AUTHORIZED DOCUMENTATION

XPlat Linux Libraries for C

Novell® Developer Kit

December 2008

www.novell.com

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell developer products, and to get updates, see the Novell Developer Community Web page (http://developer.novell.com/wiki/index.php/Developer_Home).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

XPlat Linux Libraries for C (XPlat) contains the linker and header files that you need for Novell® related application development on Linux*. Applications using XPlat don't need the Novell Client™ for Linux. However, certain operations (such as mapping network drives) won't work unless you install the Novell Client for Linux.

This guide contains the following sections:

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of this guide, see XPlat Linux Libraries for C (http://developer.novell.com/wiki/index.php/Xplat-linux).

Additional Information

For more Linux functions, see NLM™ and NetWare® Libraries for C (including CLIB and XPlat) (http://developer.novell.com/wiki/index.php/NLM_and_NetWare_Libraries_for_C_(including_CLIB_and_XPlat)).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

Overview

1

XPlat Linux Libraries for C (XPlat) contains all of the required linker and header files that you need to develop Novell® related applications on Linux.

You don't need to install the Novell Client™ for Linux. However, some operations (such as mapping network drives) won't work unless the Novell Client for Linux is installed.

If you already have the Novell Client for Linux or Open Enterprise Server 2 (OES 2) installed, you only need to install the devel and header packages from XPlat for application development.

1.1 Installing

What you need to install depends on what you already have installed:

- Section 1.1.1, "Linux Not Installed," on page 9
- Section 1.1.2, "Linux Installed," on page 9

1.1.1 Linux Not Installed

If OES 2 or the Novell Client for Linux aren't already installed, install the following packages:

Table 1-1 *Install Packages without Linux Installed*

Package	Description
`novell-xplatlib-devel`	Linker files
`novell-xplatlib-headers`	SDK header files
`novell-xplatlib`	XPlat library files
`novell-xtier-xplat`	XTier requester module, `xtxplat.so`
`novell-xtier-base`	XTier base module
`novell-xtier-core`	XTier main module
`novell-nmasclient`	NMAS™ authentication modules
`nici` or `nici64`	NICI Crypto services

1.1.2 Linux Installed

If OES 2 or the Novell Client for Linux are already installed, you need to install only the following two packages:

Table 1-2 Install Packages with Linux Installed

Package	Description
novell-xplatlib-devel	Linker files
novell-xplatlib-headers	SDK header files

1.2 Configuring

If the Novell Linux Client isn't installed, users other than the `root` user need to be added to the novlxtier group for XPlat to work. If the client is not installed, XPlat uses XTier's requester interface module. Otherwise, XPlat requests are routed through the client's requester interface in novfs to novfsd, which runs as `root` and interfaces with XTier.

Public connections that are created through the client have user scope, which means that all processes for a given user share the same connection to a server. Without the client, connections have process scope and multiple private connections from the same process can be created.

Currently, we recommend using XPlat without the client. If you need to use the client, you can temporarily disable it by entering `rcnovfsd stop` at a console, which allows you to test your application without the client running.

1.3 Compiling and Building

By default, the XPlat header files are downloaded to `/opt/novell/include/xplat`, and the XPlat libraries are downloaded to `/opt/novell/lib` and are named as follows:

- clnlnx
- ncplnx
- callnx
- loclnx
- netlnx

The obsolete NLM™ and NetWare® Libraries for C (including CLIB and XPlat) (http://developer.novell.com/wiki/index.php/NLM_and_NetWare_Libraries_for_C_(including_CLIB_and_XPlat)) functions are not included in the XPlat Linux Libraries for C, and the obsolete header files have been removed.

XPlat uses Unicode* UCS-2, not UCS-4, so the GCC `-fshort-wchar` compile flag must be used for Unicode strings.

1.4 Troubleshooting

To troubleshoot XTier, make sure `novell-xregd` is running (`rcnovell-xregd` start or stop) and that `/var/opt/novell/xtier/xregd` and `/var/opt/novell/xtier` are owned by `novlxregd` with the novlxtier group.

Also, see the note about other users needing to be added to the novlxtier group in Section 1.2, "Configuring," on page 10.

1.5 Limitations

There are a few limitations in using the NLM and NetWare Libraries for C (including CLIB and XPlat) (http://developer.novell.com/wiki/index.php/NLM_and_NetWare_Libraries_for_C_(including_CLIB_and_XPlat)) functions on Linux:

- Section 1.5.1, "System Code Pages," on page 11
- Section 1.5.2, "UTF-8 Names," on page 11

1.5.1 System Code Pages

On Linux, the Unicode NWUS and NWUX conversion functions (http://developer.novell.com/documentation/clib/ucod_enu/data/h7qwv271.html) support only the 0 system code page. (Usually, the system code page is UTF-8 on Linux.)

The system code page is determined by the current locale character type, LC_CTYPE, which is set by the NWLsetlocale function (http://developer.novell.com/ndk/doc/clib/intl_enu/data/sdk633.html).

NWLsetlocale calls the C library setlocale function to do most of its work. However, NWLsetlocale on Linux does not accept locale strings such as 437 or 932. Instead, they must be translated to en_US.IBM437 and ja_JP.SHIFT-JIS.

NWLsetlocale was based on older UNIX* functions that did not specify language and territory. In the future, we might rework the NWUX functions to use the iconv function so that you can specify other server code pages. Also, we might change it to be more consistent with the way the binaries work on other platforms.

1.5.2 UTF-8 Names

If a volume (such as a legacy volume) does not support UTF-8, the extended file and path functions that used UTF-8 names in the past now use the non-UTF-8 functions that return names in the server's code page. When this switch happens, all path and file names are converted to UTF-8, using the NWUX functions and the system code page.

Currently, the only way to control this behavior is to call NWLsetlocale (http://developer.novell.com/ndk/doc/clib/intl_enu/data/sdk633.html). In the future, we might add a function that sets a default OEM code page for the extended functions or allows them to return an error if UTF-8 is not supported.

1.6 Differences

The differences between the XPlat and NLM and NetWare Libraries for C (including CLIB and XPlat) (http://developer.novell.com/wiki/index.php/NLM_and_NetWare_Libraries_for_C_(including_CLIB_and_XPlat)) functions include the following:

- Section 1.6.1, "Iteration Handle Types," on page 12
- Section 1.6.2, "64-Bit Platforms," on page 12
- Section 1.6.3, "VERSION_INFO Structure," on page 12

1.6.1 Iteration Handle Types

NWDSSearch (http://developer.novell.com/documentation/ndslib/nds__enu/data/sdk2591.html) and other NDS® iteration functions use a new nint_ptr type, which is an integer the size of a pointer, rather than the nint32 type.

For 32-bit platforms, the nint_ptr and nint32 types are identical. However, for 64-bit platforms, the two types are not the same. You must use the nint_ptr type.

1.6.2 64-Bit Platforms

XPlat supports 64-bit platforms for Windows* Vista* and Linux (x86_64). However, it does not support DOS, OS/2*, or 16-bit Windows platforms. Only the 32-bit and 64-bit Windows platforms are supported.

The `ntypes.h` file was changed to detect the machine architecture (N_ARCH_32 or N_ARCH_64) and the OS platform when possible. XPlat detects the Windows (N_PLAT_MSW) and UNIX (N_PLAT_UNIX) platforms but doesn't detect NetWare. If you use NetWare, you still need to define N_PLAT_NLM.

1.6.3 VERSION_INFO Structure

The VERSION_INFO (http://developer.novell.com/ndk/doc/clib/srvr_enu/data/sdk864.html#sdk864) structure is updated to include the latest information that is returned by NCP™ 23 17 Get File Server Information. Some of the newest fields include

- 64BitOffsetsSupportedFlag
- NCPServerPlatform
- ServerCodePageString

The server code page string is returned by NetWare on Linux servers and by native servers since 2006.

1.7 Code Pages

The following table shows the compatibility between NetWare and Linux code pages:

Table 1-3 *Code Pages*

NetWare	ICONV	Locale
437	United States English	IBM437
850	Multilingual	IBM850
852	Slavic	IBM852
855	Cyrillic	IBM855
857	Turkish	IBM857
860	Portugese	IBM860

NetWare	ICONV	Locale
861	Icelandic	IBM861
862	Hebrew	IBM862
863	Canadian French	IBM863
865	Nordic	IBM865
866	Russian	IBM866
874	Thai	IBM874
932	Japanese	SHIFT_JIS
936	Simplified Chinese	GBK
949	Korean	IBM949
950	Traditional Chinese	BIG5

Functions

2

These functions are new and are not included in NLM™ and NetWare® Libraries for C (including CLIB and XPlat) (http://developer.novell.com/wiki/index.php/NLM_and_NetWare_Libraries_for_C_(including_CLIB_and_XPlat)).

The following file access functions are available to access NetWare files on Linux:

NWCommitFile

Flushes a written file from a cache to disk.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWCommitFile(
   NWFILE_HANDLE   fileHandle);
```

Parameters

fileHandle

(IN) Specifies the file to commit.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWGetFilePos

Returns the position of a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWGetFilePos(
   NWFILE_HANDLE   fileHandle,
   pnuint64        filePos);
```

Parameters

fileHandle

(IN) Specifies the file to return the position for.

filePos

(OUT) Points to the position of the file.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWGetFileSize

Returns the size of a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWGetFileSize(
   NWFILE_HANDLE   fileHandle,
   pnuint64        fileSize);
```

Parameters

fileHandle

(IN) Specifies the file to return a size for.

fileSize

(OUT) Points to the size of the file.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWReadFile

Reads a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWReadFile(
   NWFILE_HANDLE   fileHandle,
   nuint32         bytesToRead,
   pnuint32        bytesRead,
   pnuint8         data);
```

Parameters

fileHandle

(IN) Specifies the file to read.

bytesToRead

(IN) Specifies the number of bytes to read.

bytesRead

(OUT) Specifies the number of bytes that were read.

data

(OUT) Points to the data that was read.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWSetFilePos

Sets the position of a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWSetFilePos(
   NWFILE_HANDLE   fileHandle,
   nuint           mode,
   nint64          filePos);
```

Parameters

fileHandle

(IN) Specifies the file to set the position for.

mode

(IN) Specifies the position mode of the file:

0 SEEK_FROM_BEGINNING
1 SEEK_FROM_CURRENT
2 SEEK_FROM_END

filePos

(IN) Specifies the position of the file.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWSetFileSize

Sets the size of a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWSetFileSize(
   NWFILE_HANDLE   fileHandle,
   nuint64         fileSize);
```

Parameters

fileHandle

(IN) Specifies the file to set the size for.

fileSize

(IN) Specifies the size of the file.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

NWWriteFile

Writes a file.

Syntax

```
#include <nwfile.h>

N_EXTERN_LIBRARY (NWCCODE) NWWriteFile(
   NWFILE_HANDLE   fileHandle,
   nuint32         bytesToWrite,
   pnuint8         data);
```

Parameters

fileHandle

(IN) Specifies the file to write to.

bytesToWrite

(IN) Specifies the number of bytes to write.

data

(OUT) Points to the data that was written.

Return Values

For more information, see the Return Values for C documentation (http://developer.novell.com/documentation/general/genl_enu/data/bktitle.html).

Thread Session Functions

3

The Thread Session Context functions are available on the Linux platform. Using these functions you can migrate Netware applications that use XPlat libraries to Linux platform.

An application can manage user identities and connections using these functions.

Using these functions an application can create new XTier session contexts and assigns them to various threads. Connections and user identities are associated with a session context. Using a uinque session context, a thread can create its own private connections and identities. You can also share session contexts with threads. A thread uses the default session context if it is not assigned to a session context handle.

XPlat uses NetWare Requester API to perform the basic functions like creating connections, authenticate users etc. By default, XPlat uses Novell Client's Requester interface. If Novell Client is not installed, XPlat uses XTier's requester interface module.The client shares the connections and identities between the user processes. Even if Novell Client is installed, you can force an application to use XTier by setting an environment variable to "XPLAT_USE_XTIER".

NOTE: For the thread session functions to work, XTier's requester interface must be available.

The following thread session context functions are available only on Linux:

If XTier interface is not availble, the functions returns NWE_REQUESTER_FAILURE. Also the XPlat initialization functions like NWCallsInit(), NWCLXInit, NWNetInit fails to succeed if the XTier interface is not available.

NWCreateThreadSessionContext

Creates a new session context.

Syntax

```
#include <nwmisc.h>
N_EXTERN_LIBRARY(NWCODE)NWCreateThreadSessionContext (TSCHANDLE *phTSC);
```

Parameters

phTSC

(OUT) Points to a new session context handle.

Return Values

0x0000 SUCCESS
0x881A NWE_OUT_OF_HEAP_SPACE
0x8836 NWE_PARAM_INVALID
0x88FF NWE_REQUESTER_FAILURE

NWDestroyThreadSessionContext

Destroys a created session context.

Syntax

```
# include <nwmisc.h>
N_EXTERN_LIBRARY(NWCCODE) NWDestroyThreadSessionContext  (TSCHANDLE hTSC);
```

Parameters

hTSC

(IN) Specifies the session context handle.

Return Values

0x0000 SUCCESS
0x8836 NWE_PARAM_INVALID
0x88FF NWE_REQUESTER_FAILURE

Remarks

Destroying a session context handle automatically removes the handle from any threads that are assigned to it, causing them to revert to the global default session context.

NWSetThreadSessionContext

Assigns a session context to a thread.

Syntax

```
# include <nwmisc.h>
N_EXTERN_LIBRARY(NWCCODE) NWSetThreadSessionContext  (TSCHANDLE hTSC);
```

Parameters

hTSC

(IN) Specifies the session context handle.

Return Values

0x0000 SUCCESS
0x8800 NWE_ALREADY_ATTACHED
0x8836 NWE_PARAM_INVALID
0x88FF NWE_REQUESTER_FAILURE

Remarks

The return value NWE_ALREADY_ATTACHED indicates that a thread is already assigned to a session context which needs to be cleared from this thread for this function to succeed.

NWGetThreadSessionContext

Returns a thread's assigned session context.

Syntax

```
# include <nwmisc.h>
N_EXTERN_LIBRARY(NWCCODE) NWGetThreadSessionContext  (TSCHANDLE *phTSC);
```

Parameters

phTSC

(OUT) Points to a session context handle.

Return Values

0x0000 SUCCESS
0x8836 NWE_PARAM_INVALID
0x886F NWE_OBJECT_NOT_FOUND
0x88FF NWE_REQUESTER_FAILURE

Remarks

If there is no session context assigned to the thread, NWE_OBJECT_NOT_FOUND is returned and *phTSC is set to NULL.

NWGetThreadSessionContextCount

Returns the number of threads assigned to the session context.

Syntax

```
# include <nwmisc.h>
N_EXTERN_LIBRARY(NWCCODE) NWGetThreadSessionContextCount  (TSCHANDLE hTSC,
pnuint32 count);
```

Parameters

hTSC

(IN) Specifies the session context handle.

count

(OUT) Reference count

Return Values

0x0000 SUCCESS
0x8836 NWE_PARAM_INVALID
0x88FF NWE_REQUESTER_FAILURE

Remarks

Helper function.

NWClearThreadSessionContext

Removes a thread’s assigned session context.

Syntax

```
# include <nwmisc.h>
N_EXTERN_LIBRARY(NWCCODE) NWClearThreadSessionContext  (void);
```

Return Values

0x0000 SUCCESS
0x886F NWE_OBJECT_NOT_FOUND
0x88FF NWE_REQUESTER_FAILURE

Remarks

When a thread's session context is removed, the thread reverts to the global default session context. If a thread is not assigned any session context, NWE_OBJECT_NOT_FOUND is returned.

Documentation Revision History

A

The following sections outline the changes that have been made to the XPlat Linux Libraries for C documentation (in reverse chronological order):

December 2008	Added Chapter 3, “Thread Session Functions,” on page 23.
October 17, 2007	Added to the NDK.