

YNQ Client Evaluation Package for Linux

YNQ 1.6.0

Document version 0.27

Table of Contents

1	REFERENCED DOCUMENTS	3
2	INTRODUCTION.....	4
3	REQUIREMENTS.....	5
4	PACKAGE CONTENTS.....	6
5	INSTALLATION	7
6	BUILDING BINARIES	8
6.1	SAMPLE APPLICATION.....	8
6.2	NQ	9
6.3	KERBEROS SUPPORT IN NQ CLIENT	10
6.4	NQDRV FILESYSTEM DRIVER	11
6.5	LDAP SUPPORT IN NQ CLIENT (CORPORATE ONLY).....	12
6.6	WS-DISCOVERY CLIENT SIDE SUPPORT	13
7	CLEANING THE BINARIES	14
8	CONFIGURATION.....	15
9	RUNNING NQ.....	16
10	THE SOURCE CODE	17
10.1	NQ CLIENT SAMPLE APPLICATION SOURCES	17
10.2	NQ SOURCES	18

1 Referenced Documents

- [1] NQ Client Integration and Porting Guide
- [2] NQ User's Guide

2 Introduction

This document describes the Linux version of YNQ Client. It is complimentary to the documents [1] and [2] and describes Linux-specific solutions.

3 Requirements

NQ Evaluation Package for Linux requires the following environment:

- PC running Linux
- User should have root privileges to run the daemons
- Kernel must support the FUSE – File System User Space (required by NQ FS driver)

4 Package Contents

The NQ package contains the following files:

nqRoot/

config/	The config files for NQ Client
cm_cfg.txt	Common configuration file
cc_cfg.txt	CIFS Client configuration file
NQ Docs/	directory containing the NQ documents
NQ Library Reference/	NQ API files
src/	
client/	directory contains the NQ Client code
netbios/	directory contains the NQ NetBIOS (name resolution) code
nq/	directory containing NQ API *.h files
ldap/	directory contains the NQ LDAP API (corporate only)
service/	directory contains the NQ level II and level III modules
auth/	directory contains the NQ authentication code
common/	directory contains the NQ common operations code
network/	directory contains the NQ network operation code
/wsd	directory contains the NQ WSD operation code
os/	directory contains the NQ level III modules
linux/	directory that refer the Linux OS
driver/	directory contains the NQ driver code
sy/	directory contains the NQ system dependent code
ud/	directory contains the NQ user defined code
app/	directory containing sample application
appmain.c	sample application startup code
ccexdirtraverse.c	sample application for traversing directories
ccexfileanddir.c	sample application for file and directory operations
ccexfileop.c	sample application of file basic operations
ccexjoindomain.c	sample application for demonstrating join domain capability (corporate)
ccexgeneral.c	general functions
ccexgeneralfunc.c	general functions
ccexgeneralfunc.h	general functions
ccexldap.c	sample application for demonstrating basic LDAP operations (corporate)
ccexnetworkenum.c	sample application of network enumerating
ccexserversidecopy.c	sample application for demonstrating server-side-copy capability
Makefile	sample application makefile (executed from the main makefile)
Makefile	main makefile
osdef.mk	OS and compiler dependent makefile configurations file
common.mk	common makefile operations file
pathdef.mk	makefile source code paths configurations

5 Installation

NQ Evaluation Package installation is as simple as extracting the package archive into any directory on a Linux PC. Please see next section for building instructions.

6 Building Binaries

6.1 *Sample application*

After modifying the sample application sources, the **nqapp** executable can be rebuilt by running **make** in *nqRoot/src*.

This procedure recreates the *nqRoot/nqapp* executable. In some cases, the execute permissions can be cleared for this file in which case either run **chmod a+x nqapp** in the *nqRoot* directory.

6.2 NQ

To build the NetBIOS daemon run **make nqnd**. To update the execute permissions for the executables run **make install** in *nqRoot/src*.

6.3 **KERBEROS support in NQ Client**

By default, KERBEROS support is not included in the binaries. In order to enable it perform the following:

1. Uncomment the line “`#define UD_CC_INCLUDEEXTENDEDSECURITY_KERBEROS`” in the file *nqRoot/src/service/os/linux/ud/udparams.h*
2. Make sure `SYSLIBS` contains `-lsasl2 -lkrb5` in the file: *nqRoot/src/osdef.mk*

Depending on the KERBEROS implementation on your Linux box, you may have to adjust include and library paths as well. To do this modify the variables `INCPATH` and `LIBPATH` in the makefiles mentioned above.

Default implementation supplied uses HEIMDAL, in order to use MIT implementation modify *nqRoot/src/service/os/linux/sysasl.c*, comment “`#define HEIMDAL`”, uncomment “`#define MIT`”. When switching back and force between MIT and HEIMDAL make sure to reissue Kerberos ticket. In some case uninstalling the whole package is required.

Note: the sources have to be recompiled after the described modifications.

6.4 NQDRV filesystem driver

As NQDRV works in the FUSE environment (see <http://fuse.sourceforge.net/>) it requires its definitions to match those of FUSE.

1. Uncomment the line “#define UD_CC_INCLUDEFSDRIVER” in the file *nqRoot/src/service/os/linux/ud/udparams.h*
2. Make sure `SYSLIBS` contains `-lfuse` in the file: *nqRoot/src/osdef.mk*
3. Compile **nqdrv** executable use command **make -C src nqdriverall**

Note: The macro `_FILE_OFFSET_BITS` defines the expected file offset size. This macro is defined as 64 in *nqRoot/src/service/os/linux/driver/fsdriver.c*. Such definition matches most (if not all) of known FUSE setups. However, it may be a good idea to consult FUSE source for this value.

6.5 LDAP support in NQ Client (corporate only)

NQ Client Corporate supports LDAP operations based on OpenLDAP library.

By default, LDAP support is not included in the binaries. In order to enable it please do the following:

1. Uncomment the line “#define UD_CC_INCLUDELDAP” in the file *nqRoot/src/service/os/linux/ud/udparams.h*
2. Make sure `SYSLIBS` contains `-lldap -llber` in the file: *nqRoot/src/osdef.mk*
3. OpenLDAP requires Kerberos/SASL support (see section 6.3)

The minimal version supported for openldap package is 2.4.44 (see <http://www.openldap.org/>).

The minimal version required for cyrus-sasl package is 2.1.26 (see <http://asg.web.cmu.edu/sasl/sasl-library.html>)

Note: the sources have to be recompiled after the described modification.

6.6 WS-Discovery Client side support

NQ Client supports devices discovery using WS-Discovery protocol.

By default, Linux package includes enabled “#define UD_CM_INCLUDEWSDCLIENT” in the file *nqRoot/src/service/os/linux/ud/udparams.h*.

By default in *nqRoot/src/osdef.mk* `SYSLIBS` contains `-lxml2`.

Libxml2 package is required (see <http://xmlsoft.org/>), version tested 2.9.1.

7 Cleaning the Binaries

To remove the sample application object files and the **nqapp** executable run **make clean** to clean the **dep/ lib/** and **obj/** in *nqRoot/src*.

To remove completely all NQ and sample application objects and binaries run **make cleanall** in *nqRoot/src*. The security descriptors management library binary and its sample application are also cleaned by this process.

8 Configuration

NQ Evaluation Package contains default configuration which should allow it running on any standard Linux machine. We advise to observe a configuration file `nqRoot/config/cm_cfg.txt`, usually parameters related to domain name, user credentials and DNS IP are modified to reflect the environment. Additional information about NQ configuration files can be found in the headers of these files in `nqRoot/config` folder.

9 Running NQ

Built executables in previous steps should be run in the following sequence to provide the complete SMB/CIFS Client functionality:

1. *nqRoot/nqnd* NetBIOS Daemon
2. *nqRoot/nqapp* Sample application
3. *nqRoot/nqdrv* NQ Filesystem Driver

Notes:

- All NQ daemons should be post fixed with **&** to make them run in the background (for example: **prompt> ./nqnd&**).
- All daemons require **root** privileges to be started.
- For command line syntax run the application without arguments. Default user credentials are retrieved by *udGetCredentials()* (see Integration Guide chapter in NQ Client Integration and Porting Guide document).
- Evaluating NQ Filesystem Driver requires FUSE component to be previously installed. See <http://fuse.sourceforge.net/> for details. For command line syntax run the application without arguments. Default user credentials are retrieved by *udGetCredentials()* (see Integration Guide chapter in NQ Client Integration and Porting Guide document).

10 The Source Code

10.1 NQ client sample application sources

The sample application source code is located under *nqRoot/src/app* directory.

appmain.c file contain application startup code and normally should not be modified.

The sample application code is implemented in **ccexfileanddir.c** and can be freely modified.

File **ccexgeneral.c** demonstrated basic file operations and network enumeration of domains, servers and shares.

The default implementation of the sample application demonstrates the **test.txt** file copy procedure from the local disk to a remote Windows PC. Throughout the procedure user is advised to browse the network to find a PC, then a share on a PC, then the directory to copy file to. This clearly illustrates the scenario when user scans a document on an NQ powered scanner and then saves it on a remote PC.

During the process of browsing to the directory application requests to enter user credentials (Username, Password and Domain) for a user authorized to write into the selected directory on a PC.

Additional files **ccexjoindomain.c** demonstrates join domain capability (corporate).

File **ccexldap.c** demonstrates LDAP functionality (corporate).

Important! Sample application requires several **.h** files from *nqRoot/src/nq* and *nqRoot/src/service/os/linux/** directories.

10.2 NQ sources

NQ sources are subdivided to:

- NQ system independent client core: *nqRoot/src/client*
- NQ system independent NetBIOS core: *nqRoot/src/netbios*
- NQ system independent NQ core: *nqRoot/src/nq*
- NQ system abstraction layer for Linux: *nqRoot/src/service/os/linux*

Normally core sources are not supposed to be modified.