

```
* Please refer to the generic specification for details
*/
```

---

### 11.3.75. wordexp.h

```
/*
 * This header is architecture neutral
 * Please refer to the generic specification for details
 */
```

---

## 11.4. Interface Definitions for libc

### Table of Contents

[nldbl\\_fprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_fwprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_printf\\_chk](#) -- format and print data, with stack checking (64-bit version)  
[nldbl\\_snprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_sprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_swprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_syslog\\_chk](#) -- send messages to the system logger, with stack checking (64-bit version)  
[nldbl\\_vfprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_vfwprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_vprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_vsnprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_vsprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)  
[nldbl\\_vswprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_vsyslog\\_chk](#) -- send messages to the system logger, with stack checking (64-bit version)  
[nldbl\\_vwprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_wprintf\\_chk](#) -- convert formatted wide-character output, with stack checking (64-bit version)  
[nldbl\\_asprintf](#) -- print to allocated string (64-bit version)  
[nldbl\\_dprintf](#) -- print to a file descriptor (64-bit version)  
[nldbl\\_fprintf](#) -- convert formatted output (64-bit version)  
[nldbl\\_fscanf](#) -- input format conversion (64-bit version)  
[nldbl\\_fwprintf](#) -- convert formatted wide-character output (64-bit version)  
[nldbl\\_fwscanf](#) -- convert formatted wide-character input (64-bit version)  
[nldbl\\_printf](#) -- format and print data (64-bit version)  
[nldbl\\_scanf](#) -- input format conversion (64-bit version)  
[nldbl\\_snprintf](#) -- convert formatted output (64-bit version)  
[nldbl\\_sprintf](#) -- convert formatted output (64-bit version)

[\\_\\_nldbl\\_sscanf](#) -- input format conversion (64-bit version)  
[\\_\\_nldbl\\_strfmon](#) -- convert monetary value to a string (64-bit version)  
[\\_\\_nldbl\\_swprintf](#) -- convert formatted wide-character output (64-bit version)  
[\\_\\_nldbl\\_swscanf](#) -- convert formatted wide-character input (64-bit version)  
[\\_\\_nldbl\\_syslog](#) -- send messages to the system logger (64-bit version)  
[\\_\\_nldbl\\_vasprintf](#) -- print to allocated string (64-bit version)  
[\\_\\_nldbl\\_vdprintf](#) -- print to a file descriptor (64-bit version)  
[\\_\\_nldbl\\_vfprintf](#) -- convert formatted output (64-bit version)  
[\\_\\_nldbl\\_vfscanf](#) -- input format conversion (64-bit version)  
[\\_\\_nldbl\\_vfwprintf](#) -- convert formatted wide-character output (64-bit version)  
[\\_\\_nldbl\\_vfwscanf](#) -- wide-character formatted input of a stdarg argument list (64-bit version)  
[\\_\\_nldbl\\_vprintf](#) -- convert formatted output (64-bit version)  
[\\_\\_nldbl\\_vscanf](#) -- input format conversion (64-bit version)  
[\\_\\_nldbl\\_vsnprintf](#) -- convert formatted output (64-bit version)  
[\\_\\_nldbl\\_vsprintf](#) -- convert formatted output (64-bit version)  
[\\_\\_nldbl\\_vsscanf](#) -- input format conversion (64-bit version)  
[\\_\\_nldbl\\_vswprintf](#) -- convert formatted wide-character output (64-bit version)  
[\\_\\_nldbl\\_vswscanf](#) -- wide-character formatted input of a stdarg argument list (64-bit version)  
[\\_\\_nldbl\\_vsyslog](#) -- send messages to the system logger (64-bit version)  
[\\_\\_nldbl\\_vwprintf](#) -- convert formatted wide-character output (64-bit version)  
[\\_\\_nldbl\\_vwscanf](#) -- wide-character formatted input of a stdarg argument list (64-bit version)  
[\\_\\_nldbl\\_wprintf](#) -- convert formatted wide-character output (64-bit version)  
[\\_\\_nldbl\\_wscanf](#) -- convert formatted wide-character input (64-bit version)

The interfaces defined on the following pages are included in libc and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 11.2](#) shall behave as described in the referenced base document. For interfaces referencing LSB and not listed below, please see the generic part of the specification.

## **\_\_nldbl\_\_fprintf\_chk**

### **Name**

[\\_\\_nldbl\\_\\_fprintf\\_chk](#) -- convert formatted output, with stack checking (64-bit version)

### **Synopsis**

```
int __nldbl__fprintf_chk(FILE * stream, int flag, const char * format);
```

### **Description**

The interface [\\_\\_nldbl\\_\\_fprintf\\_chk\(\)](#) shall function in the same way as the interface [\\_\\_fprintf\\_chk\(\)](#), except that [\\_\\_nldbl\\_\\_fprintf\\_chk\(\)](#) shall accept a 64-bit long double value

(also known as double long double) where `__fprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__fprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_\_fwprintf\_chk

## Name

`__nldbl__fwprintf_chk` -- convert formatted wide-character output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__fwprintf_chk(FILE * stream, int flag, const wchar_t * format);
```

## Description

The interface `__nldbl__fwprintf_chk()` shall function in the same way as the interface `__fwprintf_chk()`, except that `__nldbl__fwprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__fwprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__fwprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_\_printf\_chk

## Name

`__nldbl__printf_chk` -- format and print data, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__printf_chk(int flag, const char * format);
```

## Description

The interface `__nldbl__printf_chk()` shall function in the same way as the interface `__printf_chk()`, except that `__nldbl__printf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__printf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__printf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# `__nldbl__snprintf_chk`

## Name

`__nldbl__snprintf_chk` -- convert formatted output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__snprintf_chk(char * str, size_t maxlen, int flag, size_t strlen, const char * format);
```

## Description

The interface `__nldbl__snprintf_chk()` shall function in the same way as the interface `__snprintf_chk()`, except that `__nldbl__snprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__snprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__snprintf_chk()` function is not in the source standard; it is only in the binary

standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_\_sprintf\_chk

## Name

`__nldbl__sprintf_chk` -- convert formatted output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__sprintf_chk(char * str, int flag, size_t strlen, const char * format);
```

## Description

The interface `__nldbl__sprintf_chk()` shall function in the same way as the interface `__sprintf_chk()`, except that `__nldbl__sprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__sprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__sprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_\_swprintf\_chk

## Name

`__nldbl__swprintf_chk` -- convert formatted wide-character output, with stack checking

(64-bit version)

## Synopsis

```
int __nldbl___swprintf_chk(wchar_t * str, size_t n, int flag, size_t strlen, const wchar_t * format);
```

## Description

The interface `__nldbl___swprintf_chk()` shall function in the same way as the interface `__swprintf_chk()`, except that `__nldbl___swprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__swprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl___swprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_\_\_syslog\_chk

## Name

`__nldbl___syslog_chk` -- send messages to the system logger, with stack checking (64-bit version)

## Synopsis

```
void __nldbl___syslog_chk(int priority, int flag, const char * format);
```

## Description

The interface `__nldbl___syslog_chk()` shall function in the same way as the interface `__syslog_chk()`, except that `__nldbl___syslog_chk()` shall accept a 64-bit long double value (also known as double long double) where `__syslog_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl___syslog_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

### **`__nldbl__vfprintf_chk`**

#### **Name**

`__nldbl__vfprintf_chk` -- convert formatted output, with stack checking (64-bit version)

#### **Synopsis**

```
int __nldbl__vfprintf_chk(FILE * str, int flag, const char * format, va_list ap);
```

#### **Description**

The interface `__nldbl__vfprintf_chk()` shall function in the same way as the interface `__vfprintf_chk()`, except that `__nldbl__vfprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vfprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__vfprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

### **`__nldbl__vfwprintf_chk`**

#### **Name**

`__nldbl__vfwprintf_chk` -- convert formatted wide-character output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl___vfwprintf_chk(FILE * str, int flag, const wchar_t * format, va_list ap);
```

## Description

The interface `__nldbl___vfwprintf_chk()` shall function in the same way as the interface `__vfwprintf_chk()`, except that `__nldbl___vfwprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vfwprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl___vfwprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## \_\_nldbl\_\_\_vprintf\_chk

### Name

`__nldbl___vprintf_chk` -- convert formatted output, with stack checking (64-bit version)

### Synopsis

```
int __nldbl___vprintf_chk(int flag, const char * format, va_list ap);
```

### Description

The interface `__nldbl___vprintf_chk()` shall function in the same way as the interface `__vprintf_chk()`, except that `__nldbl___vprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl___vprintf_chk()` function is not in the source standard; it is only in the binary standard.

### Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long



double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **`__nldbl__vsnprintf_chk`**

### **Name**

`__nldbl__vsnprintf_chk` -- convert formatted output, with stack checking (64-bit version)

### **Synopsis**

```
int __nldbl__vsnprintf_chk(char * str, size_t maxlen, int flag, size_t strlen, const char
* format, va_list ap);
```

### **Description**

The interface `__nldbl__vsnprintf_chk()` shall function in the same way as the interface `__vsnprintf_chk()`, except that `__nldbl__vsnprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vsnprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__vsnprintf_chk()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **`__nldbl__vsprintf_chk`**

### **Name**

`__nldbl__vsprintf_chk` -- convert formatted output, with stack checking (64-bit version)

### **Synopsis**

```
int __nldbl__vsprintf_chk(char * str, int flag, size_t strlen, const char * format,
```

```
va_list ap);
```

## Description

The interface `__nldbl__vsprintf_chk()` shall function in the same way as the interface `__vsprintf_chk()`, except that `__nldbl__vsprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vsprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__vsprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# `__nldbl__vswprintf_chk`

## Name

`__nldbl__vswprintf_chk` -- convert formatted wide-character output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__vswprintf_chk(wchar_t * str, size_t maxlen, int flag, size_t strlen, const
wchar_t * format, va_list ap);
```

## Description

The interface `__nldbl__vswprintf_chk()` shall function in the same way as the interface `__vswprintf_chk()`, except that `__nldbl__vswprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vswprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__vswprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this

architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **`__nldbl___vsyslog_chk`**

### **Name**

`__nldbl___vsyslog_chk` -- send messages to the system logger, with stack checking (64-bit version)

### **Synopsis**

```
void __nldbl___vsyslog_chk(int priority, int flag, const char * format, va_list ap);
```

### **Description**

The interface `__nldbl___vsyslog_chk()` shall function in the same way as the interface `__vsyslog_chk()`, except that `__nldbl___vsyslog_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vsyslog_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl___vsyslog_chk()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **`__nldbl___vwprintf_chk`**

### **Name**

`__nldbl___vwprintf_chk` -- convert formatted wide-character output, with stack checking (64-bit version)

### **Synopsis**

```
int __nldbl___vwprintf_chk(int flag, const wchar_t * format, va_list ap);
```

## Description

The interface `__nldbl__vwprintf_chk()` shall function in the same way as the interface `__vwprintf_chk()`, except that `__nldbl__vwprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__vwprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__vwprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# `__nldbl__wprintf_chk`

## Name

`__nldbl__wprintf_chk` -- convert formatted wide-character output, with stack checking (64-bit version)

## Synopsis

```
int __nldbl__wprintf_chk(int flag, const wchar_t * format);
```

## Description

The interface `__nldbl__wprintf_chk()` shall function in the same way as the interface `__wprintf_chk()`, except that `__nldbl__wprintf_chk()` shall accept a 64-bit long double value (also known as double long double) where `__wprintf_chk()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl__wprintf_chk()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-`

double-64 for the gcc compilation system).

## **\_\_nldbl\_asprintf**

### **Name**

`__nldbl_asprintf` -- print to allocated string (64-bit version)

### **Synopsis**

```
int __nldbl_asprintf(char * * str, const char * fmt);
```

### **Description**

The interface `__nldbl_asprintf()` shall function in the same way as the interface `asprintf()`, except that `__nldbl_asprintf()` shall accept a 64-bit long double value (also known as double long double) where `asprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_asprintf()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **\_\_nldbl\_dprintf**

### **Name**

`__nldbl_dprintf` -- print to a file descriptor (64-bit version)

### **Synopsis**

```
int __nldbl_dprintf(int fd, const char * format);
```

### **Description**

The interface `__nldbl_dprintf()` shall function in the same way as the interface `dprintf()`, except that `__nldbl_dprintf()` shall accept a 64-bit long double value (also known as double

long double) where `dprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_dprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_fprintf

## Name

`__nldbl_fprintf` -- convert formatted output (64-bit version)

## Synopsis

```
int __nldbl_fprintf(FILE * stream, const char * format);
```

## Description

The interface `__nldbl_fprintf()` shall function in the same way as the interface `fprintf()`, except that `__nldbl_fprintf()` shall accept a 64-bit long double value (also known as double long double) where `fprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_fprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_fscanf

## Name

`__nldbl_fscanf` -- input format conversion (64-bit version)

## Synopsis

```
int __nldbl_fscanf(FILE * stream, const char * format);
```

## Description

The interface `__nldbl_fscanf()` shall function in the same way as the interface `fscanf()`, except that `__nldbl_fscanf()` shall accept a 64-bit long double value (also known as double long double) where `fscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_fscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_fwprintf

## Name

`__nldbl_fwprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_fwprintf(FILE * stream, const wchar_t * format);
```

## Description

The interface `__nldbl_fwprintf()` shall function in the same way as the interface `fwprintf()`, except that `__nldbl_fwprintf()` shall accept a 64-bit long double value (also known as double long double) where `fwprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_fwprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_fwscanf

## Name

`__nldbl_fwscanf` -- convert formatted wide-character input (64-bit version)

## Synopsis

```
int __nldbl_fwscanf(FILE * stream, const wchar_t * format);
```

## Description

The interface `__nldbl_fwscanf()` shall function in the same way as the interface `fwscanf()`, except that `__nldbl_fwscanf()` shall accept a 64-bit long double value (also known as double long double) where `fwscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_fwscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_printf

## Name

`__nldbl_printf` -- format and print data (64-bit version)

## Synopsis



```
int __nldbl_printf(const char * format);
```

## Description

The interface `__nldbl_printf()` shall function in the same way as the interface `printf()`, except that `__nldbl_printf()` shall accept a 64-bit long double value (also known as double long double) where `printf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_printf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_scanf

## Name

`__nldbl_scanf` -- input format conversion (64-bit version)

## Synopsis

```
int __nldbl_scanf(const char * format);
```

## Description

The interface `__nldbl_scanf()` shall function in the same way as the interface `scanf()`, except that `__nldbl_scanf()` shall accept a 64-bit long double value (also known as double long double) where `scanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_scanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# **\_\_nldbl\_snprintf**

## **Name**

`__nldbl_snprintf` -- convert formatted output (64-bit version)

## **Synopsis**

```
int __nldbl_snprintf(char * str, size_t size, const char * format);
```

## **Description**

The interface `__nldbl_snprintf()` shall function in the same way as the interface `snprintf()`, except that `__nldbl_snprintf()` shall accept a 64-bit long double value (also known as double long double) where `snprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_snprintf()` function is not in the source standard; it is only in the binary standard.

## **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# **\_\_nldbl\_sprintf**

## **Name**

`__nldbl_sprintf` -- convert formatted output (64-bit version)

## **Synopsis**

```
int __nldbl_sprintf(char * str, const char * format);
```

## **Description**

The interface `__nldbl_sprintf()` shall function in the same way as the interface `sprintf()`, except that `__nldbl_sprintf()` shall accept a 64-bit long double value (also known as double long double) where `sprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_sprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_sscanf

## Name

`__nldbl_sscanf` -- input format conversion (64-bit version)

## Synopsis

```
int __nldbl_sscanf(const char * str, const char * format);
```

## Description

The interface `__nldbl_sscanf()` shall function in the same way as the interface `sscanf()`, except that `__nldbl_sscanf()` shall accept a 64-bit long double value (also known as double long double) where `sscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_sscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_strfmon

## Name

`__nldbl_strfmon` -- convert monetary value to a string (64-bit version)

## Synopsis

```
ssize_t __nldbl_strfmon(char * s, size_t max, const char * format);
```

## Description

The interface `__nldbl_strfmon()` shall function in the same way as the interface `strfmon()`, except that `__nldbl_strfmon()` shall accept a 64-bit long double value (also known as double long double) where `strfmon()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_strfmon()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_swprintf

## Name

`__nldbl_swprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_swprintf(wchar_t * wcs, size_t maxlen, const wchar_t * format);
```

## Description

The interface `__nldbl_swprintf()` shall function in the same way as the interface `swprintf()`, except that `__nldbl_swprintf()` shall accept a 64-bit long double value (also known as double long double) where `swprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_swprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long

double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **\_\_nldbl\_swscanf**

### **Name**

`__nldbl_swscanf` -- convert formatted wide-character input (64-bit version)

### **Synopsis**

```
int __nldbl_swscanf(const wchar_t * ws, const wchar_t * format);
```

### **Description**

The interface `__nldbl_swscanf()` shall function in the same way as the interface `swscanf()`, except that `__nldbl_swscanf()` shall accept a 64-bit long double value (also known as double long double) where `swscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_swscanf()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **\_\_nldbl\_syslog**

### **Name**

`__nldbl_syslog` -- send messages to the system logger (64-bit version)

### **Synopsis**

```
void __nldbl_syslog(int priority, const char * format);
```

## Description

The interface `__nldbl_syslog()` shall function in the same way as the interface `syslog()`, except that `__nldbl_syslog()` shall accept a 64-bit long double value (also known as double long double) where `syslog()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_syslog()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vasprintf

## Name

`__nldbl_vasprintf` -- print to allocated string (64-bit version)

## Synopsis

```
int __nldbl_vasprintf(char * * str, const char * fmt, va_list ap);
```

## Description

The interface `__nldbl_vasprintf()` shall function in the same way as the interface `vasprintf()`, except that `__nldbl_vasprintf()` shall accept a 64-bit long double value (also known as double long double) where `vasprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vasprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# **\_\_nldbl\_vdprintf**

## **Name**

`__nldbl_vdprintf` -- print to a file descriptor (64-bit version)

## **Synopsis**

```
int __nldbl_vdprintf(int fd, const char * format, va_list ap);
```

## **Description**

The interface `__nldbl_vdprintf()` shall function in the same way as the interface `vdprintf()`, except that `__nldbl_vdprintf()` shall accept a 64-bit long double value (also known as double long double) where `vdprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vdprintf()` function is not in the source standard; it is only in the binary standard.

## **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# **\_\_nldbl\_vfprintf**

## **Name**

`__nldbl_vfprintf` -- convert formatted output (64-bit version)

## **Synopsis**

```
int __nldbl_vfprintf(FILE * stream, const char * format, va_list ap);
```

## **Description**

The interface `__nldbl_vfprintf()` shall function in the same way as the interface `vfprintf()`, except that `__nldbl_vfprintf()` shall accept a 64-bit long double value (also known as double long double) where `vfprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vfprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vfscanf

## Name

`__nldbl_vfscanf` -- input format conversion (64-bit version)

## Synopsis

```
int __nldbl_vfscanf(FILE * stream, const char * format, va_list ap);
```

## Description

The interface `__nldbl_vfscanf()` shall function in the same way as the interface `vfscanf()`, except that `__nldbl_vfscanf()` shall accept a 64-bit long double value (also known as double long double) where `vfscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vfscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vfwprintf

## Name



`__nldbl_vfwprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_vfwprintf(FILE * stream, const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vfwprintf()` shall function in the same way as the interface `vfwprintf()`, except that `__nldbl_vfwprintf()` shall accept a 64-bit long double value (also known as double long double) where `vfwprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vfwprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vfwscanf

## Name

`__nldbl_vfwscanf` -- wide-character formatted input of a stdarg argument list (64-bit version)

## Synopsis

```
int __nldbl_vfwscanf(FILE * stream, const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vfwscanf()` shall function in the same way as the interface `vfwscanf()`, except that `__nldbl_vfwscanf()` shall accept a 64-bit long double value (also known as double long double) where `vfwscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vfwscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vprintf

## Name

`__nldbl_vprintf` -- convert formatted output (64-bit version)

## Synopsis

```
int __nldbl_vprintf(const char * format, va_list ap);
```

## Description

The interface `__nldbl_vprintf()` shall function in the same way as the interface `vprintf()`, except that `__nldbl_vprintf()` shall accept a 64-bit long double value (also known as double long double) where `vprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vscanf

## Name

`__nldbl_vscanf` -- input format conversion (64-bit version)

## Synopsis

```
int __nldbl_vscanf(const char * format, va_list ap);
```

## Description

The interface `__nldbl_vscanf()` shall function in the same way as the interface `vscanf()`, except that `__nldbl_vscanf()` shall accept a 64-bit long double value (also known as double long double) where `vscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vsnprintf

## Name

`__nldbl_vsnprintf` -- convert formatted output (64-bit version)

## Synopsis

```
int __nldbl_vsnprintf(char * str, size_t size, const char * format, va_list ap);
```

## Description

The interface `__nldbl_vsnprintf()` shall function in the same way as the interface `vsnprintf()`, except that `__nldbl_vsnprintf()` shall accept a 64-bit long double value (also known as double long double) where `vsnprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vsnprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-`

double-64 for the gcc compilation system).

## **\_\_nldbl\_vsprintf**

### **Name**

`__nldbl_vsprintf` -- convert formatted output (64-bit version)

### **Synopsis**

```
int __nldbl_vsprintf(char * str, const char * format, va_list ap);
```

### **Description**

The interface `__nldbl_vsprintf()` shall function in the same way as the interface `vsprintf()`, except that `__nldbl_vsprintf()` shall accept a 64-bit long double value (also known as double long double) where `vsprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vsprintf()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **\_\_nldbl\_vsscanf**

### **Name**

`__nldbl_vsscanf` -- input format conversion (64-bit version)

### **Synopsis**

```
int __nldbl_vsscanf(const char * str, const char * format, va_list ap);
```

### **Description**

The interface `__nldbl_vsscanf()` shall function in the same way as the interface `vsscanf()`, except that `__nldbl_vsscanf()` shall accept a 64-bit long double value (also known as double

long double) where `vsscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vsscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vswprintf

## Name

`__nldbl_vswprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_vswprintf(wchar_t * wcs, size_t maxlen, const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vswprintf()` shall function in the same way as the interface `vswprintf()`, except that `__nldbl_vswprintf()` shall accept a 64-bit long double value (also known as double long double) where `vswprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vswprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vswscanf

## Name

`__nldbl_vswscanf` -- wide-character formatted input of a stdarg argument list (64-bit version)

## Synopsis

```
int __nldbl_vswscanf(const wchar_t * ws, const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vswscanf()` shall function in the same way as the interface `vswscanf()`, except that `__nldbl_vswscanf()` shall accept a 64-bit long double value (also known as double long double) where `vswscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vswscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vsyslog

## Name

`__nldbl_vsyslog` -- send messages to the system logger (64-bit version)

## Synopsis

```
void __nldbl_vsyslog(int priority, const char * format, va_list ap);
```

## Description

The interface `__nldbl_vsyslog()` shall function in the same way as the interface `vsyslog()`, except that `__nldbl_vsyslog()` shall accept a 64-bit long double value (also known as double long double) where `vsyslog()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vsyslog()` function is not in the source standard; it is only in the binary

standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vwprintf

## Name

`__nldbl_vwprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_vwprintf(const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vwprintf()` shall function in the same way as the interface `vwprintf()`, except that `__nldbl_vwprintf()` shall accept a 64-bit long double value (also known as double long double) where `vwprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vwprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

# \_\_nldbl\_vwscanf

## Name

`__nldbl_vwscanf` -- wide-character formatted input of a stdarg argument list (64-bit version)

## Synopsis

```
int __nldbl_vwscanf(const wchar_t * format, va_list args);
```

## Description

The interface `__nldbl_vwscanf()` shall function in the same way as the interface `vwscanf()`, except that `__nldbl_vwscanf()` shall accept a 64-bit long double value (also known as double long double) where `vwscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_vwscanf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-long-double-64` for the gcc compilation system).

# \_\_nldbl\_wprintf

## Name

`__nldbl_wprintf` -- convert formatted wide-character output (64-bit version)

## Synopsis

```
int __nldbl_wprintf(const wchar_t * format);
```

## Description

The interface `__nldbl_wprintf()` shall function in the same way as the interface `wprintf()`, except that `__nldbl_wprintf()` shall accept a 64-bit long double value (also known as double long double) where `wprintf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_wprintf()` function is not in the source standard; it is only in the binary standard.

## Application Usage (informative)

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long



double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

## **\_\_nldbl\_wscanf**

### **Name**

`__nldbl_wscanf` -- convert formatted wide-character input (64-bit version)

### **Synopsis**

```
int __nldbl_wscanf(const wchar_t * format);
```

### **Description**

The interface `__nldbl_wscanf()` shall function in the same way as the interface `wscanf()`, except that `__nldbl_wscanf()` shall accept a 64-bit long double value (also known as double long double) where `wscanf()` accepts a 128-bit long double value (also known as quad long double).

The `__nldbl_wscanf()` function is not in the source standard; it is only in the binary standard.

### **Application Usage (informative)**

Beginning with version 4.0 of the Linux Standard Base Core Specification for PPC32, long double values on all supported architectures are 128-bit values. In previous versions on this architecture, they were 64-bit values. This interface is a transitional aid used to provide the old ABI for programs that require it. It may also be possible to generate programs that match the older ABI by selecting appropriate compilation options (for example, `-mlong-double-64` for the gcc compilation system).

---

## **11.5. Interfaces for libm**

[Table 11-36](#) defines the library name and shared object name for the libm library

**Table 11-36. libm Definition**

Library:	libm
SONAME:	libm.so.6

The behavior of the interfaces in this library is specified by the following specifications: