

UNIX-Type APIs (V5R2)

Software Clock APIs

Table of Contents

[Software Clock APIs](#)

- [adjtime\(\)](#) (Adjust software clock)
- [gettimeofday\(\)](#) (Get current software clock time)
- [settimeofday\(\)](#) (Set current software clock time)

[Header Files for UNIX-Type Functions](#)

[Errno Values for UNIX-Type Functions](#)

Software Clock APIs

The Software Clock APIs are:

- [adjtime\(\)](#) (Adjust software clock) makes small adjustments to the software clock, either advancing or retarding it by the time specified in the delta parameter.
- [gettimeofday\(\)](#) (Get current software clock time) retrieves the software-clock-based system time and places it in the timeval structure pointed to by tp.
- [settimeofday\(\)](#) (Set current software clock time) sets the software clock to the time contained in the timeval structure pointed to by tp.

Note: These functions use header (include) files from the library QSYSINC, which is optionally installable. Make sure QSYSINC is installed on your system before using any of the functions. See for the file and member name of each header file.

The Software Clock is a system facility that determines the current Universal Coordinated Time (UTC) in seconds and microseconds since 1 January 1970, 00:00:00 (epoch-1970), and allows for time zone and daylight savings time information to be stored and retrieved. The current time is determined by keeping an internal 'time-delta', which is an offset from the machine clock (accessible by the `machtime()` API). When **gettimeofday()** is called to retrieve the software clock time, the time returned is calculated by taking the current machine time, subtracting the QUTCOFFSET system value, then adding the internal time-delta.

Calling the **settimeofday()** API sets or changes the time-delta (and optionally the time zone and the daylight savings time information), but does not affect the machine clock or the QUTCOFFSET system value. The **adjtime()** API slowly changes the time-delta, but does not affect the machine clock either. Adjustments are made at a rate of approximately 1 second of adjustment for every 100 seconds of elapsed time.

Most system components do not base their timestamps on the software clock, but use the machine clock instead.

adjtime()-Adjust Software Clock

Syntax

```
#include <sys/time.h>

int adjtime (struct timeval *delta,
             struct timeval *olddelta);
```

Service Program Name: QP0ZSETC

Default Public Authority: *USE

Threadsafe: Yes

The **adjtime()** function makes small adjustments to the software clock, either advancing or retarding it by the time specified in the delta parameter. If delta is negative, the clock is slowed down by incrementing it more slowly than normal until the correction is complete. If delta is positive, the clock is sped up by incrementing it more quickly than normal until the correction is complete. If olddelta is not NULL, the amount of time still to be corrected from a previous **adjtime()** call is returned in the structure it points to.

Parameters

delta

(Input) A pointer to a timeval structure that contains the amount of time to be altered.

olddelta

(Output) A pointer to a timeval structure that contains the amount of time still to be corrected from a previous call to **adjtime()**

Authorities and Locks

QSYS/QP0ZXCPA Service Program Authority

*USE

Return Value

- 0 **adjtime()** was successful. The requested adjustment was initiated and the value returned in the structure pointed to by the olddelta parameter is the amount of time still to be corrected from a previous **adjtime()**.
- 1 **adjtime()** was not successful. The errno variable is set to indicate the error.

Error Conditions

If `adjtime()` is not successful, `errno` usually indicates one of the following errors. Under some conditions, `errno` could indicate an error other than those listed here.

- [EINVAL]* An invalid parameter was found.
- A parameter passed to this function is not valid.
- [EFAULT]* The address used for an argument is not correct.
- In attempting to use an argument in a call, the system detected an address that is not valid.
- While attempting to access a parameter passed to this function, the system detected an address that is not valid.
- [EPERM]* Operation not permitted.
- You must have appropriate privileges or be the owner of the object or other resource to do the requested operation.
- [EUNKNOWN]* Unknown system state.
- The operation failed because of an unknown system state. See any messages in the job log and correct any errors that are indicated, then retry the operation.

Error Messages

None.

Related Information

- The `<sys/time.h>` file (see [Header Files for UNIX-Type Functions](#))
- [gettimeofday\(\)-Get Current Software Clock Time](#)
- [settimeofday\(\)-Set Current Software Clock Time](#)

Example

See [Code disclaimer information](#) for information pertaining to code examples.

The following example initiates a software clock adjustment:

```
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>

int main(int argc, char *argv[])
{
    struct timeval adj, old;
    int rc;

    adj.tv_sec=1;
    adj.tv_usec=500000;

    rc=adjtime(&adj, &old);
    if(rc==0) {
        printf("adjtime() successful. "
               "Olddelta = %u.%06u\n",
               old.tv_sec, old.tv_usec);
    }
    else {
        printf("adjtime() failed, errno = %d\n",errno);
        return -1;
    }

    return 0;
}
```

Example Output:

```
adjtime() successful. Olddelta = 0.000000
```

API introduced: V4R2

[Top](#) | [UNIX-Type APIs](#) | [APIs by category](#)

gettimeofday()-Get Current Software Clock Time

Syntax

```
#include <sys/time.h>

int gettimeofday (struct timeval *tp,
                  struct timezone *tzp);
```

Service Program Name: QSOZCPA

Default Public Authority: *USE

Threadsafe: Yes

The **gettimeofday()** function retrieves the software-clock-based system time and places it in the timeval structure pointed to by tp. If tzp is not NULL, the time zone information is returned in the timezone structure pointed to by tzp.

Parameters

tp

(Output) A pointer to a timeval structure that contains the time in seconds and microseconds since 1 January 1970, 00:00:00 UTC (epoch-1970).

tzp

(Output) A pointer to a timezone structure that contains the local time zone (measured in minutes of time westward from Greenwich) and a flag that, if nonzero, indicates daylight saving time applies locally during the appropriate part of the year.

Authorities and Locks

None

Return Value

0 **gettimeofday()** was successful.

-1 **gettimeofday()** was not successful. The errno variable is set to indicate the error.

Error Conditions

If `gettimeofday()` is not successful, `errno` usually indicates one of the following errors. Under some conditions, `errno` could indicate an error other than those listed here.

- [EINVAL]* An invalid parameter was found.
A parameter passed to this function is not valid.
- [EFAULT]* The address used for an argument is not correct.
In attempting to use an argument in a call, the system detected an address that is not valid.
While attempting to access a parameter passed to this function, the system detected an address that is not valid.
- [EUNKNOWN]* Unknown system state.
The operation failed because of an unknown system state. See any messages in the job log and correct any errors that are indicated, then retry the operation.

Error Messages

None.

Related Information

- The `<sys/time.h>` file (see)
- [adjtime\(\)-Adjust Software Clock](#)
- [settimeofday\(\)-Set Current Software Clock Time](#)

Example

See [Code disclaimer information](#) for information pertaining to code examples.

The following example gets the current time:

```
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>

int main(int argc, char *argv[])
```

```

{
    struct timeval time;
    struct timezone timez;
    int rc;

    rc=gettimeofday(&time, &timez);
    if(rc==0) {
        printf("gettimeofday() successful.\n");
        printf("time = %u.%06u, "
            "minuteswest = %d, "
            "dsttime = %d\n",
            time.tv_sec, time.tv_usec,
            timez.tz_minuteswest,
            timez.tz_dsttime );
    }
    else {
        printf("gettimeofday() failed, errno = %d\n",
            errno);
        return -1;
    }

    return 0;
}

```

Example Output:

```

gettimeofday() successful.
time = 866208142.290944, minuteswest = 360, dsttime = 1

```

API introduced: V4R2

[Top](#) | [UNIX-Type APIs](#) | [APIs by category](#)

settimeofday()-Set Current Software Clock Time

Syntax

```
#include <sys/time.h>

int settimeofday (struct timeval *tp,
                  struct timezone *tzp);
```

Service Program Name: QP0ZSETC

Default Public Authority: *USE

Threadsafe: Yes

The **settimeofday()** function sets the software clock to the time contained in the timeval structure pointed to by tp. If tzp is not NULL, the time zone information is also set.

Parameters

tp

(Input) A pointer to a timeval structure that contains the time in seconds and microseconds since 1 January 1970, 00:00:00 UTC (epoch-1970).

tzp

(Input) A pointer to a timezone structure that contains the local time zone (measured in minutes of time westward from Greenwich) and a flag that, if nonzero, indicates daylight saving time applies locally during the appropriate part of the year.

Authorities and Locks

QSYS/QP0ZXCPA Service Program Authority

*USE

Return Value

0

settimeofday() was successful.

-1

settimeofday() was not successful. The errno variable is set to indicate the error.

Error Conditions

If `settimeofday()` is not successful, `errno` usually indicates one of the following errors. Under some conditions, `errno` could indicate an error other than those listed here.

- [EINVAL]* An invalid parameter was found.
- A parameter passed to this function is not valid.
- [EFAULT]* The address used for an argument is not correct.
- In attempting to use an argument in a call, the system detected an address that is not valid.
- While attempting to access a parameter passed to this function, the system detected an address that is not valid.
- [EPERM]* Operation not permitted.
- You must have appropriate privileges or be the owner of the object or other resource to do the requested operation.
- [EUNKNOWN]* Unknown system state.
- The operation failed because of an unknown system state. See any messages in the job log and correct any errors that are indicated, then retry the operation.

Error Messages

None.

Related Information

- The `<sys/time.h>` file (see)
- [adjtime\(\)-Adjust Software Clock](#)
- [gettimeofday\(\)-Get Current Software Clock Time](#)

Example

See [Code disclaimer information](#) for information pertaining to code examples.

The following example sets the software clock:

```
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>

int main(int argc, char *argv[])
{
    struct timeval time;
    struct timezone timez;
    int rc;

    time.tv_sec=866208142;
    time.tv_usec=290944;

    timez.tz_minuteswest=360;
    timez.tz_dsttime=1;

    rc=settimeofday(15:17:05, &timez);
    if(rc==0) {
        printf("settimeofday() successful.\n");
    }
    else {
        printf("settimeofday() failed, %s\n",
            strerror(errno));
        return -1;
    }

    return 0;
}
```

Example Output:

```
settimeofday() successful.
```

API introduced: V4R2

[Top](#) | [UNIX-Type APIs](#) | [APIs by category](#)

Header Files for UNIX-Type Functions

Programs using the UNIX-type functions must include one or more header files that contain information needed by the functions, such as:

- Macro definitions
- Data type definitions
- Structure definitions
- Function prototypes

The header files are provided in the QSYSINC library, which is optionally installable. Make sure QSYSINC is on your system before compiling programs that use these header files. For information on installing the QSYSINC library, see [Data structures and the QSYSINC Library](#).

The table below shows the file and member name in the QSYSINC library for each header file used by the UNIX-type APIs in this publication.

Name of Header File	Name of File in QSYSINC	Name of Member
arpa/inet.h	ARPA	INET
arpa/nameser.h	ARPA	NAMESER
bse.h	H	BSE
bsedos.h	H	BSEDOS
bseerr.h	H	BSEERR
dirent.h	H	DIRENT
errno.h	H	ERRNO
fcntl.h	H	FCNTL
grp.h	H	GRP
»inttypes.h	H	INTTYPES«
limits.h	H	LIMITS
»mman.h	H	MMAN«
netdbh.h	H	NETDB
»netinet/icmp6.h	NETINET	ICMP6«
net/if.h	NET	IF
netinet/in.h	NETINET	IN
netinet/ip_icmp.h	NETINET	IP_ICMP
netinet/ip.h	NETINET	IP
»netinet/ip6.h	NETINET	IP6«
netinet/tcp.h	NETINET	TCP
netinet/udp.h	NETINET	UDP
netns/idp.h	NETNS	IDP
netns/ipx.h	NETNS	IPX
netns/ns.h	NETNS	NS
netns/sp.h	NETNS	SP
net/route.h	NET	ROUTE
nettel/tel.h	NETTEL	TEL

os2.h	H	OS2
os2def.h	H	OS2DEF
pwd.h	H	PWD
Qlg.h	H	QLG
qp0lflop.h	H	QP0LFLOP
»qp0ljrnل.h	H	QP0LJRNL«
»qp0lrور.h	H	QP0LROR«
Qp0lstdi.h	H	QP0LSTDI
qp0wpid.h	H	QP0WPID
qp0zdipc.h	H	QP0ZDIPC
qp0zipc.h	H	QP0ZIPC
qp0zolip.h	H	QP0ZOLIP
qp0zolsm.h	H	QP0ZOLSM
qp0zripc.h	H	QP0ZRIPC
qp0ztrc.h	H	QP0ZTRC
qp0ztrml.h	H	QP0ZTRML
qp0z1170.h	H	QP0Z1170
»qsoasync.h	H	QSOASYNC«
qtnxaapi.h	H	QTNXAAPI
qtnxadtp.h	H	QTNXADTP
qtomeapi.h	H	QTOMEAPI
qtossapi.h	H	QTOSSAPI
resolv.h	H	RESOLVE
semaphore.h	H	SEMAPHORE
signal.h	H	SIGNAL
spawn.h	H	SPAWN
ssl.h	H	SSL
sys/errno.h	H	ERRNO
sys/ioctl.h	SYS	IOCTL
sys/ipc.h	SYS	IPC
sys/layout.h	H	LAYOUT
sys/limits.h	H	LIMITS
sys/msg.h	SYS	MSG
sys/param.h	SYS	PARAM
»sys/resource.h	SYS	RESOURCE«
sys/sem.h	SYS	SEM
sys/setjmp.h	SYS	SETJMP
sys/shm.h	SYS	SHM
sys/signal.h	SYS	SIGNAL
sys/socket.h	SYS	SOCKET
sys/stat.h	SYS	STAT
sys/statvfs.h	SYS	STATVFS

sys/time.h	SYS	TIME
sys/types.h	SYS	TYPES
sys/uio.h	SYS	UIO
sys/un.h	SYS	UN
sys/wait.h	SYS	WAIT
» ulimit.h	H	ULIMIT «
unistd.h	H	UNISTD
utime.h	H	UTIME

You can display a header file in QSYSINC by using one of the following methods:

- Using your editor. For example, to display the **unistd.h** header file using the Source Entry Utility editor, enter the following command:

```
STRSEU SRCFILE(QSYSINC/H) SRCMBR(UNISTD) OPTION(5)
```

- Using the Display Physical File Member command. For example, to display the **sys/stat.h** header file, enter the following command:

```
DSPPFM FILE(QSYSINC/SYS) MBR(STAT)
```

You can print a header file in QSYSINC by using one of the following methods:

- Using your editor. For example, to print the **unistd.h** header file using the Source Entry Utility editor, enter the following command:

```
STRSEU SRCFILE(QSYSINC/H) SRCMBR(UNISTD) OPTION(6)
```

- Using the Copy File command. For example, to print the **sys/stat.h** header file, enter the following command:

```
CPYF FROMFILE(QSYSINC/SYS) TOFILE(*PRINT) FROMMBR(STAT)
```

Symbolic links to these header files are also provided in directory /QIBM/include.

Errno Values for UNIX-Type Functions

Programs using the UNIX-type functions may receive error information as *errno* values. The possible values returned are listed here in ascending *errno* value sequence.

Name	Value	Text
EDOM	3001	A domain error occurred in a math function.
ERANGE	3002	A range error occurred.
ETRUNC	3003	Data was truncated on an input, output, or update operation.
ENOTOPEN	3004	File is not open.
ENOTREAD	3005	File is not opened for read operations.
EIO	3006	Input/output error.
ENODEV	3007	No such device.
ERECIO	3008	Cannot get single character for files opened for record I/O.
ENOTWRITE	3009	File is not opened for write operations.
ESTDIN	3010	The stdin stream cannot be opened.
ESTDOUT	3011	The stdout stream cannot be opened.
ESTDERR	3012	The stderr stream cannot be opened.
EBADSEEK	3013	The positioning parameter in fseek is not correct.
EBADNAME	3014	The object name specified is not correct.
EBADMODE	3015	The type variable specified on the open function is not correct.
EBADPOS	3017	The position specifier is not correct.
ENOPOS	3018	There is no record at the specified position.
ENUMMBRS	3019	Attempted to use ftell on multiple members.
ENUMRECS	3020	The current record position is too long for ftell.
EINVAL	3021	The value specified for the argument is not correct.
EBADFUNC	3022	Function parameter in the signal function is not set.
ENOENT	3025	No such path or directory.
ENOREC	3026	Record is not found.
EPERM	3027	The operation is not permitted.
EBADDATA	3028	Message data is not valid.
EBUSY	3029	Resource busy.
EBADOPT	3040	Option specified is not valid.
ENOTUPD	3041	File is not opened for update operations.
ENOTDLT	3042	File is not opened for delete operations.

EPAD	3043	The number of characters written is shorter than the expected record length.
EBADKEYLN	3044	A length that was not valid was specified for the key.
EPUTANDGET	3080	A read operation should not immediately follow a write operation.
EGETANDPUT	3081	A write operation should not immediately follow a read operation.
EIOERROR	3101	A nonrecoverable I/O error occurred.
EIORECERR	3102	A recoverable I/O error occurred.
EACCES	3401	Permission denied.
ENOTDIR	3403	Not a directory.
ENOSPC	3404	No space is available.
EXDEV	3405	Improper link.
EAGAIN	3406	Operation would have caused the process to be suspended.
EWOULDBLOCK	3406	Operation would have caused the process to be suspended.
EINTR	3407	Interrupted function call.
EFAULT	3408	The address used for an argument was not correct.
ETIME	3409	Operation timed out.
ENXIO	3415	No such device or address.
EAPAR	3418	Possible APAR condition or hardware failure.
ERECURSE	3419	Recursive attempt rejected.
EADDRINUSE	3420	Address already in use.
EADDRNOTAVAIL	3421	Address is not available.
EAFNOSUPPORT	3422	The type of socket is not supported in this protocol family.
EALREADY	3423	Operation is already in progress.
ECONNABORTED	3424	Connection ended abnormally.
ECONNREFUSED	3425	A remote host refused an attempted connect operation.
ECONNRESET	3426	A connection with a remote socket was reset by that socket.
EDESTADDRREQ	3427	Operation requires destination address.
EHOSTDOWN	3428	A remote host is not available.
EHOSTUNREACH	3429	A route to the remote host is not available.
EINPROGRESS	3430	Operation in progress.
EISCONN	3431	A connection has already been established.
EMSGSIZE	3432	Message size is out of range.
ENETDOWN	3433	The network currently is not available.
ENETRESET	3434	A socket is connected to a host that is no longer available.

ENETUNREACH	3435	Cannot reach the destination network.
ENOBUFS	3436	There is not enough buffer space for the requested operation.
ENOPROTOPT	3437	The protocol does not support the specified option.
ENOTCONN	3438	Requested operation requires a connection.
ENOTSOCK	3439	The specified descriptor does not reference a socket.
ENOTSUP	3440	Operation is not supported.
EOPNOTSUPP	3440	Operation is not supported.
EPFNOSUPPORT	3441	The socket protocol family is not supported.
EPROTONOSUPPORT	3442	No protocol of the specified type and domain exists.
EPROTOTYPE	3443	The socket type or protocols are not compatible.
ERCVDERR	3444	An error indication was sent by the peer program.
ESHUTDOWN	3445	Cannot send data after a shutdown.
ESOCKTNOSUPPORT	3446	The specified socket type is not supported.
ETIMEDOUT	3447	A remote host did not respond within the timeout period.
EUNATCH	3448	The protocol required to support the specified address family is not available at this time.
EBADF	3450	Descriptor is not valid.
EMFILE	3452	Too many open files for this process.
ENFILE	3453	Too many open files in the system.
EPIPE	3455	Broken pipe.
ECANCEL	3456	Operation cancelled.
EEXIST	3457	File exists.
EDEADLK	3459	Resource deadlock avoided.
ENOMEM	3460	Storage allocation request failed.
EOWNERTERM	3462	The synchronization object no longer exists because the owner is no longer running.
EDESTROYED	3463	The synchronization object was destroyed, or the object no longer exists.
ETERM	3464	Operation was terminated.
ENOENT1	3465	No such file or directory.
ENOEQFLOG	3466	Object is already linked to a dead directory.
EEMPTYDIR	3467	Directory is empty.
EMLINK	3468	Maximum link count for a file was exceeded.

ESPIPE	3469	Seek request is not supported for object.
ENOSYS	3470	Function not implemented.
EISDIR	3471	Specified target is a directory.
EROFS	3472	Read-only file system.
EUNKNOWN	3474	Unknown system state.
EITERBAD	3475	Iterator is not valid.
EITERSTE	3476	Iterator is in wrong state for operation.
EHRICLSBAD	3477	HRI class is not valid.
EHRICLBAD	3478	HRI subclass is not valid.
EHRITYPBAD	3479	HRI type is not valid.
ENOTAPPL	3480	Data requested is not applicable.
EHRIREQTYP	3481	HRI request type is not valid.
EHRINAMEBAD	3482	HRI resource name is not valid.
EDAMAGE	3484	A damaged object was encountered.
ELOOP	3485	A loop exists in the symbolic links.
ENAMETOOLONG	3486	A path name is too long.
ENOLCK	3487	No locks are available.
ENOTEMPTY	3488	Directory is not empty.
ENOSYSRSC	3489	System resources are not available.
ECONVERT	3490	Conversion error.
E2BIG	3491	Argument list is too long.
EILSEQ	3492	Conversion stopped due to input character that does not belong to the input codeset.
ETYPE	3493	Object type mismatch.
EBADDIR	3494	Attempted to reference a directory that was not found or was destroyed.
EBADOBJ	3495	Attempted to reference an object that was not found, was destroyed, or was damaged.
EIDXINVAL	3496	Data space index used as a directory is not valid.
ESOFTDAMAGE	3497	Object has soft damage.
ENOTENROLL	3498	User is not enrolled in system distribution directory.
EOffline	3499	Object is suspended.
EROOBJ	3500	Object is a read-only object.
EEAHDDSI	3501	Hard damage on extended attribute data space index.
EEASDDSI	3502	Soft damage on extended attribute data space index.
EEAHDDS	3503	Hard damage on extended attribute data space.
EEASDDS	3504	Soft damage on extended attribute data space.
EEADUPRC	3505	Duplicate extended attribute record.

ELOCKED	3506	Area being read from or written to is locked.
EFBIG	3507	Object too large.
EIDRM	3509	The semaphore, shared memory, or message queue identifier is removed from the system.
ENOMSG	3510	The queue does not contain a message of the desired type and (msgflg logically ANDed with IPC_NOWAIT).
EFILECVT	3511	File ID conversion of a directory failed.
EBADFID	3512	A file ID could not be assigned when linking an object to a directory.
ESTALE	3513	File handle was rejected by server.
ESRCH	3515	No such process.
ENOTSIGINIT	3516	Process is not enabled for signals.
ECHILD	3517	No child process.
EBADH	3520	Handle is not valid.
ETOOMANYREFS	3523	The operation would have exceeded the maximum number of references allowed for a descriptor.
ENOTSAFE	3524	Function is not allowed.
E_OVERFLOW	3525	Object is too large to process.
EJRNDDAMAGE	3526	Journal is damaged.
EJRNINACTIVE	3527	Journal is inactive.
EJRNRCVSPC	3528	Journal space or system storage error.
EJRNRMNT	3529	Journal is remote.
ENEWJRNRCV	3530	New journal receiver is needed.
ENEWJRN	3531	New journal is needed.
EJOURNALED	3532	Object already journaled.
EJRNENTTOOLONG	3533	Entry is too large to send.
EDATALINK	3534	Object is a datalink object.
ENOTAVAIL	3535	IASP is not available.
ENOTTY	3536	I/O control operation is not appropriate.
EFBIG2	3540	Attempt to write or truncate file past its sort file size limit.
ETXTBSY	3543	Text file busy.
EASPGRPNOTSET	3544	ASP group not set for thread.
ERESTART	3545	A system call was interrupted and may be restarted.