IBM Systems - iSeries
UNIX-Type -- Time APIs

Version 5 Release 4
Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 29.
Time APIs

The Time APIs include two sets of APIs:

- **System Clock APIs**
- **Software Clock APIs** on page 9

Both the System Clock and Software Clock APIs use a timeval structure that is the number of seconds and microseconds since 1 January 1970, 00:00:00 Universal Coordinated Time (UTC).

The System Clock APIs allow you to work with the system clock. The system clock is a system facility that maintains the time of the system. The `gettimeofday()` and `ftime()` APIs get the current time from the system clock. The `adjtime()` and `settimeofday()` APIs set the system clock. See the Time Management topic for more information about the system clock.

The Software Clock APIs allow you to work with the software clock. The software clock is a system facility that maintains a time that is independent from the system clock. The `Qp0zGetTimeofDay()` API gets the current time from the software clock. The `Qp0zAdjTime()` and `Qp0zSetTimeofDay()` APIs set the software clock. In previous releases, the software clock was the only way to get a UTC time in seconds and microseconds. System components do not base their timestamps on the software clock, but use the system clock instead. The software clock will be removed in a future release and its use is discouraged.

APIs

These are the APIs for this category.

System Clock APIs

The System Clock APIs are:

- **adjtime()—Adjust System Clock** (Adjust system clock) makes small adjustments to the software clock, either slowing it down or speeding it up by the time specified in the delta parameter.
- **ftime()—Get Date and Time** on page 4 (Get Date and Time) retrieves the current Coordinated Universal Time (UTC).
- **gettimeofday()—Get Current UTC Time** on page 5 (Get Current UTC Time) retrieves the current Coordinated Universal Time (UTC) and places it in the timeval structure pointed to by tp.
- **settimeofday()—Set System Clock** on page 7 (Set System Clock) sets the system clock to the Coordinated Universal Time (UTC) 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 **"Header Files for UNIX-Type Functions"** on page 16 for the file and member name of each header file.

adftime()—Adjust System Clock

Syntax
#include <sys/time.h>

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

Service Program Name: QWCTZUTC
Default Public Authority: *USE
Threadsafe: Yes

The adjtime() function makes small adjustments to the system clock, either slowing it down or speeding
it up by the time specified in the delta parameter up to a maximum of two hours. 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 for adjusting the clock.

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

Special Authority
    *ALLOBJ

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.

[EACCES] Permission denied.
    An attempt was made to access an object in a way forbidden by its object access permissions.

[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.

[ENOTSUP] Operation not supported.

The operation is not supported.

[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" on page 16)
- "gettimeofday()—Get Current UTC Time" on page 5
- "settimeofday()—Set System Clock" on page 7

Example
See Code disclaimer information for information pertaining to code examples.

The following example initiates a system clock adjustment:

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

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

    /* Speed up the clock by 1.5 seconds. */
    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
ftime()—Get Date and Time

Syntax
#include <sys/timeb.h>

int ftime (struct timeb *tp);

Service Program Name: QWCTZUTC
Default Public Authority: *USE
Threadsafe: Yes

The ftime() function retrieves the current Coordinated Universal Time (UTC) and places it in timeb structure pointed to by tp.

Parameters

tp    (Output) A pointer to a timeb structure that contains the time in seconds and milliseconds since 1 January 1970, 00:00:00 UTC (epoch-1970), the local time zone (measured in minutes west of Greenwich), and a flag that, if nonzero, indicates Daylight Saving Time is currently in effect.

Authorities and Locks
None

Return Value

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

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

[EINV AL] 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/timeb.h> file (see "Header Files for UNIX-Type Functions” on page 16)
• "gettimeofday()—Get Current UTC Time” on page 5
Example

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

The following example gets the current date and time:

```c
#include <sys/timeb.h>
#include <stdio.h>
#include <errno.h>

int main(int argc, char *argv[]) {
    struct timeb now;
    int rc;

    rc=ftime(&now);
    if(rc==0) {
        printf("ftime() successful.\n");
        printf("time = %u.%03u,  
            "timezone = %d,  
            "dstflag = %d\n",
            now.time, now.millitm,
            now.timezone, now.dstflag );
    } else {
        printf("ftime() failed, errno = %d\n",
            errno);
        return -1;
    }

    return 0;
}
```

Example Output:
ftime() successful.
time = 1019833362.226 timezone = 360, dstflag = 1

API introduced: V5R3

---

gettimeofday()—Get Current UTC Time

Syntax

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

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

Service Program Name: QWCTZUTC
Default Public Authority: *USE
Threadsafe: Yes

The `gettimeofday()` function retrieves the current Coordinated Universal Time (UTC) and places it in the `timeval` structure pointed to by `tp`. If `tp` is not NULL, the time zone information is returned in the time zone 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 time zone structure that contains the local time zone (measured in minutes west of 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.

[UNKNOWN] 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.

Usage Notes

• For the best performance, specify NULL for the tzp parameter.

Related Information

• The <sys/time.h> file (see “Header Files for UNIX-Type Functions” on page 16
• “adjtime()—Adjust System Clock” on page 1
• “ftime()—Get Date and Time” on page 4
• Qp0zCvtToMITime()—Convert Timeval Structure to _MI_Time
• “settimeofday()—Set System Clock” on page 7
Example
See Code disclaimer information for information pertaining to code examples.

The following example gets the current UTC time:
```c
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>

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

    rc=gettimeofday(&now, NULL);
    if(rc==0) {
        printf("gettimeofday() successful.
        time = %u.%06u
", now.tv_sec, now.tv_usec);
    } else {
        printf("gettimeofday() failed, errno = %d\n", errno);
        return -1;
    }
    return 0;
}
```
Example Output:
gettimeofday() successful.
time = 866208142.290944

API introduced: V4R2

settimeofday()—Set System Clock

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

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

Service Program Name: QWCTZUTC
Default Public Authority: *USE
Threadsafe: Yes

The `settimeofday()` function sets the system clock to the Coordinated Universal Time (UTC) contained in the `timeval` structure pointed to by `tp`. The `tzp` parameter is not used.

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**  This parameter is not used.
Authorities and Locks

Special Authority
*ALLOBJ

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.

[EACCES] Permission denied.
   An attempt was made to access an object in a way forbidden by its object access permissions.

[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.

[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” on page 16)
• “adjtime()—Adjust System Clock” on page 1
• “gettimeofday()—Get Current UTC Time” on page 5

Example

See Code disclaimer information for information pertaining to code examples.

The following example sets the system clock:

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

int main(int argc, char *argv[]) {
  struct timeval now;
```

IBM Systems - iSeries: UNIX-Type -- Time APIs
int rc;
now.tv_sec=866208142;
now.tv_usec=290944;
rc=settimeofday(&now, NULL);
if(rc==0) {
    printf("settimeofday() successful.");
} else {
    printf("settimeofday() failed, "
            "errno = %d\n",errno);
    return -1;
}
return 0;

Example Output:
settimeofday() successful.

API introduced: V4R2

Software Clock APIs

The Software Clock APIs are:

- "Qp0zAdjTime()—Adjust Software Clock" on page 10 (Adjust Software Clock) makes small adjustments to the software clock, either slowing it down or speeding it up by the time specified in the delta parameter.
- "Qp0zGetTimeofDay()—Get Current Software Clock Time" on page 12 (Get Current Software Clock Time) retrieves the current software clock time and places it in the timeval structure pointed to by tp.
- "Qp0zSetTimeofDay()—Set Software Clock" on page 14 (Set Software Clock) 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 "Header Files for UNIX-Type Functions" on page 16 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). The current time is determined by keeping an internal ‘time-delta’, which is an offset from the system clock. When Qp0zGetTimeofDay() is called to retrieve the software clock time, the time returned is calculated by taking the current system time, subtracting the QUTCOFFSET system value, then adding the internal time-delta.

The Qp0zSetTimeofDay() API sets or changes the time-delta, without affecting the system clock or the QUTCOFFSET system value. The Qp0zAdjTime() API slowly changes the time-delta, without affecting the system clock either. Adjustments are made at a rate of approximately 1 second of adjustment for every 100 seconds of elapsed time.

System components do not base their timestamps on the software clock, but use the system clock instead. The software clock will be removed in a future release and its use is discouraged.
The \texttt{Qp0zAdjTime()} function makes small adjustments to the software clock, either slowing it down or speeding it up by the time specified in the \textit{delta} parameter. If \textit{delta} is negative, the clock is slowed down by incrementing it more slowly than normal until the correction is complete. If \textit{delta} is positive, the clock is sped up by incrementing it more quickly than normal until the correction is complete. If \textit{olddelta} is not NULL, the amount of time still to be corrected from a previous \texttt{Qp0zAdjTime()} call is returned in the structure it points to.

The software clock maintains a time that can be set independently of the system clock. It is not integrated with the system and will be removed in a future release. The \textit{“adjtime()—Adjust System Clock” on page 1} function should be used instead.

**Parameters**

\textit{delta} (Input) A pointer to a timeval structure that contains the amount of time for adjusting the software clock.

\textit{olddelta} (Output) A pointer to a timeval structure that contains the amount of time still to be corrected from a previous call to \texttt{Qp0zAdjTime()}

**Authorities and Locks**

\textit{QSYS/QP0ZXCPO Service Program Authority}  
*USE

**Return Value**

0  \texttt{Qp0zAdjTime()} was successful. The requested adjustment was initiated and the value returned in the structure pointed to by the \textit{olddelta} parameter is the amount of time still to be corrected from a previous \texttt{Qp0zAdjTime()}.  
-1 \texttt{Qp0zAdjTime()} was not successful. The \textit{errno} variable is set to indicate the error.

**Error Conditions**

If \texttt{Qp0zAdjTime()} is not successful, \textit{errno} usually indicates one of the following errors. Under some conditions, \textit{errno} could indicate an error other than those listed here.

\textbf{[EINVAL]} An invalid parameter was found.  
A parameter passed to this function is not valid.
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.

Operation not permitted.

You must have appropriate privileges or be the owner of the object or other resource to do the requested operation.

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.

Usage Notes
If the value of the environment variable QIBM_USE_SFWCLK is “N”, Qp0zAdjTime() calls adjtime() to adjust the system clock.

Related Information
- The <sys/time.h> file (see "Header Files for UNIX-Type Functions” on page 16)
- "adjtime()—Adjust System Clock” on page 1
- “Qp0zGetTimeofDay()—Get Current Software Clock Time” on page 12
- “Qp0zSetTimeofDay()—Set Software Clock” on page 14

Example
See Code disclaimer information for information pertaining to code examples.

The following example initiates a software clock adjustment:

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

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

    /* Speed up the software clock by 1.5 seconds. */
    adj.tv_sec=1;
    adj.tv_usec=500000;
    rc=Qp0zAdjTime(&adj, &old);
    if(rc==0) {
        printf("Qp0zAdjTime() successful. \\
            Old delta = %u.%06u\n", \\
            old.tv_sec, old.tv_usec);
    } else {
        printf("Qp0zAdjTime() failed, errno = %d\n",errno);
        return -1;
    }
}```
Example Output:
Qp0zAdjTime() successful. Olddelta = 0.000000

API introduced: V5R3

Qp0zGetTimeofDay()—Get Current Software Clock Time

Syntax
#include <sys/time.h>

int Qp0zGetTimeofDay (struct timeval *tp,
                      struct timezone *tzp);

Service Program Name: QP0ZCPA
Default Public Authority: USE
Threadsafe: Yes

The Qp0zGetTimeofDay() function retrieves the current software clock 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.

The software clock maintains a time that can be set independently of the system clock. It is not integrated with the system and will be removed in a future release. The "gettimeofday()—Get Current UTC Time" on page 5 function should be used instead.

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 west of 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     Qp0zGetTimeofDay() was successful.
-1     Qp0zGetTimeofDay() was not successful. The errno variable is set to indicate the error.

Error Conditions
If Qp0zGetTimeofDay() 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.

Error Messages
None.

Usage Notes
- For the best performance, specify NULL for the tzp parameter.
- If the value of the environment variable QIBM_USE_SFWCLK is “N”, Qp0zGetTimeofDay() calls gettimeofday() to get the current UTC time from the system clock.

Related Information
- The <sys/time.h> file (see “Header Files for UNIX-Type Functions” on page 16)
- “gettimeofday()—Get Current UTC Time” on page 5
- “Qp0zAdjTime()—Adjust Software Clock” on page 10
- “Qp0zSetTimeofDay()—Set Software Clock” on page 14

Example
See Code disclaimer information for information pertaining to code examples.

The following example gets the current software clock time:

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

int main(int argc, char *argv[]) {
    struct timeval now;
    int rc;
    rc=Qp0zGetTimeofDay(&now, NULL);
    if(rc==0) {
        printf("Qp0zGetTimeofDay() successful.\n");
        printf("time = %u.%06u\n",
                now.tv_sec, now.tv_usec);
    } else {
        printf("Qp0zGetTimeofDay() failed, errno = %d\n",
                errno);
        return -1;
    }
    return 0;
}
```

Example Output:
Qp0zGetTimeofDay() successful.
time = 866208142.290944
Qp0zSetTimeofDay()—Set Software Clock

Syntax
#include <sys/time.h>

int Qp0zSetTimeofDay (struct timeval *tp,
struct timezone *tzp);

Service Program Name: QP0ZSETC
Default Public Authority: *USE
Threadsafe: Yes

The Qp0zSetTimeofDay() 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.

The software clock maintains a time that can be set independently of the system clock. It is not integrated with the system and will be removed in a future release. The “settimeofday()—Set System Clock” on page 7 function should be used instead.

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 west of 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 Qp0zSetTimeofDay() was successful.
-1 Qp0zSetTimeofDay() was not successful. The errno variable is set to indicate the error.

Error Conditions

If Qp0zSetTimeofDay() 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.
Error Messages
None.

Usage Notes
If the value of the environment variable QIBM_USE_SFWCLK is “N”, Qp0zSetTimeofDay() calls settimeofday() to adjust the system clock.

Related Information
- The <sys/time.h> file (see “Header Files for UNIX-Type Functions” on page 16)
- “settimeofday()—Set System Clock” on page 7
- “Qp0zAdjTime()—Adjust Software Clock” on page 10
- “Qp0zGetTimeofDay()—Get Current Software Clock Time” on page 12

Example
See Code disclaimer information for information pertaining to code examples.

The following example sets the software clock:

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

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

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

    rc=Qp0zSetTimeofDay(&now, NULL);
    if(rc==0) {
        printf("Qp0zSetTimeofDay() successful.\n");
    } else {
        printf("Qp0zSetTimeofDay() failed, "
            "errno = %d\n",errno);
        return -1;
    }

    return 0;
}
```

Example Output
Qp0zSetTimeofDay() successful.
API introduced: V5R3

Concepts

These are the concepts for this category.

Header Files for UNIX-Type Functions

Programs using the UNIX(R)-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 [Include files 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.

<table>
<thead>
<tr>
<th>Name of Header File</th>
<th>Name of File in QSYSINC</th>
<th>Name of Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>arpa/inet.h</td>
<td>ARPA</td>
<td>INET</td>
</tr>
<tr>
<td>arpa/nameser.h</td>
<td>ARPA</td>
<td>NAMESER</td>
</tr>
<tr>
<td>bse.h</td>
<td>H</td>
<td>BSE</td>
</tr>
<tr>
<td>bsedos.h</td>
<td>H</td>
<td>BSEDOS</td>
</tr>
<tr>
<td>bseerr.h</td>
<td>H</td>
<td>BSEERR</td>
</tr>
<tr>
<td>dirent.h</td>
<td>H</td>
<td>DIRENT</td>
</tr>
<tr>
<td>errno.h</td>
<td>H</td>
<td>ERRNO</td>
</tr>
<tr>
<td>fcntl.h</td>
<td>H</td>
<td>FCNTL</td>
</tr>
<tr>
<td>grp.h</td>
<td>H</td>
<td>GRP</td>
</tr>
<tr>
<td>inttypes.h</td>
<td>H</td>
<td>INTTYPES</td>
</tr>
<tr>
<td>limits.h</td>
<td>H</td>
<td>LIMITS</td>
</tr>
<tr>
<td>mman.h</td>
<td>H</td>
<td>MMAN</td>
</tr>
<tr>
<td>netdbh.h</td>
<td>H</td>
<td>NETDB</td>
</tr>
<tr>
<td>netinet/icmp6.h</td>
<td>NETINET</td>
<td>ICMP6</td>
</tr>
<tr>
<td>net/if.h</td>
<td>NET</td>
<td>IF</td>
</tr>
<tr>
<td>netinet/in.h</td>
<td>NETINET</td>
<td>IN</td>
</tr>
<tr>
<td>netinet/ip_icmp.h</td>
<td>NETINET</td>
<td>IP_ICMP</td>
</tr>
<tr>
<td>netinet/ip.h</td>
<td>NETINET</td>
<td>IP</td>
</tr>
<tr>
<td>netinet/ip6.h</td>
<td>NETINET</td>
<td>IP6</td>
</tr>
<tr>
<td>netinet/tcp.h</td>
<td>NETINET</td>
<td>TCP</td>
</tr>
<tr>
<td>netinet/udp.h</td>
<td>NETINET</td>
<td>UDP</td>
</tr>
<tr>
<td>netns/idp.h</td>
<td>NETNS</td>
<td>IDP</td>
</tr>
<tr>
<td>Name of Header File</td>
<td>Name of File in QSYSINC</td>
<td>Name of Member</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>netns/ipx.h</td>
<td>NETNS</td>
<td>IPX</td>
</tr>
<tr>
<td>netns/ns.h</td>
<td>NETNS</td>
<td>NS</td>
</tr>
<tr>
<td>netns/sp.h</td>
<td>NETNS</td>
<td>SP</td>
</tr>
<tr>
<td>net/route.h</td>
<td>NET</td>
<td>ROUTE</td>
</tr>
<tr>
<td>nettel/tel.h</td>
<td>NETTEL</td>
<td>TEL</td>
</tr>
<tr>
<td>os2.h</td>
<td>H</td>
<td>OS2</td>
</tr>
<tr>
<td>os2def.h</td>
<td>H</td>
<td>OS2DEF</td>
</tr>
<tr>
<td>pwd.h</td>
<td>H</td>
<td>PWD</td>
</tr>
<tr>
<td>Qlg.h</td>
<td>H</td>
<td>QLG</td>
</tr>
<tr>
<td>qp0lchsg.h</td>
<td>H</td>
<td>QP0LCHSG</td>
</tr>
<tr>
<td>qp0lflop.h</td>
<td>H</td>
<td>QP0LFLOP</td>
</tr>
<tr>
<td>qp0ljrn1.h</td>
<td>H</td>
<td>QP0LJRN1</td>
</tr>
<tr>
<td>qp0lror.h</td>
<td>H</td>
<td>QP0LROR</td>
</tr>
<tr>
<td>qp0lrro.h</td>
<td>H</td>
<td>QP0LRRO</td>
</tr>
<tr>
<td>qp0lrtsg.h</td>
<td>H</td>
<td>QP0LRTSG</td>
</tr>
<tr>
<td>qp0lscaN.h</td>
<td>H</td>
<td>QP0LSCAN</td>
</tr>
<tr>
<td>Qp0lstdi.h</td>
<td>H</td>
<td>QP0LSTDI</td>
</tr>
<tr>
<td>qp0wpid.h</td>
<td>H</td>
<td>QP0WPID</td>
</tr>
<tr>
<td>qp0zdipc.h</td>
<td>H</td>
<td>QP0ZDIPC</td>
</tr>
<tr>
<td>qp0zipc.h</td>
<td>H</td>
<td>QP0ZIPC</td>
</tr>
<tr>
<td>qp0zolip.h</td>
<td>H</td>
<td>QP0ZOLIP</td>
</tr>
<tr>
<td>qp0zolsm.h</td>
<td>H</td>
<td>QP0ZOLSM</td>
</tr>
<tr>
<td>qp0zripc.h</td>
<td>H</td>
<td>QP0ZRIPC</td>
</tr>
<tr>
<td>qp0ztrc.h</td>
<td>H</td>
<td>QP0ZTRC</td>
</tr>
<tr>
<td>qp0ztrml.h</td>
<td>H</td>
<td>QP0ZTRML</td>
</tr>
<tr>
<td>qp0z1170.h</td>
<td>H</td>
<td>QP0Z1170</td>
</tr>
<tr>
<td>qsoasync.h</td>
<td>H</td>
<td>QSOASYNC</td>
</tr>
<tr>
<td>qtnxaapi.h</td>
<td>H</td>
<td>QTNXAAPI</td>
</tr>
<tr>
<td>qtnxadtp.h</td>
<td>H</td>
<td>QTNXADTP</td>
</tr>
<tr>
<td>qtoomeapi.h</td>
<td>H</td>
<td>QTOOMAPI</td>
</tr>
<tr>
<td>qtoossapi.h</td>
<td>H</td>
<td>QTOSSAPI</td>
</tr>
<tr>
<td>resolv.h</td>
<td>H</td>
<td>RESOLVE</td>
</tr>
<tr>
<td>semaphore.h</td>
<td>H</td>
<td>SEMAPHORE</td>
</tr>
<tr>
<td>signal.h</td>
<td>H</td>
<td>SIGNAL</td>
</tr>
<tr>
<td>spawn.h</td>
<td>H</td>
<td>SPAWN</td>
</tr>
<tr>
<td>ssl.h</td>
<td>H</td>
<td>SSL</td>
</tr>
<tr>
<td>sys/errno.h</td>
<td>H</td>
<td>ERRNO</td>
</tr>
<tr>
<td>sys/ioctl.h</td>
<td>SYS</td>
<td>IOCTL</td>
</tr>
<tr>
<td>sys/ipc.h</td>
<td>SYS</td>
<td>IPC</td>
</tr>
<tr>
<td>sys/layout.h</td>
<td>H</td>
<td>LAYOUT</td>
</tr>
<tr>
<td>sys/limits.h</td>
<td>H</td>
<td>LIMITS</td>
</tr>
</tbody>
</table>
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(R)-type functions may receive error information as `errno` values. The possible values returned are listed here in ascending `errno` value sequence.
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Text</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDOM</td>
<td>3001</td>
<td>A domain error occurred in a math function.</td>
<td></td>
</tr>
<tr>
<td>ERANGE</td>
<td>3002</td>
<td>A range error occurred.</td>
<td></td>
</tr>
<tr>
<td>ETRUNC</td>
<td>3003</td>
<td>Data was truncated on an input, output, or update operation.</td>
<td></td>
</tr>
<tr>
<td>ENOTOPEN</td>
<td>3004</td>
<td>File is not open.</td>
<td>You attempted to do an operation that required the file to be open.</td>
</tr>
<tr>
<td>ENOTREAD</td>
<td>3005</td>
<td>File is not opened for read operations.</td>
<td>You tried to read a file that is not open for read operations.</td>
</tr>
<tr>
<td>EIO</td>
<td>3006</td>
<td>Input/output error.</td>
<td>A physical I/O error occurred or a referenced object was damaged.</td>
</tr>
<tr>
<td>ENODEV</td>
<td>3007</td>
<td>No such device.</td>
<td></td>
</tr>
<tr>
<td>ERECIO</td>
<td>3008</td>
<td>Cannot get single character for files opened for record I/O.</td>
<td>The file that was specified is open for record I/O and you attempted to read it as a stream file.</td>
</tr>
<tr>
<td>ENOTWRITE</td>
<td>3009</td>
<td>File is not opened for write operations.</td>
<td>You tried to update a file that has not been opened for write operations.</td>
</tr>
<tr>
<td>ESTDIN</td>
<td>3010</td>
<td>The stdin stream cannot be opened.</td>
<td></td>
</tr>
<tr>
<td>ESTDOUT</td>
<td>3011</td>
<td>The stdout stream cannot be opened.</td>
<td></td>
</tr>
<tr>
<td>ESTDERR</td>
<td>3012</td>
<td>The stderr stream cannot be opened.</td>
<td></td>
</tr>
<tr>
<td>EBADSEEK</td>
<td>3013</td>
<td>The positioning parameter in fseek is not correct.</td>
<td></td>
</tr>
<tr>
<td>EBADNAME</td>
<td>3014</td>
<td>The object name specified is not correct.</td>
<td></td>
</tr>
<tr>
<td>EBADMODE</td>
<td>3015</td>
<td>The type variable specified on the open function is not correct.</td>
<td>The mode that you attempted to open the file in is not correct.</td>
</tr>
<tr>
<td>EBADPOS</td>
<td>3017</td>
<td>The position specifier is not correct.</td>
<td></td>
</tr>
<tr>
<td>ENOPOS</td>
<td>3018</td>
<td>There is no record at the specified position.</td>
<td>You attempted to position to a record that does not exist in the file.</td>
</tr>
<tr>
<td>ENUMMBRS</td>
<td>3019</td>
<td>Attempted to use ftell on multiple members.</td>
<td>Remove all but one member from the file.</td>
</tr>
<tr>
<td>ENUMRECS</td>
<td>3020</td>
<td>The current record position is too long for ftell.</td>
<td></td>
</tr>
<tr>
<td>EINVAL</td>
<td>3021</td>
<td>The value specified for the argument is not correct.</td>
<td>A function was passed incorrect argument values, or an operation was attempted on an object and the operation specified is not supported for that type of object.</td>
</tr>
<tr>
<td>EBADFUNC</td>
<td>3022</td>
<td>Function parameter in the signal function is not set.</td>
<td></td>
</tr>
<tr>
<td>ENOENT</td>
<td>3025</td>
<td>No such path or directory.</td>
<td>The directory or a component of the path name specified does not exist.</td>
</tr>
<tr>
<td>ENOREC</td>
<td>3026</td>
<td>Record is not found.</td>
<td></td>
</tr>
<tr>
<td>EPERM</td>
<td>3027</td>
<td>The operation is not permitted.</td>
<td>You must have appropriate privileges or be the owner of the object or other resource to do the requested operation.</td>
</tr>
<tr>
<td>EBADDATA</td>
<td>3028</td>
<td>Message data is not valid.</td>
<td>The message data that was specified for the error text is not correct.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EBUSY</td>
<td>3029</td>
<td>Resource busy.</td>
<td>An attempt was made to use a system resource that is not available at this time.</td>
</tr>
<tr>
<td>EBAADOPT</td>
<td>3040</td>
<td>Option specified is not valid.</td>
<td></td>
</tr>
<tr>
<td>ENOTUPD</td>
<td>3041</td>
<td>File is not opened for update operations.</td>
<td></td>
</tr>
<tr>
<td>ENOTDLT</td>
<td>3042</td>
<td>File is not opened for delete operations.</td>
<td></td>
</tr>
<tr>
<td>EPAD</td>
<td>3043</td>
<td>The number of characters written is shorter than the expected record length.</td>
<td>The length of the record is longer than the buffer size that was specified. The data written was padded to the length of the record.</td>
</tr>
<tr>
<td>EBADKEYLN</td>
<td>3044</td>
<td>A length that was not valid was specified for the key.</td>
<td>You attempted a record I/O against a keyed file. The key length that was specified is not correct.</td>
</tr>
<tr>
<td>EPUTANDGET</td>
<td>3080</td>
<td>A read operation should not immediately follow a write operation.</td>
<td></td>
</tr>
<tr>
<td>EGETANDPUT</td>
<td>3081</td>
<td>A write operation should not immediately follow a read operation.</td>
<td></td>
</tr>
<tr>
<td>EIOERROR</td>
<td>3101</td>
<td>A nonrecoverable I/O error occurred.</td>
<td></td>
</tr>
<tr>
<td>EIORECERR</td>
<td>3102</td>
<td>A recoverable I/O error occurred.</td>
<td></td>
</tr>
<tr>
<td>EACCES</td>
<td>3401</td>
<td>Permission denied.</td>
<td>An attempt was made to access an object in a way forbidden by its object access permissions.</td>
</tr>
<tr>
<td>ENOTDIR</td>
<td>3403</td>
<td>Not a directory.</td>
<td>A component of the specified path name existed, but it was not a directory when a directory was expected.</td>
</tr>
<tr>
<td>ENOSPC</td>
<td>3404</td>
<td>No space is available.</td>
<td>The requested operations required additional space on the device and there is no space left. This could also be caused by exceeding the user profile storage limit when creating or transferring ownership of an object.</td>
</tr>
<tr>
<td>EXDEV</td>
<td>3405</td>
<td>Improper link.</td>
<td>A link to a file on another file system was attempted.</td>
</tr>
<tr>
<td>EAGAIN</td>
<td>3406</td>
<td>Operation would have caused the process to be suspended.</td>
<td></td>
</tr>
<tr>
<td>EWOULDBLOCK</td>
<td>3406</td>
<td>Operation would have caused the process to be suspended.</td>
<td></td>
</tr>
<tr>
<td>EINTR</td>
<td>3407</td>
<td>Interrupted function call.</td>
<td></td>
</tr>
<tr>
<td>EFAULT</td>
<td>3408</td>
<td>The address used for an argument was not correct.</td>
<td>In attempting to use an argument in a call, the system detected an address that is not valid.</td>
</tr>
<tr>
<td>ETIME</td>
<td>3409</td>
<td>Operation timed out.</td>
<td></td>
</tr>
<tr>
<td>ENXIO</td>
<td>3415</td>
<td>No such device or address.</td>
<td></td>
</tr>
<tr>
<td>EAPAR</td>
<td>3418</td>
<td>Possible APAR condition or hardware failure.</td>
<td></td>
</tr>
<tr>
<td>ERECURRENCE</td>
<td>3419</td>
<td>Recursive attempt rejected.</td>
<td></td>
</tr>
<tr>
<td>EADDRINUSE</td>
<td>3420</td>
<td>Address already in use.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EADDRNOTAVAIL</td>
<td>3421</td>
<td>Address is not available.</td>
<td></td>
</tr>
<tr>
<td>EAFNOSUPPORT</td>
<td>3422</td>
<td>The type of socket is not supported in this protocol family.</td>
<td></td>
</tr>
<tr>
<td>EALREADY</td>
<td>3423</td>
<td>Operation is already in progress.</td>
<td></td>
</tr>
<tr>
<td>ECONNABORTED</td>
<td>3424</td>
<td>Connection ended abnormally.</td>
<td></td>
</tr>
<tr>
<td>ECONNREFUSED</td>
<td>3425</td>
<td>A remote host refused an attempted connect operation.</td>
<td></td>
</tr>
<tr>
<td>ECONNRESET</td>
<td>3426</td>
<td>A connection with a remote socket was reset by that socket.</td>
<td></td>
</tr>
<tr>
<td>EDESTADDRREQ</td>
<td>3427</td>
<td>Operation requires destination address.</td>
<td></td>
</tr>
<tr>
<td>EHOSTDOWN</td>
<td>3428</td>
<td>A remote host is not available.</td>
<td></td>
</tr>
<tr>
<td>EHOSTUNREACH</td>
<td>3429</td>
<td>A route to the remote host is not available.</td>
<td></td>
</tr>
<tr>
<td>EINPROGRESS</td>
<td>3430</td>
<td>Operation in progress.</td>
<td></td>
</tr>
<tr>
<td>EISCONN</td>
<td>3431</td>
<td>A connection has already been established.</td>
<td></td>
</tr>
<tr>
<td>EMSGSIZE</td>
<td>3432</td>
<td>Message size is out of range.</td>
<td></td>
</tr>
<tr>
<td>ENETDOWN</td>
<td>3433</td>
<td>The network currently is not available.</td>
<td></td>
</tr>
<tr>
<td>ENETRESET</td>
<td>3434</td>
<td>A socket is connected to a host that is no longer available.</td>
<td></td>
</tr>
<tr>
<td>ENETUNREACH</td>
<td>3435</td>
<td>Cannot reach the destination network.</td>
<td></td>
</tr>
<tr>
<td>ENOBUFS</td>
<td>3436</td>
<td>There is not enough buffer space for the requested operation.</td>
<td></td>
</tr>
<tr>
<td>ENOPROTOOPT</td>
<td>3437</td>
<td>The protocol does not support the specified option.</td>
<td></td>
</tr>
<tr>
<td>ENOTCONN</td>
<td>3438</td>
<td>Requested operation requires a connection.</td>
<td></td>
</tr>
<tr>
<td>ENOTSOCK</td>
<td>3439</td>
<td>The specified descriptor does not reference a socket.</td>
<td></td>
</tr>
<tr>
<td>ENOTSUP</td>
<td>3440</td>
<td>Operation is not supported.</td>
<td>The operation, though supported in general, is not supported for the requested object or the requested arguments.</td>
</tr>
<tr>
<td>EOPNOTSUPP</td>
<td>3440</td>
<td>Operation is not supported.</td>
<td>The operation, though supported in general, is not supported for the requested object or the requested arguments.</td>
</tr>
<tr>
<td>EPFNOSUPPORT</td>
<td>3441</td>
<td>The socket protocol family is not supported.</td>
<td></td>
</tr>
<tr>
<td>EPROTONOSUPPORT</td>
<td>3442</td>
<td>No protocol of the specified type and domain exists.</td>
<td></td>
</tr>
<tr>
<td>EPROTOTYPE</td>
<td>3443</td>
<td>The socket type or protocols are not compatible.</td>
<td></td>
</tr>
<tr>
<td>ERCVDERR</td>
<td>3444</td>
<td>An error indication was sent by the peer program.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>----------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ESHUTDOWN</td>
<td>3445</td>
<td>Cannot send data after a shutdown.</td>
<td></td>
</tr>
<tr>
<td>ESOCKTNOSUPPORT</td>
<td>3446</td>
<td>The specified socket type is not supported.</td>
<td></td>
</tr>
<tr>
<td>ETIMEDOUT</td>
<td>3447</td>
<td>A remote host did not respond within the timeout period.</td>
<td></td>
</tr>
<tr>
<td>EUNATCH</td>
<td>3448</td>
<td>The protocol required to support the specified address family is not available at this time.</td>
<td></td>
</tr>
<tr>
<td>EBADF</td>
<td>3450</td>
<td>Descriptor is not valid.</td>
<td>A file descriptor argument was out of range, referred to a file that was not open, or a read or write request was made to a file that is not open for that operation.</td>
</tr>
<tr>
<td>EMFILE</td>
<td>3452</td>
<td>Too many open files for this process.</td>
<td>An attempt was made to open more files than allowed by the value of OPEN_MAX. The value of OPEN_MAX can be retrieved using the sysconf() function.</td>
</tr>
<tr>
<td>ENFILE</td>
<td>3453</td>
<td>Too many open files in the system.</td>
<td>A system limit has been reached for the number of files that are allowed to be concurrently open in the system.</td>
</tr>
<tr>
<td>EPIPE</td>
<td>3455</td>
<td>Broken pipe.</td>
<td></td>
</tr>
<tr>
<td>ECANCEL</td>
<td>3456</td>
<td>Operation cancelled.</td>
<td></td>
</tr>
<tr>
<td>EEXIST</td>
<td>3457</td>
<td>Object exists.</td>
<td>The object specified already exists and the specified operation requires that it not exist.</td>
</tr>
<tr>
<td>EDEADLK</td>
<td>3459</td>
<td>Resource deadlock avoided.</td>
<td>An attempt was made to lock a system resource that would have resulted in a deadlock situation. The lock was not obtained.</td>
</tr>
<tr>
<td>ENOMEM</td>
<td>3460</td>
<td>Storage allocation request failed.</td>
<td>A function needed to allocate storage, but no storage is available.</td>
</tr>
<tr>
<td>EOWNERTERM</td>
<td>3462</td>
<td>The synchronization object no longer exists because the owner is no longer running.</td>
<td>The process that had locked the mutex is no longer running, so the mutex was deleted.</td>
</tr>
<tr>
<td>EDESTROYED</td>
<td>3463</td>
<td>The synchronization object was destroyed, or the object no longer exists.</td>
<td></td>
</tr>
<tr>
<td>ETERM</td>
<td>3464</td>
<td>Operation was terminated.</td>
<td></td>
</tr>
<tr>
<td>ENOENT1</td>
<td>3465</td>
<td>No such file or directory.</td>
<td>A component of a specified path name did not exist, or the path name was an empty string.</td>
</tr>
<tr>
<td>ENOEQFLOG</td>
<td>3466</td>
<td>Object is already linked to a dead directory.</td>
<td>The link as a dead option was specified, but the object is already marked as dead. Only one dead link is allowed for an object.</td>
</tr>
<tr>
<td>EEMPTYDIR</td>
<td>3467</td>
<td>Directory is empty.</td>
<td>A directory with entries of only dot and dot-dot was supplied when a nonempty directory was expected.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EMLINK</td>
<td>3468</td>
<td>Maximum link count for a file was exceeded.</td>
<td>An attempt was made to have the link count of a single file exceed LINK_MAX. The value of LINK_MAX can be determined using the pathconf() or the fpathconf() function.</td>
</tr>
<tr>
<td>ESPIPE</td>
<td>3469</td>
<td>Seek request is not supported for object.</td>
<td>A seek request was specified for an object that does not support seeking.</td>
</tr>
<tr>
<td>ENOSYS</td>
<td>3470</td>
<td>Function not implemented.</td>
<td>An attempt was made to use a function that is not available in this implementation for any object or any arguments.</td>
</tr>
<tr>
<td>EISDIR</td>
<td>3471</td>
<td>Specified target is a directory.</td>
<td>The path specified named a directory where a file or object name was expected.</td>
</tr>
<tr>
<td>EROFS</td>
<td>3472</td>
<td>Read-only file system.</td>
<td>You have attempted an update operation in a file system that only supports read operations.</td>
</tr>
<tr>
<td>EUNKNOWN</td>
<td>3474</td>
<td>Unknown system state.</td>
<td>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.</td>
</tr>
<tr>
<td>EOTERBAD</td>
<td>3475</td>
<td>Iterator is not valid.</td>
<td></td>
</tr>
<tr>
<td>EOTERSTE</td>
<td>3476</td>
<td>Iterator is in wrong state for operation.</td>
<td></td>
</tr>
<tr>
<td>EHRICLSBAD</td>
<td>3477</td>
<td>HRI class is not valid.</td>
<td></td>
</tr>
<tr>
<td>EHRICLBAD</td>
<td>3478</td>
<td>HRI subclass is not valid.</td>
<td></td>
</tr>
<tr>
<td>EHRITYPBAD</td>
<td>3479</td>
<td>HRI type is not valid.</td>
<td></td>
</tr>
<tr>
<td>ENOTAPPL</td>
<td>3480</td>
<td>Data requested is not applicable.</td>
<td></td>
</tr>
<tr>
<td>EHRIREQTP</td>
<td>3481</td>
<td>HRI request type is not valid.</td>
<td></td>
</tr>
<tr>
<td>EHRINAMEBAD</td>
<td>3482</td>
<td>HRI resource name is not valid.</td>
<td></td>
</tr>
<tr>
<td>EDAMAGE</td>
<td>3484</td>
<td>A damaged object was encountered.</td>
<td></td>
</tr>
<tr>
<td>ELOOP</td>
<td>3485</td>
<td>A loop exists in the symbolic links.</td>
<td>This error is issued if the number of symbolic links encountered is more than POSIX_SYMLOOP (defined in the limits.h header file). Symbolic links are encountered during resolution of the directory or path name.</td>
</tr>
<tr>
<td>ENAMETOOLONG</td>
<td>3486</td>
<td>A path name is too long.</td>
<td>A path name is longer than PATH_MAX characters or some component of the name is longer than NAME_MAX characters while _POSIX_NO_TRUNC is in effect. For symbolic links, the length of the name string substituted for a symbolic link exceeds PATH_MAX. The PATH_MAX and NAME_MAX values can be determined using the pathconf() function.</td>
</tr>
<tr>
<td>ENOLCK</td>
<td>3487</td>
<td>No locks are available.</td>
<td>A system-imposed limit on the number of simultaneous file and record locks was reached, and no more were available at that time.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ENOTEMPTY</td>
<td>3488</td>
<td>Directory is not empty.</td>
<td>You tried to remove a directory that is not empty. A directory cannot contain objects when it is being removed.</td>
</tr>
<tr>
<td>ENOSYSRSC</td>
<td>3489</td>
<td>System resources are not available.</td>
<td></td>
</tr>
<tr>
<td>ECONVERT</td>
<td>3490</td>
<td>Conversion error.</td>
<td>One or more characters could not be converted from the source CCSID to the target CCSID.</td>
</tr>
<tr>
<td>E2BIG</td>
<td>3491</td>
<td>Argument list is too long.</td>
<td></td>
</tr>
<tr>
<td>EILSEQ</td>
<td>3492</td>
<td>Conversion stopped due to input character that does not belong to the input codeset.</td>
<td></td>
</tr>
<tr>
<td>ETYPE</td>
<td>3493</td>
<td>Object type mismatch.</td>
<td>The type of the object referenced by a descriptor does not match the type specified on the interface.</td>
</tr>
<tr>
<td>EBADDIR</td>
<td>3494</td>
<td>Attempted to reference a directory that was not found or was destroyed.</td>
<td></td>
</tr>
<tr>
<td>EBADOBJ</td>
<td>3495</td>
<td>Attempted to reference an object that was not found, was destroyed, or was damaged.</td>
<td></td>
</tr>
<tr>
<td>EIDXINVAL</td>
<td>3496</td>
<td>Data space index used as a directory is not valid.</td>
<td></td>
</tr>
<tr>
<td>ESOFTDAMAGE</td>
<td>3497</td>
<td>Object has soft damage.</td>
<td></td>
</tr>
<tr>
<td>ENOTENROLL</td>
<td>3498</td>
<td>User is not enrolled in system distribution directory.</td>
<td>You attempted to use a function that requires you to be enrolled in the system distribution directory and you are not.</td>
</tr>
<tr>
<td>EOFFLINE</td>
<td>3499</td>
<td>Object is suspended.</td>
<td>You have attempted to use an object that has had its data saved and the storage associated with it freed. An attempt to retrieve the object’s data failed. The object’s data cannot be used until it is successfully restored. The object’s data was saved and freed either by saving the object with the STG(*FREE) parameter, or by calling an API.</td>
</tr>
<tr>
<td>EROOBJ</td>
<td>3500</td>
<td>Object is read-only.</td>
<td>You have attempted to update an object that can be read only.</td>
</tr>
<tr>
<td>EEAHDDSI</td>
<td>3501</td>
<td>Hard damage on extended attribute data space index.</td>
<td></td>
</tr>
<tr>
<td>EEAHDDSI</td>
<td>3502</td>
<td>Soft damage on extended attribute data space index.</td>
<td></td>
</tr>
<tr>
<td>EEAHDDS</td>
<td>3503</td>
<td>Hard damage on extended attribute data space.</td>
<td></td>
</tr>
<tr>
<td>EEAHDDS</td>
<td>3504</td>
<td>Soft damage on extended attribute data space.</td>
<td></td>
</tr>
<tr>
<td>Eeaduprc</td>
<td>3505</td>
<td>Duplicate extended attribute record.</td>
<td></td>
</tr>
<tr>
<td>ELOCKED</td>
<td>3506</td>
<td>Area being read from or written to is locked.</td>
<td>The read or write of an area conflicts with a lock held by another process.</td>
</tr>
<tr>
<td>EF BIG</td>
<td>3507</td>
<td>Object too large.</td>
<td>The size of the object would exceed the system allowed maximum size.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EIDRM</td>
<td>3509</td>
<td>The semaphore, shared memory, or message queue identifier is removed from the system.</td>
<td></td>
</tr>
<tr>
<td>ENOMSG</td>
<td>3510</td>
<td>The queue does not contain a message of the desired type and (msgflg logically ANDed with IPC_NOWAIT).</td>
<td>To recover from this error, run the Reclaim Storage (RCLSTG) command as soon as possible.</td>
</tr>
<tr>
<td>EFILECVT</td>
<td>3511</td>
<td>File ID conversion of a directory failed.</td>
<td>To recover from this error, run the Reclaim Storage (RCLSTG) command as soon as possible.</td>
</tr>
<tr>
<td>EBADFID</td>
<td>3512</td>
<td>A file ID could not be assigned when linking an object to a directory.</td>
<td>The file ID table is missing or damaged. To recover from this error, run the Reclaim Storage (RCLSTG) command as soon as possible.</td>
</tr>
<tr>
<td>ESTALE</td>
<td>3513</td>
<td>File or object handle rejected by server.</td>
<td></td>
</tr>
<tr>
<td>ESRCH</td>
<td>3515</td>
<td>No such process.</td>
<td></td>
</tr>
<tr>
<td>ENOTSIGNIT</td>
<td>3516</td>
<td>Process is not enabled for signals.</td>
<td>An attempt was made to call a signal function under one of the following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The signal function is being called for a process that is not enabled for asynchronous signals.</td>
<td>• The signal function is being called when the system signal controls have not been initialized.</td>
</tr>
<tr>
<td>ECHILD</td>
<td>3517</td>
<td>No child process.</td>
<td></td>
</tr>
<tr>
<td>EBADH</td>
<td>3520</td>
<td>Handle is not valid.</td>
<td></td>
</tr>
<tr>
<td>ETOOMANYREFS</td>
<td>3523</td>
<td>The operation would have exceeded the maximum number of references allowed for a descriptor.</td>
<td></td>
</tr>
<tr>
<td>ENOTSAFE</td>
<td>3524</td>
<td>Function is not allowed.</td>
<td>Function is not allowed in a job that is running with multiple threads.</td>
</tr>
<tr>
<td>EOVERFLOW</td>
<td>3525</td>
<td>Object is too large to process.</td>
<td>The object’s data size exceeds the limit allowed by this function.</td>
</tr>
<tr>
<td>EJRNDAMAGE</td>
<td>3526</td>
<td>Journal is damaged.</td>
<td>A journal or all of the journal’s attached journal receivers are damaged, or the journal sequence number has exceeded the maximum value allowed. This error occurs during operations that were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>EJRNINACTIVE</td>
<td>3527</td>
<td>Journal is inactive.</td>
<td>The journaling state for the journal is *INACTIVE. This error occurs during operations that were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Text</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EJRNRCVSPC</td>
<td>3528</td>
<td>Journal space or system storage error.</td>
<td>The attached journal receiver does not have space for the entry because the storage limit has been exceeded for the system, the object, the user profile, or the group profile. This error occurs during operations that were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>EJRNRMRT</td>
<td>3529</td>
<td>Journal is remote.</td>
<td>The journal is a remote journal. Journal entries cannot be sent to a remote journal. This error occurs during operations that were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>ENEWJRNRCV</td>
<td>3530</td>
<td>New journal receiver is needed.</td>
<td>A new journal receiver must be attached to the journal before entries can be journaled. This error occurs during operations that were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>ENEWJRNY</td>
<td>3531</td>
<td>New journal is needed.</td>
<td>The journal was not completely created, or an attempt to delete it did not complete successfully. This error occurs during operations that were attempting to start or end journaling, or were attempting to send an entry to the journal.</td>
</tr>
<tr>
<td>EJOURNALED</td>
<td>3532</td>
<td>Object already journaled.</td>
<td>A start journaling operation was attempted on an object that is already being journaled.</td>
</tr>
<tr>
<td>EJRNENTTOOLONG</td>
<td>3533</td>
<td>Entry is too large to send.</td>
<td>The journal entry generated by this operation is too large to send to the journal.</td>
</tr>
<tr>
<td>EDATAALINK</td>
<td>3534</td>
<td>Object is a datalink object.</td>
<td></td>
</tr>
<tr>
<td>ENOTAVAIL</td>
<td>3535</td>
<td>Independent Auxiliary Storage Pool (ASP) is not available.</td>
<td>The independent ASP is in Vary Configuration (VRYCFG) or Reclaim Storage (RCLSTG) processing. To recover from this error, wait until processing has completed for the independent ASP.</td>
</tr>
<tr>
<td>ENOTTY</td>
<td>3536</td>
<td>I/O control operation is not appropriate.</td>
<td></td>
</tr>
<tr>
<td>EFBIG2</td>
<td>3540</td>
<td>Attempt to write or truncate file past its sort file size limit.</td>
<td></td>
</tr>
<tr>
<td>ETXTBSSY</td>
<td>3543</td>
<td>Text file busy.</td>
<td>An attempt was made to execute an i5/OS PASE program that is currently open for writing, or an attempt has been made to open for writing an i5/OS PASE program that is being executed.</td>
</tr>
<tr>
<td>EASPGRPNOTSET</td>
<td>3544</td>
<td>ASP group not set for thread.</td>
<td></td>
</tr>
<tr>
<td>ERESTART</td>
<td>3545</td>
<td>A system call was interrupted and may be restarted.</td>
<td>An object has been marked as a scan failure due to processing by an exit program associated with the scan-related integrated file system exit points.</td>
</tr>
<tr>
<td>ESCANFAILURE</td>
<td>3546</td>
<td>Object had scan failure.</td>
<td></td>
</tr>
</tbody>
</table>
Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This Application Programming Interfaces (API) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.
Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
Advanced 36
Advanced Function Printing
Advanced Peer-to-Peer Networking
AFP
AIX
AS/400
COBOL/400
CUA
DB2
DB2 Universal Database
Distributd Relational Database Architecture
Domino
DPI
DRDA
eServer
GDDM
IBM
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
i5/OS
iSeries
Lotus Notes
MVS
Netfinity
Net.Data
NetView
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
PowerPC
PrintManager
Print Services Facility
RISC System/6000
RPG/400
RS/6000
SAA
SecureWay
System/36
System/370
System/38
System/390
VisualAge
WebSphere
xSeries

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

**Terms and Conditions**

Permissions for the use of these Publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.