:mod:`msvcrt` --- Useful routines from the MS VC++ runtime

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 1**); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 4**)

Unknown directive type "module".

```
.. module:: msvcrt
   :platform: Windows
   :synopsis: Miscellaneous useful routines from the MS VC++ runtime.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 8**)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Fred L. Drake, Jr. <fdrake@acm.org>
```

These functions provide access to some useful capabilities on Windows platforms. Some higher-level modules use these functions to build the Windows implementations of their services. For example, the :mod:`getpass` module uses this in the implementation of the :func:`getpass` function.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 12**); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 12**); *backlink*

Unknown interpreted text role "func".

Further documentation on these functions can be found in the Platform API documentation.

The module implements both the normal and wide char variants of the console I/O api. The normal API deals only with ASCII characters and is of limited use for internationalized applications. The wide char API should be used where ever possible.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 25**)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   Operations in this module now raise :exc:`OSError` where :exc:`IOError`
   was raised.
```

File Operations

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`, **line 36**)

Unknown directive type "function".

```
.. function:: locking(fd, mode, nbytes)

   Lock part of a file based on file descriptor *fd* from the C runtime.  Raises
```

:exc:`OSError` on failure. The locked region of the file extends from the
current file position for *nbytes* bytes, and may continue beyond the end of the
file. *mode* must be one of the :const:`LK_\*` constants listed below. Multiple
regions in a file may be locked at the same time, but may not overlap. Adjacent
regions are not merged; they must be unlocked individually.

```
.. audit-event:: msvcrt.locking fd,mode,nbytes msvcrt.locking
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`
`main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`**, line 48)**

Unknown directive type "data".

```
.. data:: LK_LOCK
          LK_RLCK

   Locks the specified bytes. If the bytes cannot be locked, the program
   immediately tries again after 1 second.  If, after 10 attempts, the bytes cannot
   be locked, :exc:`OSError` is raised.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`
`main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`**, line 56)**

Unknown directive type "data".

```
.. data:: LK_NBLCK
          LK_NBRLCK

   Locks the specified bytes. If the bytes cannot be locked, :exc:`OSError` is
   raised.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`
`main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`**, line 63)**

Unknown directive type "data".

```
.. data:: LK_UNLCK

   Unlocks the specified bytes, which must have been previously locked.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`
`main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`**, line 68)**

Unknown directive type "function".

```
.. function:: setmode(fd, flags)

   Set the line-end translation mode for the file descriptor *fd*. To set it to
   text mode, *flags* should be :const:`os.O_TEXT`; for binary, it should be
   :const:`os.O_BINARY`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`
`main\Doc\library\(cpython-main)(Doc)(library)msvcrt.rst`**, line 75)**

Unknown directive type "function".

```
.. function:: open_osfhandle(handle, flags)

   Create a C runtime file descriptor from the file handle *handle*.  The *flags*
   parameter should be a bitwise OR of :const:`os.O_APPEND`, :const:`os.O_RDONLY`,
   and :const:`os.O_TEXT`.  The returned file descriptor may be used as a parameter
   to :func:`os.fdopen` to create a file object.

   .. audit-event:: msvcrt.open_osfhandle handle,flags msvcrt.open_osfhandle
```

```
.. function:: get_osfhandle(fd)

   Return the file handle for the file descriptor *fd*.  Raises :exc:`OSError` if
   *fd* is not recognized.

   .. audit-event:: msvcrt.get_osfhandle fd msvcrt.get_osfhandle
```

Console I/O

```
.. function:: kbhit()

   Return ``True`` if a keypress is waiting to be read.
```

```
.. function:: getch()

   Read a keypress and return the resulting character as a byte string.
   Nothing is echoed to the console.  This call will block if a keypress
   is not already available, but will not wait for :kbd:`Enter` to be
   pressed. If the pressed key was a special function key, this will
   return ``'\000'`` or ``'\xe0'``; the next call will return the keycode.
   The :kbd:`Control-C` keypress cannot be read with this function.
```

```
.. function:: getwch()

   Wide char variant of :func:`getch`, returning a Unicode value.
```

```
.. function:: getche()

   Similar to :func:`getch`, but the keypress will be echoed if it  represents a
   printable character.
```

```
.. function:: getwche()

   Wide char variant of :func:`getche`, returning a Unicode value.
```

Other Functions