

jNQ™: A Pure Java SMB Client

Table of Contents

1. Executive Summary	3
2. The issue: Lack of an Updated and Effective SMB Solution for Development in Java	4
3. The Solution	5
4. Pure Java Architecture	6
5. Functionality	9
6. Compliance and Connectivity	10
7. Summary	11

Table of Figures

<i>Figure 1: The architecture of jNQ™</i>	6
<i>Figure 2: jNQ™ Connectivity</i>	10

1. Executive Summary

The need to integrate Java with the latest SMB version has grown dramatically following the WannaCry and Petya cyber-attacks that occurred in mid-2017, and by Microsoft disabling SMBv1 by default as of Windows 10 RS3.

The Visuality latest and most up-to-date implementation of Microsoft's SMB, jNQ™ provides Java developers secured file sharing over the encrypted SMB 3.1.1 version.

Written in pure Java, jNQ™ is a client software library, which is available through its API and works in any Java environment starting from 1.5 (Android 4.4.x). With jNQ™, Java developers can enjoy security through message signing, encryption, active directory authentication, Kerberos authentication and pre-logon integrity.

2. The Issue: Lack of an Updated and Effective SMB Solution for Development in Java

The SMB protocol, unlike FTP and HTTP, not only allows copying an entire file, but also grants access to files over the network. For example, file editing can be executed over SMB without changing its location.

The SMB implementation that supports the latest Microsoft SMB specifications in Java has been sorely lacking. The need to integrate Java with the latest SMB version has also grown dramatically following the WannaCry and Petya cyber-attacks. Another shortcoming felt in Java is that SMBv1 cannot communicate with the latest Windows versions, since Microsoft has disabled SMBv1 by default as of Windows 10 RS3 (September 2017) to address security issues and low performance.

The latest SMBv3.x is today the industry requirement and this is also the case for Java. Integration with the latest SMB protocol will offer compatibility with all Windows versions, and will also prevent malicious attacks by means of message signing, encryption, Active Directory authentication, Kerberos authentication and pre-logon integrity.

3. The Solution

Visuality Systems has taken the initiative to develop and provide Java developers with the latest and up-to-date implementation of the Microsoft SMB file sharing connectivity. Named jNQ™ and written in pure Java, jNQ™ is a client software library available through its API and supports all SMB versions up to and including SMB3.1.1.

jNQ™ is the continuation of the NQE™ product, the most well-known and commercially used SMB stack developed by Visuality Systems since 1998.

4. Architecture of jNQ™

jNQ™ is a pure Java library and runs on any Java platform as from version 1.5 (Android 4.4.x) upwards. This approach grants compatibility with numerous platforms: Windows, Mac OS, Linux, UNIX, Android, etc.

The following figure depicts the principal classes of jNQ API.

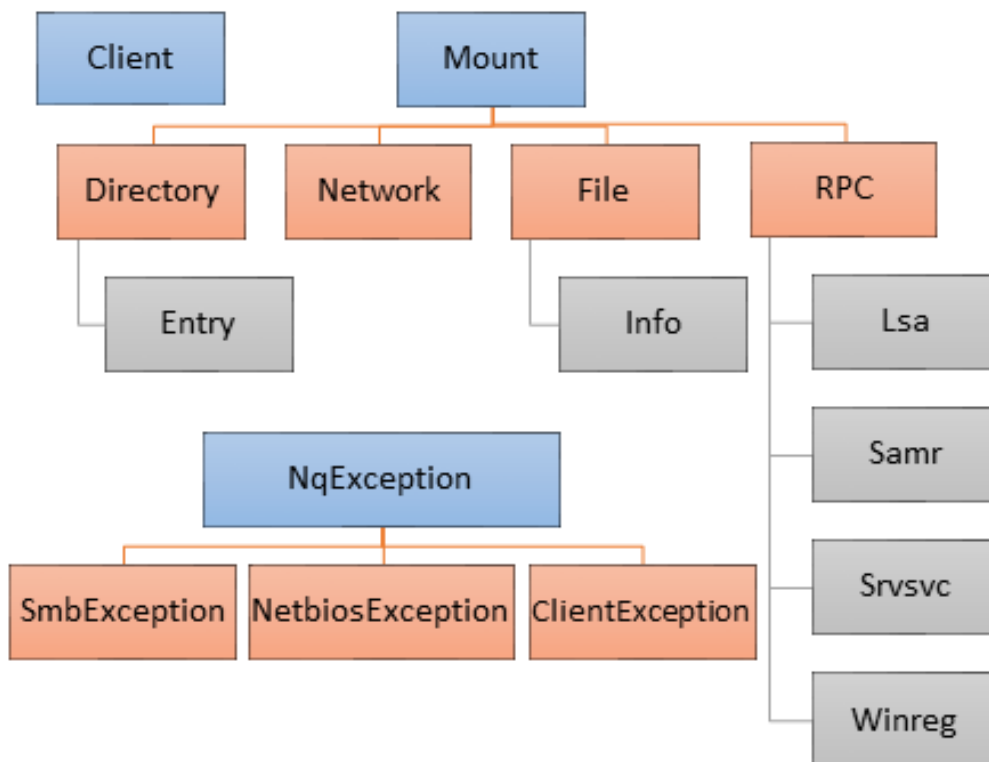


Figure 1: The architecture of jNQ™

Using jNQ™

jNQ™ is available through its API, which comprises the following classes:

Main classes:

Class	Method examples	Aim
Mount		A mount point referring to a remote share
File	read(), write(), getInfo(), setInfo(), delete(), rename()	A file open for reading, writing, or meta-data operations
SmbInputStream		A stream wrapper around file
SmbOutputStream		A stream wrapper around file
Directory		Directory scan
Network (*)	enumerateDomains(), enumerateServers(), enumerateShares()	Network discovery
Client	checkCredentials(), setDialect()	Client management

RPC classes:

Class	Method examples	Aim
Dcerpc (*)		A generic DCERPC framework for developing yet another RPC
Dssetup (*)	roleGetPrimaryDomainInformation()	DSSETUP sub-pipe of LSA pipe
Lsar (*)	openPolicy(), lookupName(), lookupSid(), queryDomainInfo()	LSA pipe
Samr (*)	openDomain(), lookupDomain(), getmemberNamesInAlias(), openAlias()	SAMR pipe
Srvsvc (*)	shareEnum(), shareAdd(), shareDel(), shareGetInfo(), shareSetInfo()	SRVSVC pipe
Winreg (*)	openLocalMachine(), openKey(), enumValue(), setValue()	WINREG pipe

Auxiliary classes:

Class	Methods examples	Aim
NqException		jNQ-specific error description
SmbException		SMB error description
NetbiosException		Name resolution error description
PasswordCredentials		Credentials container: user, password, domain
SubjectCredentials		A container for a Kerberos ticket
Config		jNQ configuration parameters

Notes:

- The tables above are an illustration only and not a complete API reference.
- The functionality marked by (*) is a subject to access rights. For instance, it may not be available without a root access.
- A complete API Javadoc is available on demand.

Sample application

The following code is a sample application on top of jNQ API.

```
import com.visuality.nq.auth.PasswordCredentials;
import com.visuality.nq.client.Mount;
import com.visuality.nq.common.Buffer;
import com.visuality.nq.common.SmbException;
import com.visuality.nq.client.File.Params;

public class Test {
public static void main(String[] args) throws
Exception {
    try {
        PasswordCredentials creds = new
PasswordCredentials("John", "johnspassword",
"johnsdomain");
        Mount mountpoint = new
Mount("someserver", "someshare", creds);

        // create file and write into it
        Params params = new
File.Params(File.ACCESS_WRITE, File.SHARE_FULL,
File.DISPOSITION_OPEN_IF, false);
        File file = new File(mountpoint,
"somefolder/ test.txt", params);
        Buffer writeData = new
Buffer(DATASIZE);
        writeData.dataLen =
writeData.data.length;
        file.write(writeData);
        file.close();

    } catch (NqException e) {
        e.printStackTrace();
    }
}
}
```


5. Functionality

jNQ™ SMB Client features:

- SMB dialect support from NTLM0.12 (SMB1) to SMB 3.1.1
- Rich set of calls:
 - Full set of file data operations
 - Full set of file meta-data calls
 - File tree traversal – directory search
 - Network discovery calls
 - Run-time fine-tuning
 - Asynchronous reads and writes (optional)
 - Host resolution through DNS and NetBIOS
- RPC calls: SAMR, LSA, SRVSVC, WINREG
- Multi-threading
- Various methods of authentication:
 - From LM to NTLMV2, either “naked” or wrapped into SPNEGO
 - Kerberos (not supported in Android)
- Message signing
- SMB encryption

6. Compliance and Connectivity

jNQ™ complies with Microsoft SMB/SMB2/SMB3 specifications. jNQ™ supports all SMB dialects, from NTLM 0.12 to 3.1.1. This grants connectivity to all server versions of Microsoft, Apple Macintosh, NQE™ and Samba.

The table in Figure 2 demonstrates connectivity between the most common Microsoft SMB implementations. In all cases, jNQ™ negotiates the latest SMB dialect offered by the


SERVER \ CLIENT	Windows 10 RS3	Windows 10 WS 2016	Windows 8.1 WS 2012 R2	Windows 8 WS 2012	Windows 7 WS 2008 R2	Windows Vista WS 2008	Previous Versions
	SMB 3.1.1	SMB 3.1.1	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 10 RS3	SMB 3.1.1	SMB 3.1.1	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	X
Windows 10 WS 2016	SMB 3.1.1	SMB 3.1.1	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 8.1 WS 2012 R2	SMB 3.02	SMB 3.02	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 8 WS 2012	SMB 3.0	SMB 3.0	SMB 3.0	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 7 WS 2008 R2	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.0	SMB 1.0
Windows Vista WS 2008	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 1.0
Previous Versions	X	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0

Figure 2: jNQ™ Connectivity

7. Summary

Visuality Systems has developed the jNQ™ product to provide a solution that was anticipated by the Java market for a long time. Finally, Java developers can implement the latest SMB dialect, benefit from a commercial code and have the backing of an SMB specialist providing the latest specifications and periodical upgrades.

With 20 years of experience in the SMB/CIFS market, Visuality Systems offers its Java SMB solution, jNQ™, which is portable and can be integrated into any environment, thus bringing SMB client capabilities to any Java-based device or platform under a commercial license.

jNQ™ 1.00 is now available for integration.