

<sup>(1)</sup>

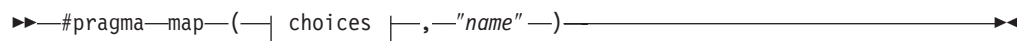
**Notes:**

1 C++ supports `#pragma linkage(fetchable, -funcname -)`

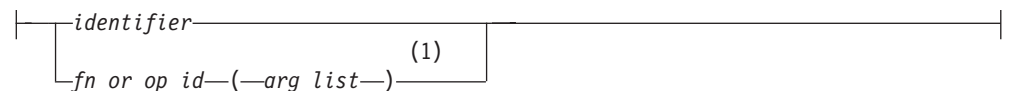
**#pragma longname:**



**#pragma map:**



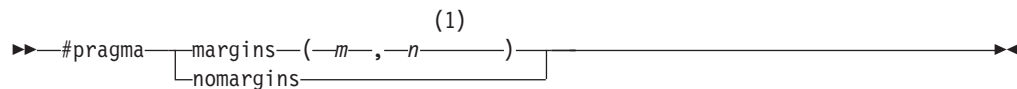
**choices:**



**Notes:**

1 Not supported under C

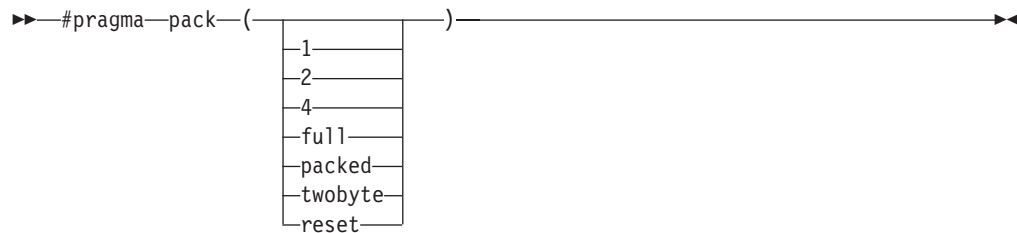
**#pragma margins:**



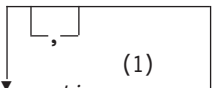
**Notes:**

1 Not supported under C++

**#pragma pack:**




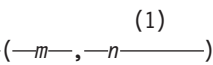
**#pragma runopts:**

►► #pragma runopts (  option ) ►►

**Notes:**

- 1 Refer to the *-OS/390 Language Environment Programming Reference* and to the *-OS/390 C/C++ User's Guide* for information about the available run time options.

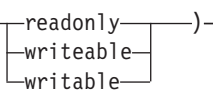
**#pragma sequence:**

►► #pragma  sequence (  (1) ) ►►


**Notes:**

- 1 Not supported under C++

**#pragma strings:**

►► #pragma strings (  ) ►►

**#pragma variable:**


►► #pragma variable (  ) ►►

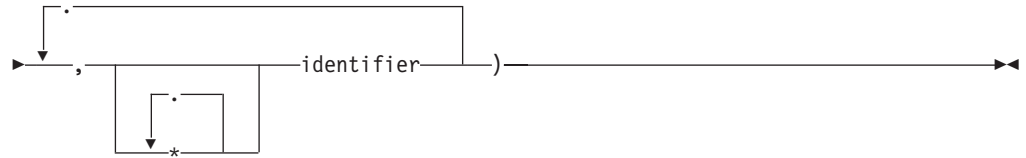
**#pragma wsizeof:**

►► #pragma wsizeof (  ) ►►

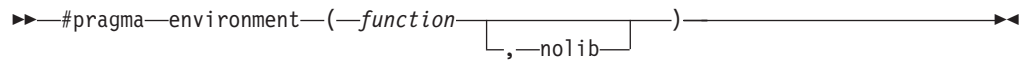
## Supported by C Only

**#pragma disjoint:**

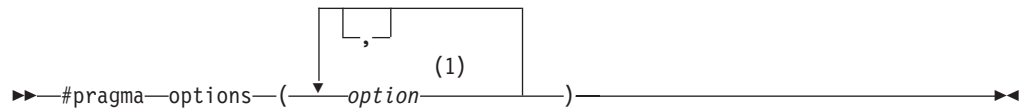
►► #pragma disjoint (  identifier ) ►►



### #pragma environment:



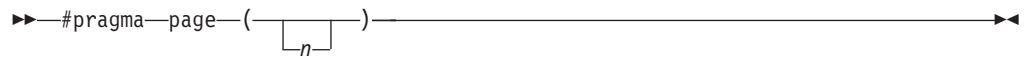
### #pragma options:



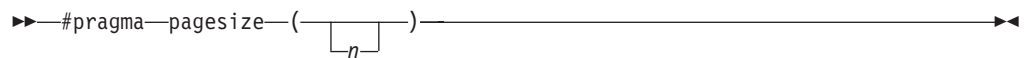
### Notes:

- 1 -option can be -AGG, -ALI, -ARCH, -CHE, -DECK, -GONUM, -INL, -LIB, -MAXM, -OBJ, -OPT, -RENT, -SERV, -SP, -STA, -TUN, -TEST, -UPC, or -XR.

### #pragma page:



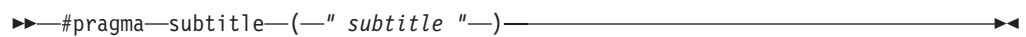
### #pragma pagesize:



### #pragma skip:



### #pragma subtitle:



### #pragma target:

►► #pragma target (  ) ►►

#### #pragma title:

►► #pragma title (—" title "—) ►►

## Supported By C++ Only

#### #pragma define:

►► #pragma define (—*template-class-name*—) ►►

#### #pragma implementation:

►► #pragma implementation (—*string-literal*—) ►►

#### #pragma info:

►► #pragma info (—resume suspend—) ►►


#### #pragma priority:

►► #pragma priority (—*n*—) ►►

#### #pragma SOM:

►► #pragma SOM ►►

#### #pragma SOMAsDefault:

►► #pragma SOMAsDefault (  ) ►►

#### #pragma SOMAttribute:

```

>> #pragma SOMAttribute(—DataMember—, —indirect—)
                                     |
                                     |—nodata—
                                     |—noget—
                                     |—noset—
                                     |—privatedata—
                                     |—protectedata—
                                     |—publicdata—
                                     |—readonly—
                                     |—virtualaccessors—

```

#### #pragma SOMCallStyle:

```

>> #pragma SOMCallStyle(—OIDL—)
                          |
                          |—IDL—

```

#### #pragma SOMClassInit:

```

>> #pragma SOMClassInit(—*—, —SOM-Linkage-Prototype—)
                        |
                        |—C++ ClassName—

```

#### #pragma SOMClassName:

```

>> #pragma SOMClassName(—*—, —"NameOfSomClass"—)
                        |
                        |—C++ ClassName—

```

#### #pragma SOMClassVersion:

```

>> #pragma SOMClassVersion(—C++ ClassName—, —Major—, —Minor—)
                           |
                           |—*—

```

#### #pragma SOMDataName:

```

>> #pragma SOMDataName(—C++ DataMember—, —"SomName"—)

```

#### #pragma SOMDefine:

```

>> #pragma SOMDefine(—*—)
                    |
                    |—on—
                    |—off—
                    |—pop—
                    |—C++ ClassName—

```

### #pragma SOMMetaClass:

▶▶#pragma SOMMetaClass(C++ ClassName\*,"SOMClassName" C++ MetaClassName)

### #pragma SOMMethodAppend:

▶▶#pragma SOMMethodAppend(C++ FunctionPrototype,"string")

### #pragma SOMMethodName:

▶▶#pragma SOMMethodName(C++ Prototype,"SomMethodName")  
C++ FunctionName

### #pragma SOMNoDataDirect:

▶▶#pragma SOMNoDataDirect(\*  
on  
off  
off  
pop)

### #pragma SOMReleaseOrder:

▶▶#pragma SOMReleaseOrder(\*  
StaticDataMember  
choices  
!)

### choices:

Attribute  
C++ MemFnProto  
C++ UnambFnName  
"SOMMethodName"



---

## Library Functions

### Function Format

---

```
#include <stdlib.h>
void abort(void);
```

---

```
#include <stdlib.h>
int abs(int n);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int accept(int socket, struct sockaddr *address,
           size_t *address_len);
```

---

```
#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int accept(int socket, struct sockaddr *address,
           int *address_len);
```

---

```
#define _OPEN_SYS_SOCK_EXT2
#include <sys/socket.h>
int accept_and_recv(int socket, int *accept_socket,
                    struct sockaddr *remote_address,
                    socklen_t *remote_address_len,
                    struct sockaddr *local_address,
                    socklen_t *local_address_len,
                    void *buffer, size_t length);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int access(const char *pathname, int how);
```

---

```
#include <math.h>
double acos(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double acosh(double x);
```

---

```
#define _XOPEN_SOURCE
#include <regex.h>
int advance(const char *string, const char *expbuf);
extern char *loc2, *locs;
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
unsigned int alarm(unsigned int seconds);
```

---

```
#include <stdlib.h>
char *alloca(int size);
```

---

```
#include <time.h>
char *asctime(const struct tm *timeptr);
```

---

```
#include <math.h>
double asin(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double asinh(double x);
```

---

```
#include <assert.h>
void assert(int expression);
```

---

## Function Format

---

```
#include <math.h>
double atan(double x);
double atan2(double y, double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double atanh(double x);
```

---

```
#include <stdlib.h>
int atexit(void (*func)(void));
```

---

```
#include <unistd.h>
int __atof(char string);
```

---

```
#include <unistd.h>
int __atof_l(void * bufferptr, int leng);
```

---

```
#include <stdlib.h>
double atof(const char *string);
```

---

```
#include <stdlib.h>
int atoi(const char *string);
```

---

```
#include <stdlib.h>
long int atol(const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
long a64l (const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <libgen.h>
char *basename(char *path);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
int bcmp(const void *s1, const void *s2, size_t n);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
void bcopy(const void *s1, void *s2, size_t n);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int bind(int socket, const struct sockaddr *address,
        size_t address_len);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int bind(int socket, struct sockaddr *address,
        int *address_len);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int brk(void * addr);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
void (*bsd_signal (int sig,
                  void (*func)(int)))(int);
```

---

```
#include <stdlib.h>
void *bsearch(const void *key, const void *base,
              size_t num, size_t size,
              int (*compare) (const void *element1,
                             const void *element2));
```

## Function Format

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
void bzero(void *s, size_t n);

#include <stdlib.h>
void *calloc(size_t num, size_t size);

#define _XOPEN_SOURCE
#include <nl_types.h>
int catclose(nl_catd catd);

#define _XOPEN_SOURCE
#include <nl_types.h>
char *catgets(nl_catd catd, int set_id,
              int msg_id, const char *s);

#define _XOPEN_SOURCE
#include <nl_types.h>
nl_catd catopen(const char *name, int oflag);

#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double cbrt(double x);

#include <collate.h>
int cclass(char *class, collat_t **list);

#include <stdlib.h>
int cds(cds_t *oldptr, cds_t *curptr,
        cds_t newwords);

#include <ctest.h>
int cdump(char *dumptitle);

#include <math.h>
double ceil(double x);

#define _POSIX_SOURCE
#include <termios.h>
speed_t cfgetispeed(const struct termios *termpr);

#define _POSIX_SOURCE
#include <termios.h>
speed_t cfgetospeed(const struct termios *termpr);

#define _POSIX_SOURCE
#include <termios.h>
int cfsetispeed(struct termios *termpr,
               speed_t speed);

#define _POSIX_SOURCE
#include <termios.h>
int cfsetospeed(struct termios *termpr,
               speed_t speed);

#define _OPEN_SYS 1
#include <sys/stat.h>
int chaudit(const char *pathname, unsigned int flags,
            unsigned int option);

#define _POSIX_SOURCE
#include <unistd.h>
int chdir(const char *pathname);
```

## Function Format

---

```
#include <unistd.h>
int __check_resource_auth_np(char *principal_uuid,
                             char *cell_uuid,
                             char *userid,
                             char *security_class,
                             char *entity_name,
                             int *access_type);
```

---

```
#include <sys/_wlm.h>
int __CheckSchEnv(const char *sched_env,
                  const char *system_name);
```

---

```
#define _POSIX_SOURCE
#include <sys/stat.h>
int chmod(const char *pathname, mode_t mode);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int chown(const char *pathname, uid_t owner,
          gid_t group);
```

---

```
#define _OPEN_SOURCE 2
#include <sys/resource.h>
int chpriority (int which, id_t who,
               int prioritytype,
               int priority);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
int chroot(const char *path);
```

---

```
#define _POSIX1_SOURCE 2
#include <env.h>
int clearenv(void);
#include <stdlib.h>
int clearenv(void);
```

---

```
#include <stdio.h>
void clearerr (FILE *stream);
```

---

```
#include <time.h>
clock_t clock(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int close(int fildes);
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int close(int socket);
#define _OE_SOCKETS
#include <unistd.h>
int close(int socket);
```

---

```
#define _POSIX_SOURCE
#include <dirent.h>
int closedir(DIR *dir);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <syslog.h>
void closelog(void);
```

---

```
#include <stdio.h>
int clrmemf(int level);
```

## Function Format

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
void __cnvblk(char bits[8], char bytes[64],
              int flag);

#include <collate.h>
int collequiv(collel_t c, collel_t **list);

#include <collate.h>
int collorder(collel_t **list);

#include <collate.h>
int collrange(collel_t start, collel_t end,
              collel_t **list);

#include <collate.h>
char *colltostr(collel_t c);

#define INIT declarations
#define GETC() getc_code
#define PEEK() peek_code
#define UNGETC() ungetc_code
#define RETURN(ptr) return_code
#define ERROR(val) error_code
#define _XOPEN_SOURCE
#include <regex.h>
char *compile(char *instring, char *expbuf,
              const char *endbuf, int eof);

#define _XOPEN_SOURCE
#include <unistd.h>
size_t confstr(int name, char *buf, size_t len);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int connect(int socket, const struct sockaddr
            *address, size_t address_len);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int connect (int socket, struct sockaddr *address,
            int address_len);

#include <sys/_wlm.h.h>
int __ConnectServer(const char *subsystem,
                   const char *subsysname,
                   const char *applenv,
                   int paralleleu);

#include <sys/_wlm.h.h>
int __ConnectWorkMgr(const char *subsysname,
                   const char *subsysname);

#include <sys/_messag.h>
int __console(struct __cons_msg *cons,
              char *modstr, int *concmd);

#include <sys/_wlm.h.h>
int __ContinueWorkUnit(wlmetok_t *enclavetoken);

#include <unistd.h>
int __convert_id_np(int function_code,
                   char *principal_uuid,
                   char *cell_uuid,
                   char *userid);
```

## Function Format

---

```
#include <math.h>
double cos(double x);
```

---

```
#include <math.h>
double cosh(double x);
```

---

```
#define _POSIX_SOURCE
#include <fcntl.h>
int creat(const char *pathname, mode_t mode);
```

---

```
#include <sys/_wlm.h.h>
int __CreateWorkUnit(wlmetok_t *enclavetoken,
                    server_classify_t classify,
                    char *arrival_time,
                    char *func_name);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
char *crypt(const char *key, const char *salt);
```

---

```
#include <stdlib.h>
int cs(cs_t *oldptr, cs_t *curptr, cs_t newword);
```

---

```
#include <stdlib.h>
int csid(const char *c);
```

---

```
#include <ctest.h>
int csnap(char *dumptitle);
```

---

```
#include <csp.h>
__csplist;7
```

---

```
#pragma runopts(env(IMS),plist(OS))
#include <ims.h>
or
#include <cics.h>
int ctdli(int parmcount, const char *function, ...);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
char *ctermid(char *string);
```

---

```
#include <ctest.h>
int ctest(char *command);
```

---

```
#include <time.h>
char *ctime(const time_t *timer);
```

---

```
#include <ctest.h>
int ctrace(char *dumptitle);
```

---

```
#define _XOPEN_SOURCE
#include <stdio.h>
char *cuserid(char *s);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
int dbm_clearerr(DBM *db);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
void dbm_close(DBM * db);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
int dbm_delete(DBM *db, datum key);
```

## Function Format

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
int dbm_error(DBM *db);

#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
datum dbm_fetch(DBM *db, datum key);

#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
datum dbm_firstkey(DBM *db);

#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
datum dbm_nextkey(DBM *db);

#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
DBM *dbm_open(const char *file, int open_flags,
              mode_t file_mode);

#define _XOPEN_SOURCE_EXTENDED 1
#include <ndbm.h>
int dbm_store(DBM *db, datum key,
              datum content, int store_mode);

#include <decimal.h>
decimal(n,p) decabs(decimal(n,p) pdec);7

#include <decimal.h>
int decchk(decimal(n,p)pdec);7

#include <decimal.h>
decimal(n,p) decfix(decimal(n,p) pdec);7

#include <sys/_wlm.h.h>
int __DeleteWorkUnit(wlmetok_t *enclavetoken);

#include <time.h>
double difftime(time_t time2, time_t time1);

#define _XOPEN_SOURCE_EXTENDED 1
#include <libgen.h>
char *dirname(char *path);

#include <sys/_wlm.h.h>
int __DisconnectServer(int *conn_tkn,

#include <stdlib.h>
div_t div(int numerator, int denominator);

#include <dll.h>
int dllfree(dllhandle* dllHandle);

#include <dll.h>
dllhandle* dllload(const char *dllName);

#include <dll.h>
void (* dllqueryfn(dllhandle *dllHandle,
                  const char *funcName)) ();

#include <dll.h>
void* dllqueryvar(dllhandle *dllHandle,
                  const char *varName);
```

## Function Format

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
double drand48(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int dup(int fildes);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int dup2(int fd1, int fd2);
```

---

```
#include <dynit.h>
int dynalloc(__dyn_t *dyn_parms);
```

---

```
#include <dynit.h>
int dynfree(__dyn_t *dyn_parms);
```

---

```
#include <dynit.h>
int dyninit(__dyn_t *dyn_parms);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *ecvt(double x, int ndigit,
           int *decpt, int *sign);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
void encrypt(char block[64], int edflag);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <grp.h>
void endgrent (void);
struct group *getgrent (void);
void setgrent (void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void endhostent(void);

#define _OE_SOCKETS
#include <netdb.h>
void endhostent();
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void endnetent(void);

#define _OE_SOCKETS
#include <netdb.h>
void endnetent();
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void endprotoent(void);

#define _OE_SOCKETS
#include <netdb.h>
void endprotoent();
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <pwd.h>
void endpwent (void);
struct passwd *getpwent(void);
void setpwent (void);
```

---



## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void endservent(void);
```

---

```
#define _OE_SOCKETS
#include <netdb.h>
void endservent();
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
void endutxent(void);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
double erand48(unsigned short int x16v[3]);
```

---

```
#include <math.h>
double erf(double x);
double erfc(double x);
```

---

```
int *___er2ad(void);
```

---

```
int __errno2(void);
```

---

```
#include <unistd.h>
int __etoa(char * string);
```

---

```
#include <unistd.h>
int __etoa_l(void * bufferptr, int leng);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
extern char **environ;
int execl(const char *path, const char *arg,
    ..., NULL);
int execle(const char *path, const char *arg,
    ..., NULL, char *const envp[]);
int execlp(const char *file, const char *arg,
    ..., NULL);
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[],
    char *const envp[]);
int execvp(const char *file, char *const argv[]);
```

---

```
#include <stdlib.h>
void exit(int status);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
void _exit(int status);
```

---

```
#include <math.h>
double exp(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double expm1(double x);
```

---

```
#include <unistd.h>
int extlink_np(const char *ename, const char *elink);
```

---

```
#include <math.h>
double fabs(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int fattach(int fildes, const char *path);
```

---

## Function Format

---

```
#include <sys/stat.h>
int fchmod(int fildes, unsigned int flags,
           unsigned int option);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int fchdir(int fildes);
```

---

```
#define _POSIX1_SOURCE 2
#include <sys/stat.h>
int fchmod(int fildes, mode_t mode);
```

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int fchown(int fildes, uid_t owner,
           gid_t group);
```

---

```
#include <stdio.h>
int fclose(FILE stream);
```

---

```
#define _POSIX_SOURCE
#include <fcntl.h>
int fcntl(int fildes, int action, ...);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *fcvt(double x, int ndigit,
           int decpt, int sign);
```

---

```
#include <stdio.h>
int fdelrec(FILE stream);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int fdetach(const char path);
```

---

```
#define _POSIX_SOURCE
#include <stdio.h>
FILE *fdopen(int fildes, const char options);
```

---

```
#include <stdio.h>
int feof(FILE stream);
```

---

```
#include <stdio.h>
int ferror(FILE stream);
```

---

```
#include <stdlib.h>
void (*fetch(const char name))();7
```

---

```
#include <stdlib.h>
void ( *fetchep( void ( entry_point )()))();7
```

---

```
#include <stdio.h>
int fflush(FILE stream);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
int ffs(int i);
```

---

```
#include <stdio.h>
int fgetc(FILE stream);
```

---

```
#include <stdio.h>
int fgetpos(FILE stream, fpos_t pos);
```

---

## Function Format

---

```
#include <stdio.h>
char *fgets(char *string, int n, FILE *stream);
```

---

```
#include <wchar.h>
wint_t fgetwc(FILE *stream);
```

---

```
#include <wchar.h>
wchar_t *fgetws(wchar_t *wcs, int n, FILE *stream);
```

---

```
#define _POSIX_SOURCE
#include <stdio.h>
int fileno(const FILE *stream);
```

---

```
#include <stdio.h>
int fldata(FILE *file, char *filename,
           fldata_t *info);
```

---

```
#include <stdio.h>
int flocate(FILE *stream, const void *key,
            size_t key_len, int options);
```

---

```
#include <math.h>
double floor(double x);
```

---

```
#include <math.h>
double fmod(double x, double y);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <fmtmsg.h>
int fmtmsg(long classification, const char *label,
           int severity, const char *text,
           const char *action, const char *tag);
```

---

```
#define _XOPEN_SOURCE
#include <fnmatch.h>
int fnmatch(const char *pattern, const char *string,
            int flags);
```

---

```
#include <stdio.h>
FILE *fopen(const char *filename, const char *mode);
```

---

```
#define _POSIX_SOURCE
#include <sys/types.h>
#include <unistd.h>
pid_t fork(void);
```

---

```
#include <stdlib.h>
int fortrc(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
long fpathconf(int fildes, int varcode);
```

---

```
#include <stdio.h>
int fprintf(FILE *stream, const char *format-string,
           ...);
int printf(const char *format-string, ...);
int sprintf(char *buffer, const char *format-string,
           ...);
```

---

```
#include <stdio.h>
int fputc(int c, FILE *stream);
```

---

```
#include <stdio.h>
int fputs(const char *string, FILE *stream);
```

## Function Format

---

```
#include <stdio.h>
#include <wchar.h>
wint_t fputwc(wchar_t wc, FILE *stream);

#define _XOPEN_SOURCE
#include <stdio.h>
#include <wchar.h>
wint_t fputwc(wint_t wc, FILE *stream);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <stdio.h>
#include <wchar.h>
wint_t fputwc(wchar_t wc, FILE *stream);

#include <wchar.h>
int fputws(const wchar_t *wcs, FILE *stream);

#include <stdio.h>
size_t fread(void *buffer, size_t size,
             size_t count, FILE *stream);

#include <stdlib.h>
void free(void *ptr);

#include <stdio.h>
FILE *freopen(const char *filename, const char *mode,
              FILE *stream);

#include <math.h>
double frexp(double x, int *exp_ptr);

#include <stdio.h>
int fscanf (FILE *stream, const char *format-string,
           ...);
int scanf(const char *format-string, ...);
int sscanf(const char *buffer, const char *format,
           ...);

#include <stdio.h>
int fseek(FILE *stream, long int offset, int origin);

#include <stdio.h>
int fsetpos(FILE *stream, const fpos_t *pos);

#define _POSIX_SOURCE
#include <sys/stat.h>
int fstat(int fildes, struct stat *info);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/statvfs.h>
int fstatvfs(int fildes, struct statvfs *fsinfo);

#define _POSIX1_SOURCE 2
#include <unistd.h>
int fsync(int fildes);

#include <stdio.h>
long int ftell(FILE *stream);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/timeb.h>
int ftime(struct timeb *tp);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/ipc.h>
key_t ftok(const char *path, int id);
```

## Function Format

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int ftruncate(int fildev, off_t length);

#define _XOPEN_SOURCE
#include <ftw.h>
int ftw(const char *path, int (*fn)
        (const char *, const struct stat *, int),
        int ndirs);

#include <stdio.h>
size_t fupdate(const void *buffer, size_t size,
               FILE *stream);

#include <stdio.h>
size_t fwrite(const void *buffer, size_t size,
              size_t count, FILE *stream);

#include <math.h>
double gamma(double x);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *gcvt(double x, int ndigit, char *buf);

#define _ALL_SOURCE_NO_THREADS
#include <stdio.h>
int getc(FILE *stream);
int getchar(void);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
#include <sys/types.h>
int getclientid(int domain,
                struct clientid *clientid);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
#include <sys/types.h>
int __getclientid(int domain,
                  struct clientid *clientid);

#include <ucontext.h>
int getcontext(ucontext_t *ucp);

#define _POSIX_SOURCE
#include <unistd.h>
char *getcwd(char *buffer, size_t size);

#define _XOPEN_SOURCE_EXTENDED 1
#include <time.h>
struct tm *getdate(const char *string);
extern int getdate_err;

#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int getdtablesize(void);

#define _POSIX_SOURCE
#include <unistd.h>
gid_t getegid(void);

#include <stdlib.h>
char *getenv(const char *varname);
```

## Function Format

---

```
#include <stdlib.h>
char *__getenv(const char *varname);

#define _POSIX_SOURCE
#include <unistd.h>
uid_t geteuid(void);

#define _POSIX_SOURCE
#include <unistd.h>
gid_t getgid(void);

#define _XOPEN_SOURCE_EXTENDED 1
#include <grp.h>
void endgrent (void);
struct group *getgrent (void);
void setgrent (void);

#define _POSIX_SOURCE
#include <grp.h>
struct group *getgrgid(gid_t gid);

#define _POSIX_SOURCE
#include <grp.h>
struct group *getgrnam(const char *name);

#define _POSIX_SOURCE
#include <unistd.h>
int getgroups(int size, gid_t list[ ]);

#define _POSIX_SOURCE
#include <unistd.h>
int getgroupsbyname(char username[ ], int size,
                    gid_t list[ ]);

#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
extern int h_errno;
struct hostent *gethostbyaddr(const void *address,
                              size_t len,
                              int type);

#define _OE_SOCKETS
#include <netdb.h>
struct hostent *gethostbyaddr(char *address,
                              int address_len,
                              int domain);

#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
extern int h_errno;
struct hostent *gethostbyname(const char *name);

#define _OE_SOCKETS
#include <netdb.h>
struct hostent *gethostbyname(char *name);

#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct hostent *gethostent(void);

#define _OE_SOCKETS
#include <netdb.h>
struct hostent *gethostent(void);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
long gethostid(void);

#define _OE_SOCKETS
#include <unistd.h>
int gethostid();

#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int gethostname(char *name, size_t namelen);

#define _OE_SOCKETS
#include <unistd.h>
int gethostname(char *name, int namelen);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int getibmopt(int cmd, struct ibm_gettcpinfo *bfrp);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int getibmsockopt(int s, int level, int optname,
                  char *optval, int *optlen);

#define _XOPEN_SOURCE
#include <sys/_getip.h>
int __getipc(int token_id, IPCQPROC *bufptr,
             size_t buflen, int cmd);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/time.h>
int getitimer(int which, struct itimerval *value);

#define _POSIX_SOURCE
#include <unistd.h>
char *getlogin(void);

#define _XOPEN_SOURCE
#include <unistd.h>
char *getlogin(void);

#define _POSIX_SOURCE
#include <unistd.h>
char *__getlogin1(void);

#include <collate.h>
collel_t getmccoll(char **src);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int getmsg(int fildes, struct strbuf *ctlptr,
           struct strbuf *dataptr, int *flagsp);

#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct netent *getnetbyaddr(ip_addr_t net, int type);

#define _OE_SOCKETS
#include <sys/types.h>
#include <netdb.h>
struct netent *getnetbyaddr(unsigned long net,
                             int type);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct netent *getnetbyname(const char *name);

#define _OE_SOCKETS
#include <netdb.h>
struct netent *getnetbyname(name);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct netent *getnetent(void);

#define _OE_SOCKETS
#include <netdb.h>
struct netent *getnetent(void);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
int getopt(int argc, char * const argv[],
           const char *optstring);
extern char *optarg;
extern int optind, opterr, optopt;
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int getpagesize(void);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
char *getpass(const char *prompt);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int getpeername(int socket, struct sockaddr *address,
                size_t *address_len);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int getpeername(int socket, struct sockaddr *name,
                int namelen);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
pid_t getpgid(pid_t pid);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
pid_t getpgrp(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
pid_t getpid(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int getpmsg(int fildes, struct strbuf *ctlptr,
            struct strbuf *dataptr, int *bandp,
            int *flagsp);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
pid_t getppid(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/resource.h>
int getpriority(int which, id_t who);
```



## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct protoent *getprotobyname(const char *name);
```

```
#define _OE_SOCKETS
#include <netdb.h>
struct protoent *getprotobyname(char name);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct protoent *getprotobynumber(int proto);
#define OE_SOCKETS
#include <netdb.h>
struct protoent *getprotobynumber(int proto);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct protoent *getprotoent(void);
#define _OE_SOCKETS
#include <netdb.h>
struct protoent *getprotoent(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <pwd.h>
void endpwent (void);
struct passwd *getpwent(void);
void setpwent (void);
```

---

```
#define _POSIX_SOURCE
#include <pwd.h>
struct passwd *getpwnam(const char *name);
```

---

```
#define _POSIX_SOURCE
#include <pwd.h>
struct passwd *getpwuid(uid_t uid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/resource.h>
int getrlimit(int resource, struct rlimit *rlp);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/resource.h>
int getrusage(int who, struct rusage *r_usage);
```

---

```
#include <stdio.h>
char *gets(char *buffer);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct servent *getservbyname(const char *name,
                               const char *proto);
```

```
#define _OE_SOCKETS
#include <netdb.h>
struct servent *getservbyname(char name, char proto);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct servent *getservbyport(int port,
                               const char *proto);
```

```
#define _OE_SOCKETS
#include <netdb.h>
struct servent *getservbyport(int port,
                               char *proto);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
struct servent *getservent(void);

#define _OE_SOCKETS
#include <netdb.h>
struct servent *getservent(void);

#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
pid_t getsid(pid_t pid);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int getsockname(int socket,
                 struct sockaddr *address,
                 size_t *address_len);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int getsockname(int socket, struct sockaddr name,
                 int namelen);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int getsockopt(int socket,
               int level,
               int option_name,
               void *option_value,
               size_t *option_len);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int getsockopt(int socket, int level,
               int option_name, char *option_value,
               int *option_len);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int getstablesz(void);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
int getsubopt(char **optionp, char * const * tokens,
              char **valuep);

#include <variant.h>
struct variant *getsyntax(void);

#define _XOPEN_SOURCE_EXTENDED 1
#undef _ALL_SOURCE
#include <sys/time.h>
int gettimeofday(struct timeval *tp, void *tzp);
#define _ALL_SOURCE
#include <sys/time.h>
int gettimeofday(struct timeval *tp,
                 struct timezone *tzp);

#define _POSIX_SOURCE
#include <unistd.h>
uid_t getuid(void);

#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
struct utmpx *getutxent(void);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
struct utmpx *getutxid(const struct utmpx *id);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
struct utmpx *getutxline(const struct utmpx *line);
```

---

```
#define _XOPEN_SOURCE
#include <stdio.h>
int getw(FILE *stream);
```

---

```
#include <wchar.h>
wint_t getwc(FILE *stream);
```

---

```
#include <wchar.h>
wint_t getwchar(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
char *getwd(char *path_name);
```

---

```
#include <collate.h>
collel_t getwmccoll(wchar_t **src);
```

---

```
#define _OPEN_SYS_SOCKET_EXT
#include <sys/socket.h>
int givesocket(int d, struct clientid *clientid);
```

---

```
#define _XOPEN_SOURCE
#include <glob.h>
int glob(const char *pattern, int flags,
         int (*errfunc)(const char *epath,
                        int eerrno), glob_t *pglob);
```

---

```
#define _XOPEN_SOURCE
#include <glob.h>
void globfree(glob_t *pglob);
```

---

```
#include <time.h>
struct tm *gmtime(const time_t *timer);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
int grantpt(int fildes);
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
int hcreate(size_t nel);
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
void hdestroy(void);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
int __heaprpt(hreport_t *heap_report_structure);
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
ENTRY *hsearch(ENTRY item, ACTION action);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t htonl (in_addr_t hostlong);

#define _OE_SOCKETS
#include <sys/types.h>
#include <arpa/inet.h>
#include <netinet/in.h>
unsigned long htonl(unsigned long a);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_port_t htons(in_port_t hostshort);

#define _OE_SOCKETS
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned short htons(unsigned short a);
```

---

```
#include <math.h>
double hypot(double side1, double side2);
```

---

```
#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int ibmsflush(int s);
```

---

```
#include <iconv.h>
size_t iconv(iconv_t cd, char **inbuf,
             size_t *inbytesleft, char **outbuf,
             size_t *outbytesleft);
```

---

```
#include <iconv.h>
int iconv_close(iconv_t cd);
```

---

```
#include <iconv.h>
iconv_t iconv_open(const char *tocode,
                  const char *fromcode);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
int ilogb(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
char *index(const char *string, int c);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t inet_addr(const char *cp);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t inet_lnaof(struct in_addr in);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned long inet_lnaof(struct in_addr in);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
struct in_addr inet_makeaddr(in_addr_t net,
                             in_addr_t lna);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
struct in_addr inet_makeaddr(unsigned long net,
                             unsigned long lna);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t inet_netof(struct in_addr in);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned long inet_netof(struct in_addr in);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t inet_network(const char *cp);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned long inet_network(char cp);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
char *inet_ntoa(struct in_addr in);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
char *inet_ntoa(struct in_addr in);
```

---

```
#define _OPEN_SYS
#include <sys/types.h>
#include <grp.h>
int initgroups(const char *user,
              const gid_t basegid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *initstate(unsigned seed, char *state,
               size_t size);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <search.h>
void insque(void *element, void *pred);
```

## Function Format

---

```
/** Terminals */
#include <sys/ioctl.h>
int ioctl(int fildev int cmd, ... /* arg */);

/** Sockets */
#define _XOPEN_SOURCE_EXTENDED 1

/** OR */
#define _OE_SOCKETS

#include <sys/ioctl.h>
#include <net/rtroute.h>
#include <net/if.h>
int ioctl(int fildev, int cmd, ... /* arg */);

/** STREAMS */
#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int ioctl(int fildev int cmd, ... /* arg */);

#include <__ftp.h>
struct __ipdbcss *__ipdbcs(void);

#include <__ftp.h>
char *__ipdspx(void);

#include <__ftp.h>
char *__iphost(void);

#include <__ftp.h>
int __ipmsgc(void);

#include <__ftp.h>
char *__ipnode(void);

#include <__ftp.h>
char *__iptcpn(void);

#include <ctype.h>
int isalnum(int c);
int isalpha(int c);
int isblank(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);

#define _XOPEN_SOURCE
#include <ctype.h>
int isascii(int c);

#define _ALL_SOURCE
#include <ctype.h>
int isascii(int c);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int isastream(int fildev);

#define _POSIX_SOURCE
#include <unistd.h>
int isatty(int fildev);
```

## Function Format

---

```
#include <cics.h>
int iscics(void);
```

---

```
#include <collate.h>
int ismccollet(collet_t c);
```

---

```
#define _XOPEN_SOURCE
#include <math.h>
int isnan( double x);
```

---

```
#include <unistd.h>
int __isPosixOn(void);
```

---

```
#include <wctype.h>
int iswalnum(wint_t wc);
int iswalpha(wint_t wc);
int iswblank(wint_t wc);
int iswcntrl(wint_t wc);
int iswctype(wint_t wc, wctype_t wc_prop);
int iswdigit(wint_t wc);
int iswgraph(wint_t wc);
int iswlower(wint_t wc);
int iswprint(wint_t wc);
int iswpunct(wint_t wc);
int iswspace(wint_t wc);
int iswupper(wint_t wc);
int iswxdigit(wint_t wc);
```

---

```
#include <sys/_wlm.h.h>
int __JoinWorkUnit(wlmetok_t *enclavetoken);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
long int jrand48(unsigned short int x16v[3]);
```

---

```
#include <math.h>
double j0(double x);
double j1(double x);
double jn(int n, double x);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int kill(pid_t pid, int sig);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int killpg(pid_t pgrp, int sig);
```

---

```
#include <stdlib.h>
long int labs(long int n);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int lchown(const char *path, uid_t owner,
           gid_t group);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
void lcong48(unsigned short int param[7]);
```

---

```
#include <math.h>
double ldexp(double x, int exp);
```

---

```
#include <stdlib.h>
ldiv_t ldiv(long int numerator,
            long int denominator);
```

## Function Format

---

```
#include <sys/_wlm.h>
int __LeaveWorkUnit(wlmetok_t *enclavetoken);
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
void *lfind(const void *key, const void *base,
            size_t *nel, size_t width, int (*compar)
            (const void *, const void *));
```

---

```
#define _XOPEN_SOURCE
#include <math.h>
double lgamma(double x);
extern int signgam;
int *__signgam(void);
```

---

```
#include <stdlib.h>
int __librel(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int link(const char *oldfile, const char *newname);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int listen(int socket, int backlog);

#define _OE_SOCKETS
#include <sys/socket.h>
int listen(int socket, int backlog);
```

---

```
#include <locale.h>
struct dtconv *localdtconv(void);
```

---

```
#include <locale.h>
struct lconv *localeconv(void);
```

---

```
#include <time.h>
struct tm *localtime(const time_t *timeval);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int lockf(int fildes, int function, off_t size);
```

---

```
#include <math.h>
double log(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double logb(double x);
```

---

```
#define _OPEN_SYS
#include <unistd.h>
int __login(int function_code,
            int identity_type,
            int identity_length,
            void *identity,
            int pass_length,
            char *pass,
            int certificate_length,
            char *certificate,
            int option_flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double loglp(double x);
```



## Function Format

---

```
#include <math.h>
double log10(double x);

#include <setjmp.h>
void longjmp(jmp_buf env, int value);

#define _XOPEN_SOURCE_EXTENDED 1
#include <setjmp.h>
void _longjmp(jmp_buf env, int value);

#define _XOPEN_SOURCE
#include <stdlib.h>
long int lrand48(void);

#define _XOPEN_SOURCE
#include <search.h>
void *lsearch(const void *key, void *base,
              size_t *nel, size_t width,
              int (*compar) (const void *,
                             const void *));

#define _POSIX_SOURCE
#include <unistd.h>
off_t lseek(int fildes, off_t offset, int pos);

#define _POSIX1_SOURCE 2
#include <sys/stat.h>
int lstat(const char *pathname, struct stat *buf);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *l64a (long value);

#include <ucontext.h>
void makecontext(ucontext_t *ucp, void (*func) (),
                 int argc, ...);

#include <stdlib.h>
void *malloc(size_t size);

#include <collate.h>
coll_t maxcoll(void);

#define _OPEN_SYS_SOCKET_EXT
#include <sys/types.h>
#include <sys/socket.h>
int maxdesc(int *totdesc, int *inetdesc);

#include <stdlib.h>
int mblen(const char *string, size_t n);

#include <wchar.h>
size_t mbrlen (const char *s, size_t n,
               mbstate_t *ps);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
size_t mbrlen (const char *s, size_t n,
               mbstate_t *ps);
```

## Function Format

---

```
#include <wchar.h>
size_t mbrtowc (wchar_t *pwc, const char *s,
               size_t n, mbstate_t *ps);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
size_t mbrtowc (wchar_t *pwc, const char *s,
               size_t n, mbstate_t *ps);

#include <wchar.h>
int mbsinit( const mbstate_t *ps );

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int mbsinit( const mbstate_t *ps );

#include <wchar.h>
size_t mbsrtowcs(wchar_t *dst, const char **src,
                 size_t len, mbstate_t *ps);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
size_t mbsrtowcs(wchar_t *dst, const char **src,
                 size_t len, mbstate_t *ps);

#include <stdlib.h>
size_t mbstowcs(wchar_t *pwc, const char *string,
               size_t n);

#include <stdlib.h>
int mbtowc(wchar_t *pwc, const char *string,
           size_t n);

#define _XOPEN_SOURCE
#include <string.h>
void * memccpy(void *s1, const void *s2, int c,
               size_t n);

#include <string.h>
void *memchr(const void *buf, int c, size_t count);

#include <string.h>
int memcmp(const void *buf1, const void *buf2,
           size_t count);

#include <string.h>
void *memcpy(void *dest, const void *src,
             size_t count);

#include <string.h>
void *memmove(void *dest, const void *src,
              size_t count);

#include <string.h>
void *memset(void *dest, int c, size_t count);

#define _POSIX_SOURCE
#include <sys/stat.h>
int mkdir(char *pathname, mode_t mode);

#define _POSIX_SOURCE
#include <sys/stat.h>
int mkfifo(const char *pathname, mode_t mode);
```

## Function Format

```
#define _OPEN_SYS
#include <sys/stat.h>
int mknod(const char *path, mode_t mode,
          rdev_t dev_identifler);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/stat.h>
int mknod(const char *path, mode_t mode,
          dev_t dev_identifler);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
int mkstemp(char *template);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char * mktemp(char *template);

#include <time.h>
time_t mktime(struct tm *tmpr);

#define _OPEN_SOURCE_EXTENDED 2
#include <sys/mman.h>
int __mlockall(int flags);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/mman.h>
void *mmap(void *addr, size_t len, int prot,
           int flags, int fildes, off_t off);

#include <math.h>
double modf(double x, double *intptr);

#include <sys/stat.h>
int mount(const char *path, char *filesystem,
          char *filesystype, mtm_t mtm,
          int parmlen, char *parm);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/mman.h>
int mprotect(void *addr, size_t len, int prot);

#define _XOPEN_SOURCE
#include <stdlib.h>
long int mrnd48(void);

#define _XOPEN_SOURCE
#include <sys/msg.h>
int msgctl(int msgid, int cmd,
           struct msqid_ds *buf);

#define _XOPEN_SOURCE
#include <sys/msg.h>
int msgget(key_t key, int msgflg);

#define _XOPEN_SOURCE
#include <sys/msg.h>
int msgrcv(int msgid, void *msgp, size_t msgsz,
           long int msgtyp, int msgflg);

#define _XOPEN_SOURCE
#include <sys/msg.h>
int msgsnd(int msgid, const void *msgp,
           size_t msgsz, int msgflg);
```

## Function Format

---

```
#define _OPEN_SYS_IPC_EXTENSIONS
#include <sys/msg.h>
int msgxrcv(int msgid, void *msgp, size_t msgsz,
            long int msgtyp, int msgflg);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/mman.h>
int msync(void *addr, size_t len, int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/mman.h>
int munmap(void *addr, size_t len);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double nextafter(double x, double y);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <ftw.h>
int nftw(const char *path,
         int (*fn)(const char *, const struct stat *,
                   int, struct FTW *), int ndirs, int flags);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
int nice(int increment);
```

---

```
#include <nlist.h>
int nlist (const char * loadname, struct nlist * np);
```

---

```
#include <langinfo.h>
char *nl_langinfo(nl_item item);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
long int nrand48(unsigned short int x16v[3]);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_addr_t ntohl(in_addr_t netlong);

#define _OE_SOCKETS
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned long ntohl(unsigned long a);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <arpa/inet.h>
in_port_t ntohs(in_port_t netshort);

#define _OE_SOCKETS
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned short ntohs(unsigned short a);
```

---

```
#define _POSIX_SOURCE
#include <fcntl.h>
int open(const char *pathname, int options, ...);
```

---

```
#define _POSIX_SOURCE
#include <dirent.h>
DIR *opendir(const char *dirname);
```

## Function Format

```
#define _OPEN_SYS_DIR_EXT
#include <dirent.h>
DIR *__opendir2(const char *dirname, size_t bufsize);
```

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <syslog.h>
void openlog(const char *ident, int logopt,
             int facility);
```

```
#include <unistd.h>
int __openMvsRel(void);
```

```
#define _POSIX_SOURCE
#include <fcntl.h>
#include <sys/stat.h>
int __open_stat(const char *pathname, int options,
               mode_t mode, struct stat *info);
```

```
int __passwd(const char *username,
             const char *oldpass,
             const char *newpass);
```

```
#define _POSIX_SOURCE
#include <unistd.h>
long pathconf(const char *pathname, int varcode);
```

```
#define _POSIX_SOURCE
#include <unistd.h>
int pause(void);
```

```
#define _XOPEN_SOURCE
#include <stdio.h>
int pclose(FILE *stream);
```

```
#include <stdio.h>
void perror(const char *string);
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdio.h>
void perror(const char *string);
#define _OE_SOCKETS
#include <stdio.h>
void perror(const char *string);
```

```
#define _OPEN_SYS
#include <unistd.h>
int __pid_affinity(int function_code,
                  pid_t target_pid,
                  pid_t signal_pid,
                  int signal);
```

```
#define _POSIX_SOURCE
#include <unistd.h>
int pipe(int fdinfo[2]);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <poll.h>
int poll(struct pollfd fds[], nfds_t nmsgsfds,
         int timeout);

#define _XOPEN_SOURCE_EXTENDED 1
#define _OPEN_MSGQ_EXT
#include <sys/types.h>
#include <sys/time.h>
#include <sys/msg.h>
#include <poll.h>
int poll(void *listptr, nmsgsfds_t nmsgsfds,
         int timeout);

#define _XOPEN_SOURCE
#include <stdio.h>
FILE *popen(const char *command, const char *mode);

#include <math.h>
double pow(double x, double y);

#include <stdio.h>
int printf(const char *format-string, ...);

#define
_OPEN_THREADS
#include <pthread.h>
int pthread_attr_destroy(pthread_attr_t *attr);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_getdetachstate(pthread_attr_t *
                               attr);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_getstacksize(pthread_attr_t *attr,
                              size_t *stacksize);

#define _OPEN_SYS
#include <pthread.h>
int pthread_attr_getsynctype_np(pthread_attr_t *
                                attr);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_getweight_np(pthread_attr_t *attr);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_init(pthread_attr_t *attr);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_setdetachstate(pthread_attr_t *attr,
                               int *detachstate);

#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_setstacksize(pthread_attr_t *attr,
                              size_t stacksize);

#define _OPEN_SYS
#include <pthread.h>
int pthread_attr_setsynctype_np(pthread_attr_t *attr,
                                int synctype);
```

## Function Format

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_attr_setweight_np(pthread_attr_t *attr,
                             int threadweight);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cancel(pthread_t thread);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
void pthread_cleanup_pop(int execute);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
void pthread_cleanup_push(void (*routine) (void *arg),
                          void *arg);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_broadcast(pthread_cond_t *cond);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_destroy(pthread_cond_t *cond);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_init(pthread_cond_t *cond,
                     pthread_condattr_t *attr);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_signal(pthread_cond_t *cond);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_timedwait(pthread_cond_t *cond,
                           pthread_mutex_t *mutex,
                           const struct timespec
                           *abstime);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_cond_wait(pthread_cond_t *cond,
                     pthread_mutex_t *mutex);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_condattr_destroy
(pthread_condattr_t *attr);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_condattr_getkind_np
(pthread_condattr_t *attr, int *kind);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_condattr_init(pthread_condattr_t *attr);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_condattr_setkind_np
(pthread_condattr_t *attr, int *kind);
```

---

## Function Format

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_create(pthread_t *thread,
    pthread_attr_t *attr,
    void * (*start_routine) (void *arg), void *arg);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_detach(pthread_t *thread);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_equal(pthread_t t1, pthread_t t2);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
void pthread_exit(void *status);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_getspecific(pthread_key_t key,
    void **value);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
void *pthread_getspecific_d8_np(pthread_key_t key);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_join(pthread_t thread, void **status);
```

---

```
#define _OPEN_SYS
#include <pthread.h>
int pthread_join_d4_np(pthread_t thread,
    void **status);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_key_create(pthread_key_t *key,
    void (*destructor) (void *));
```

---

```
#define _OPEN_THREADS
#include <signal.h>
int pthread_kill(pthread_t thread, int sig);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutex_destroy(pthread_mutex_t *mutex);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutex_init(pthread_mutex_t *mutex,
    pthread_mutexattr_t *attr);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutex_lock(pthread_mutex_t *mutex);
```

---

```
#define
_OPEN_THREADS
#include <pthread.h>
int pthread_mutex_trylock(pthread_mutex_t *mutex);
```

---

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

---



## Function Format

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutexattr_destroy(pthread_mutexattr_t
                             *attr);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutexattr_getkind_np(pthread_mutexattr_t
                                 *attr, int *kind);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutexattr_init(pthread_mutexattr_t
                           *attr);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_mutexattr_setkind_np(pthread_mutexattr_t
                                 *attr, int kind);
```

```
#define _OPEN_THREADS
#include <pthread.h>
pthread_once_t once_control = PTHREAD_ONCE_INIT;
int pthread_once(pthread_once_t *once_control,
                 void(*init_routine) ());
```

```
#define _OPEN_SYS 1
#include <pthread.h>
int pthread_security_np(int function_code,
                       int identity_type,
                       size_t identity_length,
                       void *identity,
                       char *password,
                       int options);
```

```
#define _OPEN_THREADS
#include <pthread.h>
pthread_t pthread_self(void);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_setintr(int state);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_setintrtype(int type);
```

```
#define _OPEN_SYS
#include <pthread.h>
int pthread_set_limit_np(int action,
                        int maxthreadtasks,
                        int maxthreads);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_setspecific(pthread_key_t key,
                       void *value);
```

```
#define _OPEN_THREADS
#include <pthread.h>
int pthread_tag_np(const char * newtag,
                  char * oldtag);
```

## Function Format

---

```
#define
_OPEN_THREADS
#include <pthread.h>
void pthread_testintr(void);

#define _OPEN_THREADS
#include <pthread.h>
void pthread_yield(NULL);

#include <stdlib.h>
char *ptsname(int fildes);

#define _ALL_SOURCE_NO_THREADS
#include <stdio.h>
int putc(int c, FILE *stream);
int putchar(int c);

#define _XOPEN_SOURCE
#include <stdlib.h>
int putenv(const char *envvar);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stropts.h>
int putmsg(int fildes, const struct strbuf *ctlptr,
           const struct strbuf *dataptr, int flags);
int putpmsg(int fildes, const struct strbuf *ctlptr,
            const struct strbuf *dataptr, int band,
            int flags);

#include <stdio.h>
int puts(const char *string);

#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
struct utmpx *pututxline(const struct utmpx *utmpx);

#define _XOPEN_SOURCE
#include <stdio.h>
int putw(int w, FILE *stream);

#include <stdio.h>
#include <wchar.h>
wint_t putwc(wchar_t wc, FILE *stream);

#define _XOPEN_SOURCE
#include <stdio.h>
#include <wchar.h>
wint_t putwc(wint_t wc, FILE *stream);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <stdio.h>
#include <wchar.h>
wint_t putwc(wchar_t wc, FILE *stream);
```

## Function Format

---

```
#include <stdio.h>
#include <wchar.h>
wint_t putwchar(wchar_t wc);

#define _XOPEN_SOURCE
#include <stdio.h>
#include <wchar.h>
wint_t putwchar(wint_t wc);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <stdio.h>
#include <wchar.h>
wint_t putwchar(wchar_t wc);

#include <stdlib.h>
void qsort(void *base, size_t num, size_t width,
           int(*compare)(const void *element1,
                        const void *element2));

#include <sys/_wlm.h.h>
int __QueryMetrics(struct sysi *sysi_ptr,
                  int *anslen);

#include <sys/_wlm.h.h>
int __QuerySchEnv(struct sysi *sysi_ptr,
                  int *anslen);

#include <signal.h>
int raise(int sig);

#include <stdlib.h>
int rand(void);

#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
long random(void);

#define _POSIX_SOURCE
#include <unistd.h>
ssize_t read(int fildes, void *buf, size_t N);

#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
ssize_t read(int socket, void *buf, ssize_t len);

#define _OE_SOCKETS
#include <unistd.h>
ssize_t read(int socket, void *buf, ssize_t len);

#define _POSIX_SOURCE
#include <dirent.h>
struct dirent *readdir(DIR *dir);

#define _OPEN_SYS_DIR_EXT
#include <dirent.h>
struct dirent *__readdir2(DIR *dir,
                          struct stat *info);

#define _POSIX1_SOURCE 2
#include <unistd.h>
int readlink(const char *path, char *buf,
             size_t bufsiz);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/uio.h>
ssize_t readv(int fildev, const struct iovec *iov,
              int iovcnt);
```

```
#define _OE_SOCKETS
#include <sys/uio.h>
int readv(int socket, struct iovec *iov, int
          iovcnt);
```

---

```
#include <stdlib.h>
void *realloc(void *ptr, size_t size);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *realpath(const char *file_name,
               char *resolved_name);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <re_comp.h>
char *re_comp(const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
ssize_t recv(int socket, void *buffer,
              size_t length, int flags);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int recv(int socket, char *buffer,
          int length, int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int recvfrom(int socket, void *buffer,
              size_t length, int flags,
              struct sockaddr *address,
              size_t *address_length);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int recvfrom(int socket, char *buffer,
              int length, int flags,
              struct sockaddr *address,
              int *address_length);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
ssize_t recvmsg(int socket, struct msghdr *message,
                 int flags);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int recvmsg(int socket, struct msghdr *message,
             int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <re_comp.h>
int re_exec(const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <libgen.h>
char *regcmp(const char *pattern[...], (char *)0);
```

---

```
#include <regex.h>
int regcomp(regex_t *preg, const char *pattern,
             int cflags);
```

## Function Format

---

```
#include <regex.h>
size_t regerror(int errcode, const regex_t *preg,
               char *errbuf, size_t errbuf_size);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <libgen.h>
char *regex(const char *cmppat,
            const char *subject[,subexp,...]);
extern char *__loc1;
```

---

```
#include <regex.h>
int regexec(const regex_t *preg, const char *string,
            size_t nmatch, regmatch_t *pmatch,
            int eflags);
```

---

```
#include <regex.h>
void regfree(regex_t *preg);
```

---

```
#include <stdlib.h>
int release(void(*fetch_ptr)());7
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double remainder(double x, double y);
```

---

```
#include <stdio.h>
int remove(const char *filename);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <search.h>
void remque(void *element);
```

---

```
#include <stdio.h>
int rename(const char *oldname, const char *newname);
```

---

```
#include <stdio.h>
void rewind(FILE *stream);
```

---

```
#define POSIX_SOURCE
#include <dirent.h>
void rewinddir(DIR *dir);
```

---

```
#include <rexec.h>
int rexec (char **host, int port, char *user,
           char *password, char* command,
           int *errFileDescParam);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
char *rindex(const char *string, int c);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double rint(double x);
```

---

```
#define POSIX_SOURCE
#include <unistd.h>
int rmdir(const char *pathname);
```

---

```
#include <stdlib.h>
int rpmatch(const char *response);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
void *sbrk(int incr);
```

---

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <math.h>
double scalb(double x, double n);

#include <stdio.h>
int scanf(const char *format-string, ...);

#define _XOPEN_SOURCE
#include <stdlib.h>
unsigned short int *seed48(unsigned short
                           int seed16v[3]);

#define _XOPEN_SOURCE
#include <dirent.h>
void seekdir(DIR *dirp, long int loc);

#define _XOPEN_SOURCE_EXTENDED 1
#define _OPEN_MSGQ_EXT
#include <sys/types.h>
#include <sys/time.h>
#include <sys/msg.h>
int select(int nmsgsfds, fd_set *readlist,
           fd_set *writelist, fd_set *exceptlist,
           struct timeval *timeout);

#define _OE_SOCKETS
#define _OPEN_MSGQ_EXT
#include <sys/types.h>
#include <sys/time.h>
#include <sys/msg.h>
int select(int nmsgsfds, fd_set *readlist,
           fd_set *writelist, fd_set *exceptlist,
           struct timeval *timeout);

#define _XOPEN_SOURCE_EXTENDED 1
#define _ALL_SOURCE
#define _OPEN_MSGQ_EXT
#include <sys/types.h>
#include <sys/time.h>
#include <sys/msg.h>
int selectex(int nmsgsfds, fd_set *readlist,
             fd_set *writelist, fd_set *exceptlist,
             struct timeval *timeout, int *ecbptr);

#define _OE_SOCKETS
#define _ALL_SOURCE
#define _OPEN_MSGQ_EXT
#include <sys/types.h>
#include <sys/time.h>
#include <sys/msg.h>
int selectex(int nmsgsfds, fd_set *readlist,
             fd_set *writelist, fd_set *exceptlist,
             struct timeval *timeout, int *ecbptr);

#define _XOPEN_SOURCE
#include <sys/sem.h>
int semctl(int semid, int semnum, int cmd, ...);

#define _XOPEN_SOURCE
#include <sys/sem.h>
int semget(key_t key, int nsems, int semflg);

#define _XOPEN_SOURCE
#include <sys/sem.h>
int semop(int semid, struct sembuf *sops,
          size_t nsops);
```

---

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
ssize_t send(int socket, const void *buffer,
             size_t length, int flags);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int send(int socket, char *buffer,
         int length, int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
ssize_t sendmsg(int socket, struct msghdr *message,
               int flags);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int sendmsg(int socket, struct msghdr *message,
            int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
ssize_t sendto(int socket, const void *buffer,
               size_t length, int flags,
               const struct sockaddr *address,
               size_t *address_length);
```

```
#define _OE_SOCKETS
#include <sys/socket.h>
int sendto(int socket, char *buffer,
           int length, int flags,
           struct sockaddr *address,
           int address_len);
```

---

```
#include <sys/server.h>
int __server_classify(server_classify_t handle,
                     server_classify_field_t field,
                     const char *value);
```

---

```
#include <sys/server.h>
server_classify_t __server_classify_create(
    server_classify_t handle,
    server_classify_field_t field,
    const char *value);
```

---

```
#include <sys/server.h>
void __server_classify_destroy(
    server_classify_t area);
```

---

```
#include <sys/server.h>
void __server_classify_reset(server_classify_t area);
```

---

```
#include <sys/server.h>
int __server_init(int *managertype,
                  const char *subsystem,
                  const char *subsysname,
                  const char *applenv,
                  int paralleu);
```

## Function Format

---

```
#include <sys/server.h>
int __server_pwu(int fcncode,
                 const char *transclass,
                 const char *applenv,
                 int classifylen,
                 const void *classify,
                 int *apldataalen,
                 void **apldata,
                 struct srv_fd_list **fdlstruc);
```

---

```
#include <stdio.h>
void setbuf(FILE *stream, char *buffer);
```

---

```
#include <ucontext.h>
int setcontext(const ucontext_t *ucp);
```

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int setegid(gid_t gid);
```

---

```
#define _POSIX1_SOURCE 2
#include <stdlib.h>
int setenv(const char *var_name,
           const char *new_value,
           int change_flag);
```

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int seteuid(uid_t uid);
```

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int setgid(gid_t gid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <grp.h>
void endgrent (void),
struct group *getgrent (void);
void setgrent (void);
```

---

```
#define _OPEN_SYS
#include <sys/types.h>
#include <grp.h>
int setgroups(const int size, const gid_t list[ ]);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void sethostent(int stayopen);
#define _OE_SOCKETS
#include <netdb.h>
void sethostent(int stayopen);
```

---

```
#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int setibmopt(int cmd, struct ibm_tcpimage *bfrp);
```

---

```
#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
int setibmsockopt(int s,int level,int optname,
                  char *optval,int optlen);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/time.h>
int setitimer(int which, struct itimerval *value,
              struct itimerval *ovalue);
```

---



## Function Format

---

```
#include <setjmp.h>
int setjmp(jmp_buf env);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <setjmp.h>
int _setjmp(jmp_buf env);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
void setkey(const char *key);
```

---

```
#include <locale.h>
char *setlocale(int category, const char *locale);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <syslog.h>
int setlogmask(int maskpri);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void setnetent(int stayopen);

#define _OE_SOCKETS
#include <netdb.h>
void setnetent(int stayopen);
```

---

```
#include <new.h>
typedef void(*PNH)();
PNH set_new_handler(PNH);8
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int setpeer(int socket, struct sockaddr *address,
            int length, char *address);

#define _OE_SOCKETS
#include <sys/socket.h>
int setpeer(int socket, struct sockaddr *address,
            int length, char *address);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int setpgid(pid_t pid, pid_t pgid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
pid_t setpgrp(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/resource.h>
int setpriority(int which, id_t who, int priority);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void setprotoent(int stayopen);

#define _OE_SOCKETS
#include <netdb.h>
void setprotoent(int stayopen);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <pwd.h>
void endpwent (void),
struct passwd *getpwent(void);
void setpwent (void);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int setregid(gid_t rgid, gid_t egid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int setreuid(uid_t ruid, uid_t euid);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/resource.h>
int setrlimit(int resource, const struct rlimit *rlp);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <netdb.h>
void setservent(int stayopen);

#define _OE_SOCKETS
#include <netdb.h>
void setservent(int stayopen);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
pid_t setsid(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int setsockopt(int socket, int level,
               int option_name,
               const void *option_value,
               size_t option_length);

#define _OE_SOCKETS
#include <sys/types.h>
#include <sys/socket.h>
int setsockopt(int socket, int level,
               int option_name, char *option_value,
               int *option_length);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
char *setstate(const char *state);
```

---

```
#include <terminate.h>
typedef void(*PFV)();
PFV set_terminate(PFV);8
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int setuid(uid_t uid);
```

---

```
#include <unistd.h>
typedef void(*PFV)();
PFV set_unexpected(PFV);8
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
void setutxent(void);
```

---

```
#include <stdio.h>
int setvbuf(FILE *stream, char *buf, int type,
            size_t size);
```

---

```
#define _XOPEN_SOURCE
#include <sys/shm.h>
void *shmatt(int shmid, const void *shmaddr,
             int shmflg);
```

## Function Format

---

```
#define _XOPEN_SOURCE
#include <sys/shm.h>
int shmctl(int shmid, int cmd,
            struct shmids *buf);
```

---

```
#define _XOPEN_SOURCE
#include <sys/shm.h>
int shmdt(const void *shmaddr);
```

---

```
#define _XOPEN_SOURCE
#include <sys/shm.h>
int shmget(key_t key, size_t size, int shmflg);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
long shutdown(int socket, int how);

#define _OE_SOCKETS
#include <sys/socket.h>
long shutdown(int *s, int how);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigaction(int sig, const struct sigaction *new,
              struct sigaction *old);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int __sigactionset(size_t newct,
                  const __sigactionset_t new[],
                  size_t *oldct,
                  __sigactionset_t old[],
                  int options);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigaddset(sigset_t *set, int signal);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sigaltstack(const stack_t *ss, stack_t *oss);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigdelset(sigset_t *set, int signal);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigemptyset(sigset_t *set);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigfillset(sigset_t *set);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sighold(int sig);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sigignore(int sig);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int siginterrupt(int sig, int flag);
```

---

## Function Format

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigismember(const sigset_t *set, int signal);
```

---

```
#define _POSIX_SOURCE
#include <setjmp.h>
void siglongjmp(sigjmp_buf env, int val);
```

---

```
#include <signal.h>
void(*signal (int sig, void(*func)(int)))(int);
```

---

```
#define _XOPEN_SOURCE
#include <math.h>
int *__signgam(void);
#define signgam (*__signgam())
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sigpause(int sig);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigpending(sigset_t *set);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigprocmask(int option, const sigset_t *new_set,
                sigset_t *old_set);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sigrelse(int sig);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
void (*sigset(int sig, void (*disp)(int)))(int);
```

---

```
#define _POSIX_SOURCE
#include <setjmp.h>
int sigsetjmp(sigjmp_buf env, int savemask);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <signal.h>
int sigstack(struct sigstack *ss,
             struct sigstack *oss);
```

---

```
#define _POSIX_SOURCE
#include <signal.h>
int sigsuspend(const sigset_t *mask);
```

---

```
#define _OPEN_THREADS
#include <signal.h>
int sigwait(sigset_t *set);
```

---

```
#include <math.h>
double sin(double x);
```

---

```
#include <math.h>
double sinh(double x);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
unsigned int sleep(unsigned int seconds);
```

---

## Function Format

---

```
#include <sys/_wlm.h>
int __smf_record(int smf_record_type,
                 int smf_record_subtype,
                 int smf_record_length,
                 char *smf_record);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
void sock_debug(int onoff);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
void sock_debug_bulk_perf0(int onoff);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
void sock_do_bulkmode(int onoff);

#define _OPEN_SYS_SOCK_EXT
#include <sys/socket.h>
void sock_do_teststor(int onoff);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int socket(int domain, int type, int protocol);

#define _OE_SOCKETS
#include <sys/socket.h>
int socket(int *domain, int type, int protocol);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/socket.h>
int socketpair(int *domain, int type,
               int protocol,
               int socket_vector[2]);

#define _OE_SOCKETS
#include <sys/socket.h>
int socketpair(int *domain, int type,
               int protocol, int sv[2]);

#define _POSIX_SOURCE
#include <spawn.h>
pid_t spawn(const char *path, const int fd_count,
             const int fd_map[],
             const struct inheritance *inherit,
             const char *argv[],
             const char *envp[]);
pid_t spawnp(const char *file, const int fd_count,
              const int fd_map[],
              const struct inheritance *inherit,
              const char *argv[],
              const char *envp[]);

#define _POSIX_SOURCE
#include <spawn.h>
pid_t __spawn2(const char *path, const int fd_count,
               const int fd_map[],
               const struct inheritance *inherit,
               const char *argv[],
               const char *envp[]);
pid_t __spawnp2(const char *file, const int fd_count,
                 const int fd_map[],
                 const struct inheritance *inherit,
                 const char *argv[],
                 const char *envp[]);
```

## Function Format

---

```
#include <stdio.h>
int sprintf(char *buffer, const char *format-string,
...);
```

---

```
#include <math.h>
double sqrt(double x);
```

---

```
#include <stdlib.h>
void srand(unsigned int seed);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
void srandom(unsigned seed);
```

---

```
#define _XOPEN_SOURCE
#include <stdlib.h>
void srand48(long int seedval);
```

---

```
#include <stdio.h>
int sscanf(const char *buffer, const char *format,
...);
```

---

```
#define _POSIX_SOURCE
#include <sys/stat.h>
int stat(const char *pathname, struct stat *info);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/statvfs.h>
int statvfs(const char *pathname,
struct statvfs *fsinfo);
```

---

```
#define _XOPEN_SOURCE
#include <regex.h>
int step(const char *string, const char *expbuf);
extern char *loc1, *loc2;
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
int strcasecmp(const char *string1,
const char *string2);
```

---

```
#include <string.h>
char *strcat(char *string1,
const char *string2);
```

---

```
#include <string.h>
char *strchr(const char *string, int c);
```

---

```
#include <string.h>
int strcmp(const char *string1, const char *string2);
```

---

```
#include <string.h>
int strcoll(const char *string1, const char *string2);
```

---

```
#include <string.h>
char *strcpy(char *string1, const char *string2);
```

---

```
#include <string.h>
size_t strcspn(const char *string1,
const char *string2);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <string.h>
char *strdup(const char *string);
```

---

## Function Format

---

```
#include <string.h>
char *strerror(int errnum);
```

---

```
#include <monetary.h>
int strfmon(char *s, size_t maxsize,
            const char *format, ...);
```

---

```
#include <time.h>
size_t strftime(char *dest, size_t maxsize,
                const char *format,
                const struct tm *timeptr);
```

---

```
#include <string.h>
size_t strlen(const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <strings.h>
int strncasecmp(const char *string1,
                const char *string2, size_t n);
```

---

```
#include <string.h>
char *strncat(char *string1, const char *string2,
              size_t count);
```

---

```
#include <string.h>
int strncmp(const char *string1, const char *string2,
            size_t count);
```

---

```
#include <string.h>
char *strncpy(char *string1, const char *string2,
              size_t count);
```

---

```
#include <string.h>
char *strpbrk(const char *string1,
              const char *string2);
```

---

```
#include <time.h>
char *strptime(const char *buf, const char *fmt,
               struct tm *tm);
```

---

```
#include <string.h>
char *strchr(const char *string, int c);
```

---

```
#include <string.h>
size_t strspn(const char *string1,
              const char *string2);
```

---

```
#include <string.h>
char *strstr(const char *string1,
             const char *string2);
```

---

```
#include <collate.h>
collel_t strtocoll(char *s);
```

---

```
#include <stdlib.h>
double strtod(const char *nptr, char **endptr);
```

---

```
#include <string.h>
char *strtok(char *string1, const char *string2);
```

---

```
#include <stdlib.h>
long int strtol(const char *nptr, char **endptr,
                int base);
```

## Function Format

---

```
#include <stdlib.h>
unsigned long int strtoul(const char *string1,
                        char **string2, int base);
```

---

```
#include <string.h>
size_t strxfrm(char *s1, const char *s2, size_t n);
```

---

```
#include <stdio.h>
int svc99(__S99parms *string);
```

---

```
#define _XOPEN_SOURCE
#include <unistd.h>
void swab(const void *src, void *dest,
          ssize_t nbytes);
```

---

```
#include <ucontext.h>
int swapcontext(ucontext_t *oucp,
               const ucontext_t *ucp);
```

---

```
#include <wchar.h>
int swprintf(wchar_t *wcs, size_t n,
             const wchar_t *format, ...);
```

```
#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int swprintf(wchar_t *wcs, size_t n,
             const wchar_t *format, ...);
```

---

```
#include <wchar.h>
int swscanf(const wchar_t *wcs,
            const wchar_t *format, ...);
```

```
#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int swscanf(const wchar_t *wcs,
            const wchar_t *format, ...);
```

---

```
#define _POSIX1_SOURCE 2
#include <unistd.h>
int symlink(const char *pathname, const char *slink);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
void sync(void);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
long sysconf(int name);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <syslog.h>
int syslog(int priority, const char *message,
           ... /* argument */);
```

---

```
#include <stdlib.h>
int system(const char *string);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_accept(int fd, int resfd, struct t_call *call);
```

---

```
#define _OPEN_SYS_SOCKET_EXT
#include <sys/types.h>
#include <socket.h>
int takesocket(struct clientid *clientid, int sdesc);
```



## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
char *t_alloc(int fd, int struct_type, int fields);
```

---

```
#include <math.h>
double tan(double x);
```

---

```
#include <math.h>
double tanh(double x);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_bind(int fd, struct t_bind *req,
           struct t_bind *ret);
```

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcdrain(int filides);
```

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcflow(int filides, int action);
```

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcflush(int filides, int where);
```

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcgetattr(int filides, struct termios *termpptr);
```

---

```
#define _OPEN_SYS_PTY_EXTENSIONS
#include <termios.h>
int __tcgetcp(int filedes, int termcplen,
              struct __termcp *termcpptr);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
pid_t tcgetpgrp(int filides);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <termios.h>
pid_t tcgetsid(int filides);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_close(int fd);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_connect(int fd, struct t_call *call,
              struct t_call *rcvcall);
```

---

```
#define _OPEN_SYS_SOCKET_EXT
#include <sys/socket.h>
#include <stdio.h>
#include <errno.h>
void tcperror(char *s);
```

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcsendbreak(int filides, int duration);
```

## Function Format

---

```
#define _POSIX_SOURCE
#include <termios.h>
int tcsetattr(int fd, int when,
               const struct termios *termpr);

#define _OPEN_SYS_PTY_EXTENSIONS
#include <termios.h>
int __tcsetcp(int filedes, int termcplen,
               const struct __termcp *termcp);

#define _POSIX_SOURCE
#include <unistd.h>
int tcsetpgrp(int filedes, pid_t newid);

#define
_OPEN_SYS_PTY_EXTENSIONS
#include <termios.h>
int __tcsettables(int filedes, int termcplen,
                  const struct __termcp *termcp,
                  const char atoe[256],
                  const char etoe[256]);

#define _XOPEN_SOURCE
#include <search.h>
void *tdelete(const void *key, void **rootp,
              int (*compar)(const void *, const void *));

#define _XOPEN_SOURCE
#include <dirent.h>
long telldir(DIR *dirp);

#define _XOPEN_SOURCE
#include <stdio.h>
char *tempnam(const char *dir, const char *pfx);

#include <terminate.h>
void terminate();8

#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_error(char *errmsg);

#define _XOPEN_SOURCE
#include <search.h>
void *tfind(const void *key, void *const *rootp,
            int (*compar)(const void *, const void *));

#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_free(char *ptr, int struct_type);

#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_getinfo(int fd, struct t_info *info);

#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_getprotaddr(int fd, struct t_bind *boundaddr,
                  struct t_bind *peeraddr);

#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_getstate(int fd);

#include <time.h>
time_t time(time_t *timeptr);
```

---

## Function Format

---

```
#define _POSIX_SOURCE
#include <sys/times.h>
clock_t times(struct tms *buffer);
```

---

```
#include <mtf.h>
int tinit(const char *parallel_loadmod_name,
          int num_subtasks);7
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_listen(int fd, struct t_call *call);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_look(int fd);
```

---

```
#include <stdio.h>
FILE *tmpfile(void);
```

---

```
#include <stdio.h>
char *tmpnam(char *string);
```

---

```
#define _XOPEN_SOURCE
#include <ctype.h>
int toascii(int c);
#define _ALL_SOURCE
#include <ctype.h>
int toascii(int c);
```

---

```
#include <ctype.h>
int tolower(int c);
int toupper(int c);
```

---

```
#define _XOPEN_SOURCE
#include <ctype.h>
int _tolower(int c);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_open(char *name, int oflag,
           struct t_info *info);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_optmgt(int fd, struct t_optmgt *req,
            struct t_optmgt *ret);
```

---

```
#define _XOPEN_SOURCE
#include <ctype.h>
int _toupper(int c);
```

---

```
#include <wctype.h>
wint_t towlower(wint_t wc);
wint_t towupper(wint_t wc);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcv(int fd, char *buf, unsigned int nbytes,
         int *flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcvconnect(int fd, struct t_call *call);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcvdis(int fd, struct t_discon *discon);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcvrel(int fd);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcvudata(int fd, struct t_unitdata *unitdata,
               int *flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_rcvuderr(int fd, struct t_uderr *uderr);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int truncate(const char *path, off_t length);
```

---

```
#include <mtf.h>
int tsched(int task_id, const char *func_name,...);7
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
void *tsearch(const void *key, void **rootp,
              int (*compar)(const void *,
                             const void *));
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_snd(int fd, char *buf, unsigned int nbytes,
          int flags);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_snddis(int fd, struct t_call *call);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_sndrel(int fd);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_sndudata(int fd, struct t_unitdata *unitdata);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
char *t_strerror(int errnum);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_sync(int fd);
```

---

```
#include <mtf.h>
int tsyncro(int MTF_ANY|MTF_ALL|nn);7
```

---

```
#include <mtf.h>
int tterm(void);7
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
char *ttyname(int fildev);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
int ttyslot(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <xti.h>
int t_unbind(int fd);
```

---

```
#define _XOPEN_SOURCE
#include <search.h>
void *twalk(const void *root,
            void (*action)(const void *,
                           VISIT, int));
```

---

```
#define _POSIX_SOURCE
#include <time.h>
void tzset(void);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
useconds_t ualarm(useconds_t uscs,
                  useconds_t intrval);
```

---

```
#define _XOPEN_SOURCE
#include <ulimit.h>
long int ulimit(int cmd, ...);
```

---

```
#define _POSIX_SOURCE
#include <sys/stat.h>
mode_t umask(mode_t newmask);
```

---

```
#include <sys/stat.h>
int umount(const char *filesystem, mtm_t mtm);
```

---

```
#define _POSIX_SOURCE
#include <sys/utsname.h>
int uname(struct utsname *name);
```

---

```
#include <unistd.h>
void unexpected;8
```

---

```
#include <stdio.h>
int ungetc(int c, FILE *stream);
```

---

```
#include <wchar.h>
wint_t ungetwc(wint_t wc, FILE *stream);
```

---

```
#define _POSIX_SOURCE
#include <unistd.h>
int unlink(const char *pathname);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
int unlockpt(int fildev);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
int usleep( useconds_t useconds);
```

---

```
#define _POSIX_SOURCE
#include <utime.h>
int utime(const char *pathname,
          const struct utimbuf *newtimes);
```

## Function Format

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/time.h>
int utimes(const char *path,
           const struct timeval times);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <utmpx.h>
int __utmpxname(char *file);
```

---

```
#include <stdarg.h>
var_type va_arg(va_list arg_ptr, var_type);
void va_end(va_list arg_ptr);
void va_start(va_list arg_ptr, variable_name);

#define _XOPEN_SOURCE
#include <varargs.h>
var_type va_arg(va_list arg_ptr, var_type);
void va_end(va_list arg_ptr);
void va_start(va_list arg_ptr);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <stdlib.h>
void *valloc(size_t size);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/types.h>
#include <unistd.h>
pid_t vfork(void);
```

---

```
#include <stdarg.h>
#include <stdio.h>
int vfprintf(FILE *stream, const char *format,
             va_list arg_ptr);
```

---

```
#include <stdarg.h>
#include <stdio.h>
int vprintf(const char *format, va_list arg_ptr);
```

---

```
#include <stdarg.h>
#include <stdio.h>
int vsprintf(char *target-string, const char *format,
             va_list arg_ptr);
```

---

```
#include <stdarg.h>
#include <wchar.h>
int vswprintf(wchar_t *wcs, size_t n,
             const wchar_t * format,
             va_list arg);
```

---

```
#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <stdarg.h>
#include <wchar.h>
int vswprintf(wchar_t *wcs, size_t n,
             const wchar_t * format,
             va_list arg);
```

---

```
#define _POSIX_SOURCE
#include <sys/wait.h>
pid_t wait(int *status_ptr);
```

---

```
#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/wait.h>
int waitid(idtype_t idtype, id_t id,
           siginfo_t *info, int options);
```

---

## Function Format

---

```
#define _POSIX_SOURCE
#include <sys/wait.h>
pid_t waitpid(pid_t pid, int *status_ptr,
              int options);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/wait.h>
pid_t wait3 (int *stat_loc, int options,
             struct rusage *resource_usage);

#include <wchar.h>
int wctomb(char *s, wchar_t wchar, mbstate_t *pss);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int wctomb(char *s, wchar_t wchar, mbstate_t *pss);

#include <wchar.h>
/* or #include <wcstr.h> */
wchar_t *wcscat(wchar_t *string1,
                const wchar_t *string2);

#include <wchar.h>
/* or #include <wcstr.h> */
wchar_t *wcschr(const wchar_t *string1,
                wchar_t character);

#include <wchar.h>
/* or #include <wcstr.h> */
int wscmp(const wchar_t *string1,
          const wchar_t *string2);

#include <wchar.h>
int wscoll(const wchar_t *wcs1,
           const wchar_t *wcs2);

#include <wchar.h>
/*or #include <wcstr.h>*/
wchar_t *wcscpy(wchar_t *string1,
                const wchar_t *string2);

#include <wchar.h>
/*or #include <wcstr.h>*/
size_t wcsncpy(const wchar_t *string1,
               const wchar_t *string2);

#include <wchar.h>
size_t wcsftime(wchar_t *wcs, size_t maxsize,
                const wchar_t *format,
                const struct tm *time_ptr);

#define _XOPEN_SOURCE
#include <wchar.h>
size_t wcsftime(wchar_t *wcs, size_t maxsize,
                const char *format,
                const struct tm *time_ptr);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
size_t wcsftime(wchar_t *wcs, size_t maxsize,
                const wchar_t *format,
                const struct tm *time_ptr);

#include <stdlib.h>
int wcsid(const wchar_t c);
```

## Function Format

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
size_t wcslen(const wchar_t *string);
```

---

```
#include <wchar.h>
/*or #include <wcstr.h>*/
wchar_t *wcsncat(wchar_t *string1,
                 const wchar_t *string2,
                 size_t count);
```

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
int wcsncmp(const wchar_t *string1,
            const wchar_t *string2,
            size_t count);
```

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
wchar_t *wcsncpy(wchar_t *string1,
                 const wchar_t *string2,
                 size_t count);
```

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
wchar_t *wcspbrk(const wchar_t *string1,
                 const wchar_t *string2);
```

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
wchar_t *wcsrchr(const wchar_t *string,
                 wchar_t character);
```

---

```
#include <wchar.h>
size_t wcsrtombs(char *dst, const wchar_t **src,
                 size_t len, mbstate_t *ps);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
size_t wcsrtombs(char *dst, const wchar_t **src,
                 size_t len, mbstate_t *ps);
```

---

```
#include <wchar.h>
/* or #include <wcstr.h> */
size_t wcsspncpy(const wchar_t *string1,
                 const wchar_t *string2);
```

---

```
#include <wchar.h>
wchar_t *wcsstr(const wchar_t *wcs1,
                const wchar_t *wcs2);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wcsstr(const wchar_t *wcs1,
                const wchar_t *wcs2);
```

---

```
#include <wchar.h>
double wcstod(const wchar_t *nptr, wchar_t **endptr);
```



## Function Format

---

```
#include <wchar.h>
wchar_t *wcstok(wchar_t *wcs1, const wchar_t *wcs2,
               wchar_t **ptr);

#define _XOPEN_SOURCE
#include <wchar.h>
wchar_t *wcstok(wchar_t *wcs1, const wchar_t *wcs2);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wcstok(wchar_t *wcs1, const wchar_t *wcs2,
               wchar_t **ptr);

#include <wchar.h>
long int wcstol(const wchar_t *nptr,
               wchar_t **endptr,
               int base);

#include <stdlib.h>
size_t wcstombs(char *dest, const wchar_t *string,
               size_t count);

#include <wchar.h>
unsigned long int wcstoul(const wchar_t *nptr,
                       wchar_t **endptr,
                       int base);

#include <wcstr.h>
wchar_t *wcswcs(const wchar_t *string1,
               const wchar_t *string2);

#include <wchar.h>
int wcswidth(const wchar_t *wcs, size_t n);

#include <wchar.h>
size_t wcsxfrm(wchar_t *wcs1, const wchar_t *wcs2,
               size_t n);

#include <wchar.h>
int wctob(wint_t c);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int wctob(wint_t c);

#include <stdlib.h>
int wctomb(char *string, wchar_t character);

#include <wchar.h>
wctype_t wctype(const char *property);

#include <wchar.h>
int wcwidth(const wint_t wc);

#define _XOPEN_SOURCE
#include <wchar.h>
int wcwidth(const wchar_t wc);

#include <sys/mntent.h>
int w_getmntent(char *buffer, int size);

#include <sys/ps.h>
int w_getpsent(int token, W_PSPROC *buffptr,
               size_t length);
```

## Function Format

---

```
#include <termios.h>
int w_ioctl(int fildes, int cmd,
            int arglen, void *arg);

#include <wchar.h>
wchar_t *wmemchr(const wchar_t *s, wchar_t c,
                 size_t n);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wmemchr(const wchar_t *s, wchar_t c,
                 size_t n);

#include <wchar.h>
int wmemcmp(const wchar_t *s1, const wchar_t *s2,
            size_t n);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
int wmemcmp(const wchar_t *s1,
            const wchar_t *s2, size_t n);

#include <wchar.h>
wchar_t *wmemcpy(wchar_t *s1,
                 const wchar_t *s2, size_t n);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wmemcpy(wchar_t *s1,
                 const wchar_t *s2, size_t n);

#include <wchar.h>
wchar_t *wmemmove(wchar_t *s1,
                  const wchar_t *s2, size_t n);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wmemmove(wchar_t *s1,
                  const wchar_t *s2, size_t n);

#include <wchar.h>
wchar_t *wmemset(wchar_t *s, wchar_t c, size_t n);

#define _XOPEN_SOURCE
#define _MSE_PROTOS
#include <wchar.h>
wchar_t *wmemset(wchar_t *s, wchar_t c, size_t n);

#define _XOPEN_SOURCE
#include <wordexp.h>
int wordexp(const char *words, wordexp_t *pwordexp,
            int flags);

#define _XOPEN_SOURCE
#include <wordexp.h>
void wordfree(wordexp_t *pwordexp);
```

---

## Function Format

---

```
#define _POSIX_SOURCE
#include <unistd.h>
ssize_t write(int fildev, const void *buf,
              size_t N);

#define _XOPEN_SOURCE_EXTENDED 1
#include <unistd.h>
ssize_t write(int socket, const void *buffer,
              ssize_t length);

#define _OE_SOCKETS
#include <unistd.h>
ssize_t write(int socket, const void *buffer,
              ssize_t length);

#define _XOPEN_SOURCE_EXTENDED 1
#include <sys/uio.h>
ssize_t writev(int fildev, const struct iovec *iov,
               int iovcnt);

#define _OE_SOCKETS
#include <sys/uio.h>
int writev(int *s, struct iovec *iov,
           int iovcnt);

#include <unistd.h>
int __wsinit( void (*func_ptr)() );

#include <sys/statfs.h>
int w_statfs(const char *filesystem,
             struct w_statfs *statbuf,
             size_t length);

#define _OPEN_SOURCE 2
#include <sys/statvfs.h>
int w_statvfs(const char *filesystem,
              struct statvfs *buffer,
              size_t buflen);

#include <spc.h>
void * __xhotc(void * handle, int stack,
              int location);7

#include <spc.h>
void * __xhotl(void * handle, int stack,
              int location);7

#include <spc.h>
void __xhott(void * handle);7

#include <spc.h>
void * __xhotu(void * handle,
              void * funcname, ...);7

#include <spc.h>
int __xregs(int register);7

#include <spc.h>
void __xsacc(int message);7

#include <spc.h>
void * __xsrvc(int message);7

#include <spc.h>
void * __xusr(void);7

#include <spc.h>
void * __xusr2(void);7
```

### Function Format

---

```
#include <math.h>
double y0(double x);
double y1(double x);
double yn(int n, double x);
```

---

```
#include <spc.h>
void * __24malc(size_t size);7
```

---

```
#include <spc.h>
void * __4kmalc(size_t size);8
```

---

7. C only function.

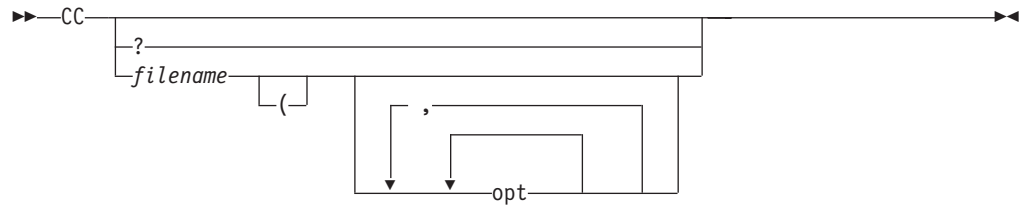
8. C++ only function.

---

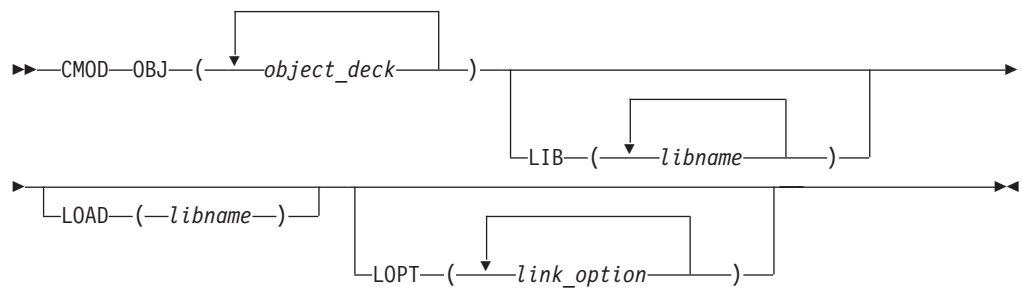
## Utilities

The following sections list the utilities available under OS/390 C/C++:

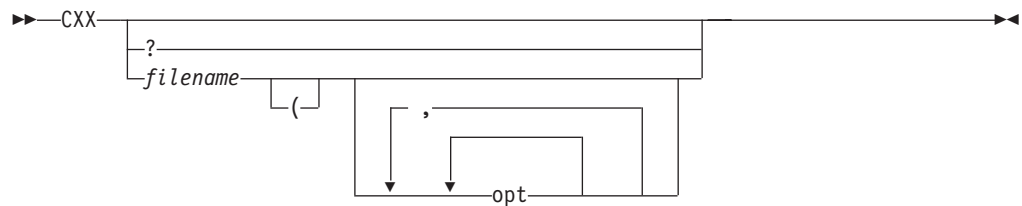
### CC Utility



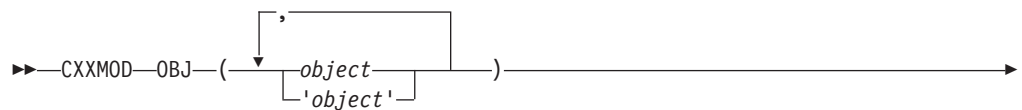
### CMOD Utility

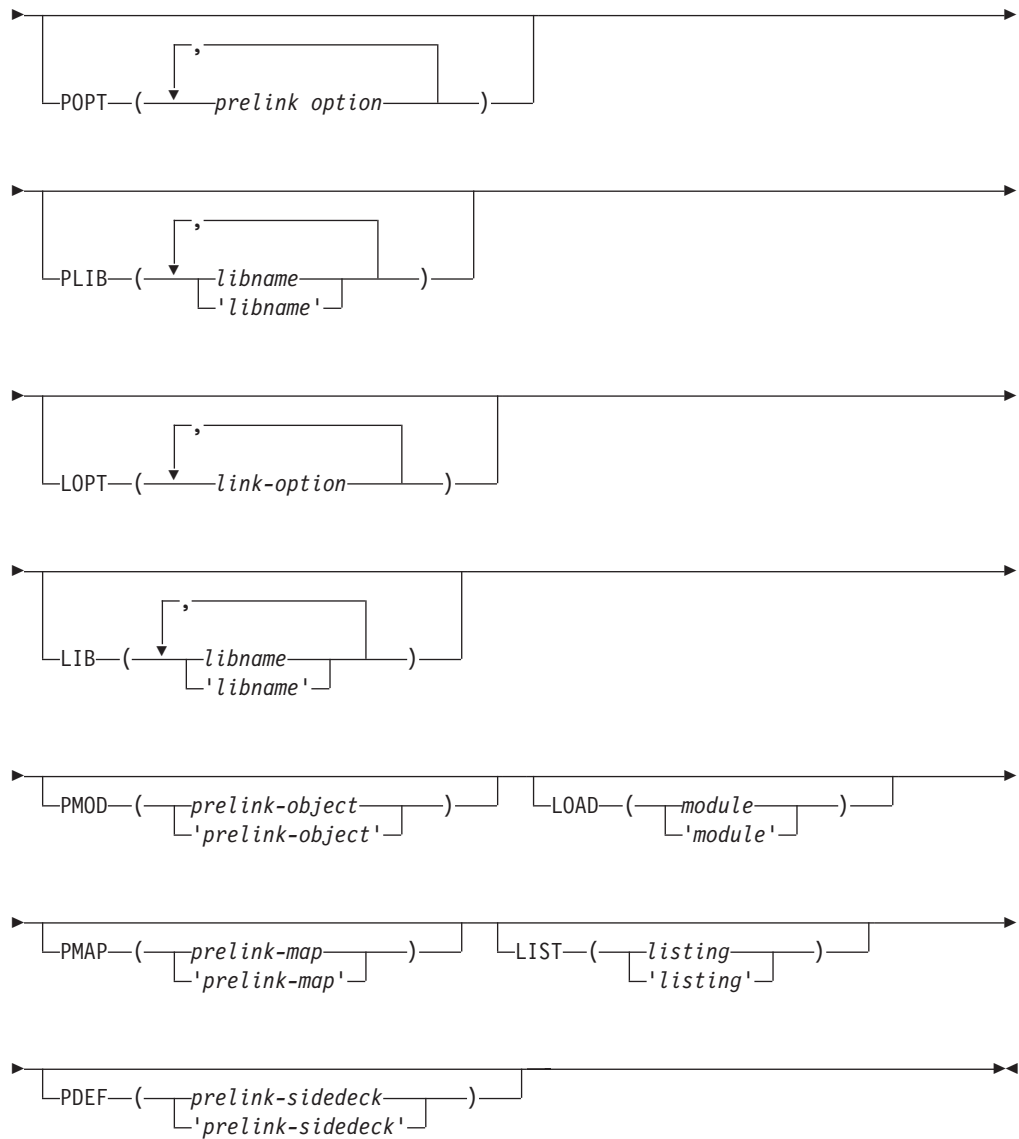


### CXX Utility

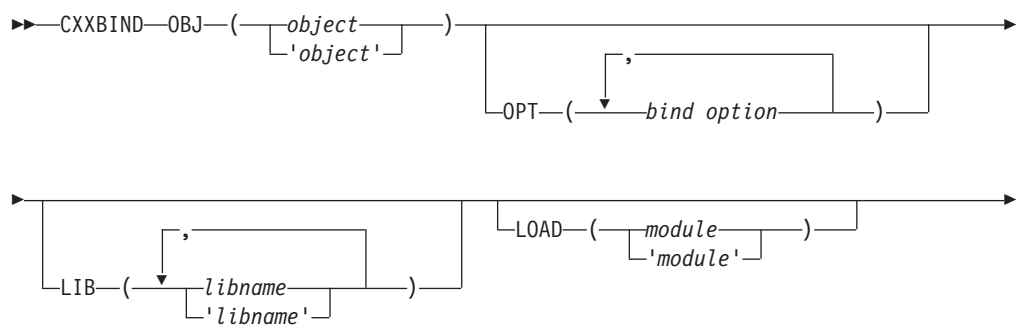


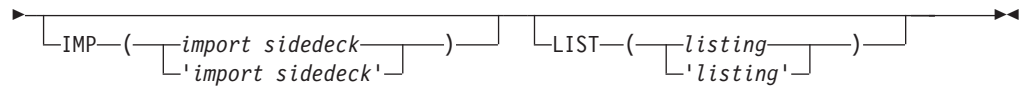
### CXXMOD Utility





## CXXBIND Utility

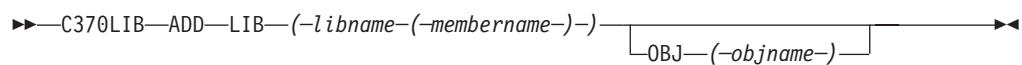




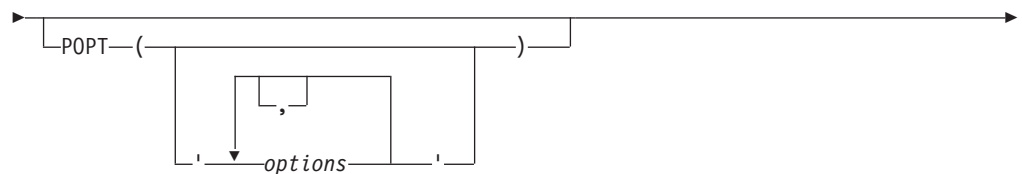
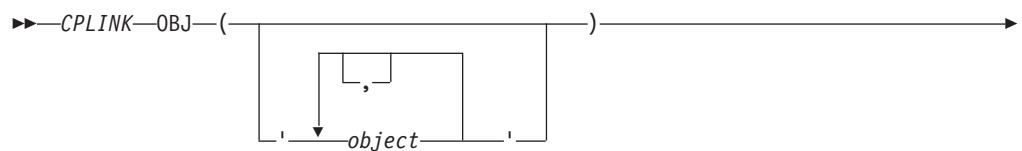
## C/C++ Compilation Utilities under the OS/390 UNIX Shell

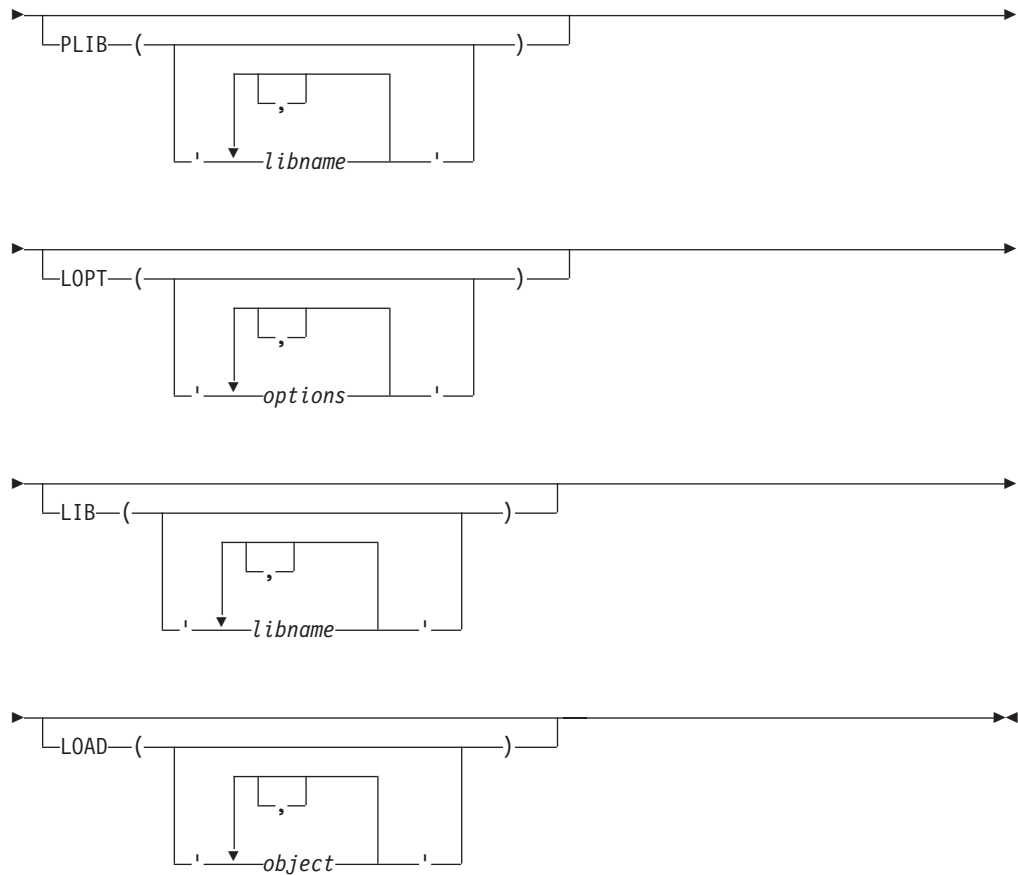
```
c89|cc|c++|cxx [-options ...]
[file.c ...]for C
[file.C ...]for C++
[file.a ...]
[file.o ...]
[-l libname]
```

## Object Library Utility

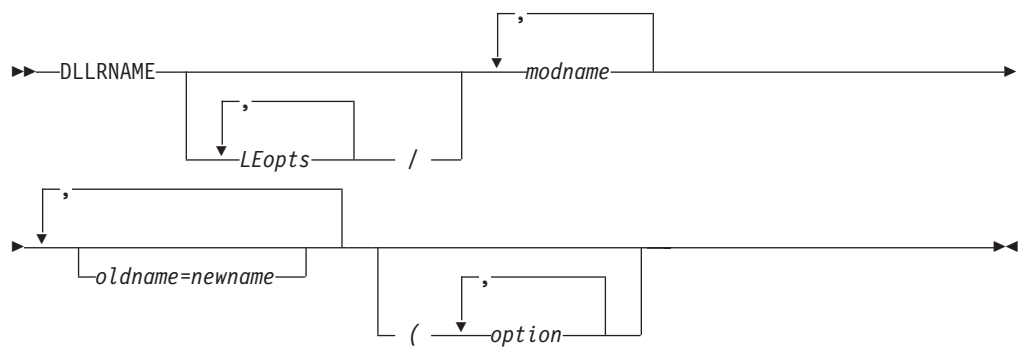


## CPLINK Utility

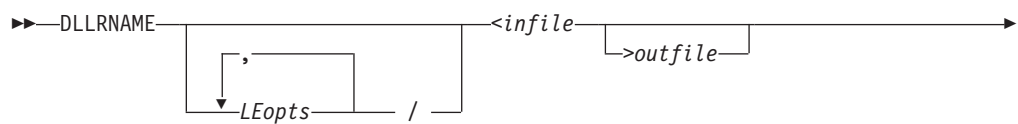




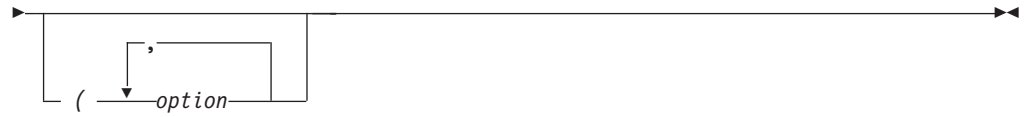
## DLLRNAME Utility



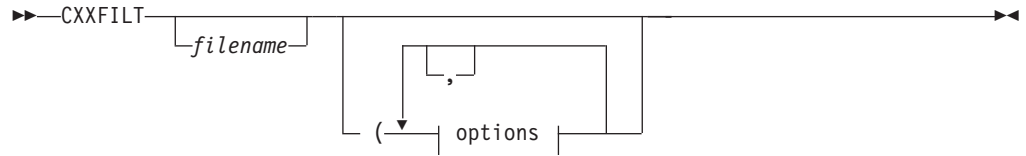
OR



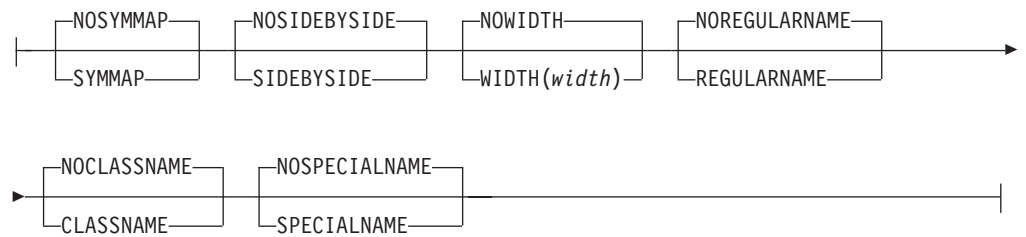




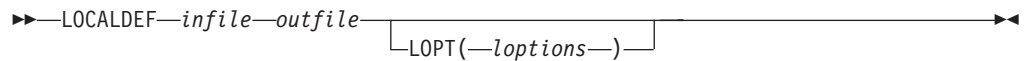
## CXXFILT Utility



### options:



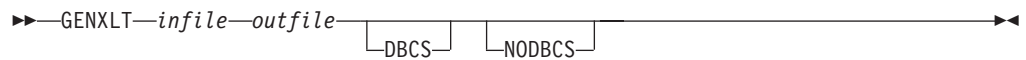
## Locale Utility (localedef)



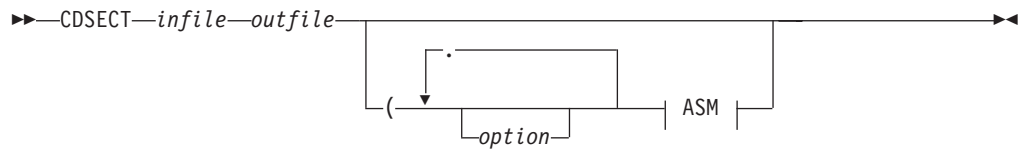
## iconv Utility



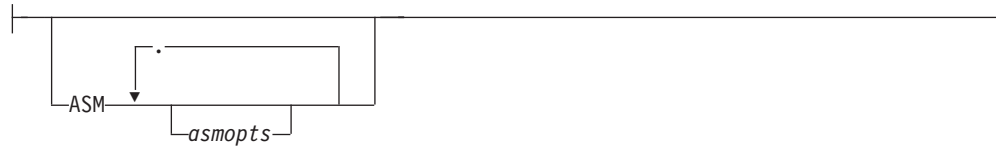
## genxlt Utility



## DSECT Conversion Utility



### ASM:



## DSECT Options

DSECT Utility Option	Abbreviated Name	IBM Supplied Default
SECT[( <i>name</i> ,...)]	None	SECT(ALL)
BITF0XL   NOBITF0XL	BITF   NOBITF	NOBITF0XL
COMMENT[( <i>delim</i> ,...)]   NOCOMMENT	COM   NOCOM	COMMENT
DEFSUB   NODEFSUB	DEF   NODEF	DEFSUB
EQUATE[( <i>suboptions</i> ,...)]   NOEQUATE	EQU   NOEQU	NOEQUATE
HDRSKIP[( <i>length</i> )]   NOHDRSKIP	HDR   NOHDR	NOHDRSKIP
LOCALE[( <i>name</i> )]   NOLOCALE	LOC   NOLOC	NOLOCALE
INDENT[( <i>count</i> )]   NOINDENT	IN   NOIN	INDENT(2)
LOWERCASE   NOLOWERCASE	LC   NOLC	LOWERCASE
OPTFILE[( <i>filename</i> )]   NOOPTFILE	OPTF   NOOPTF	NOOPTFILE
PPCOND[( <i>switch</i> )]   NOPPCOND	PP   NOPP	NOPPCOND
SEQUENCE   NOSEQUENCE	SEQ   NOSEQ	NOSEQUENCE
UNNAMED   NOUNNAMED	UNN   NOUNN	NOUNNAMED
OUTPUT[( <i>filename</i> )]	OUT	OUTPUT (DD:EDCDSECT)
RECFM[( <i>recfm</i> )]	None	C or C++ library defaults
LRECL[( <i>lrecl</i> )]	None	C or C++ library defaults
BLKSIZE[( <i>blksize</i> )]	None	C or C++ library defaults





Printed in the United States of America

SX09-1313-03

