



MSDN Home

| Developer Centers | Library | Downloads | How to Buy | Subscribers | Worldwide

Search for

Welcome to the MSDN Library

[Advanced Search](#)
▼ ▲ sync toc ✖

+ Welcome to the MSDN Library
- Development Tools and Languages
+ Visual Studio 2005
- Visual Studio .NET
+ Introducing Visual Studio .NET
+ Getting Assistance
+ Samples and Walkthroughs
+ Developing with Visual Studio .NET
+ Visual Basic and Visual C#
+ Visual J#
- Visual C++
+ Getting Started
+ Porting and Upgrading
+ Creating and Managing Projects
+ Building a C/C++ Program
+ Adding Functionality
+ Managed Extensions for C++ Programming
- Reference
- Visual C++ Libraries
+ ATL Reference
+ ATL Server Reference
+ MFC Reference
+ Shared Classes
+ OLE DB Templates
- Run-Time Library Reference
+ C Run-Time Libraries
+ Run-Time Routines by Category
. Run-Time Routines and .NET Framework Equivalents
+ Global Variables and Standard Types

[MSDN Home](#) > [MSDN Library](#) > [Development Tools and Languages](#) > [Visual Studio .NET](#) > [Visual C++](#) > [Reference](#) > [Visual C++ Libraries](#) > [Run-Time Library Reference](#) > [Alphabetical Function Reference](#) > [M Through R](#)

Run-Time Library Reference

qsort

Performs a quick sort.

```
void qsort(
    void *base,
    size_t num,
    size_t width,
    int (_cdecl *compare )(const void *, const void *));
;
```

Parameters

base
Start of target array.

num
Array size in elements.

width
Element size in bytes.

compare
Comparison function. The first parameter is a pointer to the key for the search and the second parameter is a pointer to the array element to be compared with the key.

Remarks

The **qsort** function implements a quick-sort algorithm to sort an array of *num* elements, each of *width* bytes. The argument *base* is a pointer to the base of the array to be sorted. **qsort** overwrites this array with the sorted elements. The argument *compare* is a pointer to a user-supplied routine that compares two array elements and returns a value specifying their relationship. **qsort** calls the *compare* routine one or more times during the sort, passing pointers to two array elements on each call:

```
compare( void * ) elem1, (void * ) elem2 );
```

The routine must compare the elements, then return one of the following values:

Return value	Description
--------------	-------------

+ Global Constants
- Alphabetical Function Reference
+ A Through F
+ G Through L
- M Through R
. _makepath, . _wmakepath . malloc . _malloc_dbg . _matherr . __max . _mbbtombc . _mbbtype . _mbccpy . _mbcjstojms, . _mbcjmstojis . _mbclen, mblen . _mbctohira, . _mbctokata . _mbctolower, . _mbctoupper . _mbctombb . _mbsbtype . _mbsdec, _strdec, . _wcsdec . _mbsinc, _strinc, . _wcsinc . mbsinit . _mbsnbcat . _mbsnbcmp . _mbsnbcnt, . _mbsnccnt, _strncnt, . _wcscnt . _mbsnbcoll, . _mbsnbicoll . _mbsnbcpy . _mbsnbicmp . _mbsnbset . _mbsnextc, . _strnextc, _wcsnextc . _mbsninc, _strninc, . _wcsninc . _mbsspnp, _strspnp, . _wcsspnp . mbstowcs . mbtowc . _memccpy . memchr, wmemchr . memcmp, wmemcmp . memcpy, wmemcpy . _memicmp . memmove, . wmemmove

< 0	elem1 less than elem2
0	elem1 equivalent to elem2
> 0	elem1 greater than elem2

The array is sorted in increasing order, as defined by the comparison function. To sort an array in decreasing order, reverse the sense of "greater than" and "less than" in the comparison function.

Requirements

Routine	Required header	Compatibility
qsort	<stdlib.h> and <search.h>	ANSI, Win 98, Win Me, Win NT, Win 2000, Win XP

For additional compatibility information, see [Compatibility](#) in the Introduction.

Libraries

All versions of the [C run-time libraries](#).

Example

```
// crt_qsort.c
// arguments: every good boy deserves favor

/* This program reads the command-line
 * parameters and uses qsort to sort them. It
 * then displays the sorted arguments.
 */

#include <stdlib.h>
#include <string.h>
#include <stdio.h>

int compare( const void *arg1, const void *arg2 );

int main( int argc, char **argv )
{
    int i;
    /* Eliminate argv[0] from sort: */
    argv++;
    argc--;
    /* Sort remaining args using Quicksort algorithm: */
    qsort( (void *)argv, (size_t)argc, sizeof( char * ), compare );

    /* Output sorted list: */
    for( i = 0; i < argc; ++i )
        printf( " %s", argv[i] );
    printf( "\n" );
}

int compare( const void *arg1, const void *arg2 )
{
    /* Compare all of both strings: */
    return _stricmp( *(char**) arg1, *(char**) arg2 );
}
```

. memset, wmemset
. __min
. _mkdir, _wmkdir
. _mktemp, _wmktemp
. mktime, _mktime64
. modf, modff
. _msize
. _msize_dbg
. _nextafter
. operator new
. operator new[]
. offsetof
. _onexit
. _open, _wopen
. _open_osfhandle
. _outp, _outpw,
_outpd
. _pclose
. perror, _wperror
. _pipe
. _popen, _wpopen
. pow, powf
. printf, wprintf
. Format Specification
Fields: printf and
wprintf Functions
. printf Type Field
Characters
. Flag Directives
. printf Width
Specification
. Precision
Specification
. Size and Distance
Specification
. putc, putwc, putchar,
putwchar
. _putch, _putwch
. _putenv, _wpputenv
. puts, _putws
. _putw
. qsort
. _query_new_handler
. _query_new_mode
. raise
. rand
. _read
. realloc
. _realloc_dbg
. remove, _wremove
. rename, _wrename
. _resetstkoflw
. rewind
. _rmdir, _wrmdir
. _rmtmp
. _rotl, _rotl64,

Output

boy deserves every favor good

See Also

[Searching and Sorting Routines](#) | [bsearch](#) | [lsearch](#) | [Run-Time Routines and .NET Framework Equivalents](#)

[Manage Your Profile](#) |
[Legal](#) | [Contact Us](#) |
[MSDN Flash Newsletter](#)

© 2005 Microsoft
Corporation. All rights
reserved. [Terms of Use](#) |
[Trademarks](#) |
[Privacy Statement](#)



- _rotr_rotr64
 - . _RPT, _RPTF Macros
 - . _RTC_GetErrDesc
 - . _RTC_NumErrors
 - . _RTC_SetErrorFunc
 - . _RTC_SetErrorType
- + S Through Z
 - + Generic-Text Mappings
 - + Language and Country/Region Strings
- + Standard C++ Library Reference
 - + C++ Attributes Reference
 - + C/C++ Languages
 - + Visual C++ Extensibility Object Model
 - + Microsoft Macro Assembler Reference
 - + Samples
 - + JScript
 - + Articles and Columns
 - + Book Excerpts
 - + Periodicals
- .NET Framework SDK (see .NET Development)
 - + Visual Studio 6.0
 - + Visual FoxPro
- + Mobile and Embedded Development
 - + .NET Development
 - + Office Solutions Development
 - + Servers and Enterprise Development
 - + Web Development
 - + Win32 and COM Development
- .MSDN Library Archive