

=====
Rexx LA vSymposium 2020
Presentation
=====

DBM Key/Value Data Store APIs for
Net Rexx and BSF400Rexx

by Tony Dycks – September 24, 2020



Overview of APIs & Rexx Implementations Covered

- DBMs
 - [QDBM](#) – Quick DBM
 - [Tokyo Cabinet](#) – Successor Project to QDBM
- Rexx Implementation Code Examples (Java APIs)
 - NetRexx (v3.03GA or Later with Java 8)
 - [QDBM](#) / [Tokyo Cabinet](#)
 - BSF4ooRexx (v641-20200130-Beta with Java 8)
 - [QDBM](#) / [Tokyo Cabinet](#)

What Is a DBM?

- DBMs are ...
 - Key / Value Data Stores that are persisted as files
 - Data Structures are usually Hashes, Map or B Tree Structures
- DBMs aren't ...
 - Relational DB Management Systems (Not a DBMS; a DBM)
 - inherently Secure since they are file based rather than Server based.

Traits of Good DBM Implementations

- Data Files are Portable on Different Hardware and OS Platforms
- Upward and Downward Version Compatibility for APIs
- Easy to Install and Use APIs
- APIs are Consistent & Portable Across Different Language and Platforms & CPU Architectures
- APIs methods are Intuitive for the Developer

I. QDBM – Quick DBM

- Overview
 - DBM File Can Be a Hash or B+ Tree Data Structure
 - Keys and Values are Variable Length Serial Byte Data Types
 - In Hash Structure ...
 - Keys Are Unique
 - If Key is Reused Data Value is Overwritten
 - In B+ Tree Structure ...
 - Duplicate Keys Can Be Stored

II. QDBM – QDBM APIs

- APIs
 - Available for the Following Programming Languages:
 - C++
 - Java (JDK or SE Versions 7 and 8)
 - Perl (v5.8 Or Later)
 - Python (2.7.3 Or Later)
 - AND Rexx (via Java API; Subject of This Presentation)
 - Frequently Used Platforms for the APIs
 - Linux (Debian and RPM Packages and Source Tar Install Files)
 - Windows (Binary Zip File)
 - Mac OS X / Cygwin for Windows (Source Tar Install Files)

III. QDBM – Library Setups for Rexx

- Debian, Mint or Ubuntu Linux Distros
 - Install the Following Packages with Dependencies:
 - libqdbm-dev
 - libqdbm-java
- Linux Source Tar ball **<= Better Install Option**
 - Download:
<https://fallabs.com/qdbm/qdbm-1.8.78.tar.gz>
 - Source tar.gz File Also Contains Java API after extraction

IV. QDBM – Library Setups for Rexx

- Linux Source Tarball (cont.)
 - Build Pre-requisite for Java API:
 - **Recommendation:** Install 32 bit Java 8 SE (Oracle Website) or openjdk8 Java Package (Linux Distro) for Java Support
 - Add **JAVA_HOME** Environment Variable and check **configure.in** File so configure utility has ability to find **javac** executable file
 - Run the Following to Build Base Library with Admin Privileges
 - `./configure`
 - `make`
 - `make check`
 - `make install`

V. QDBM – Library Setups for Rexx

- Linux Source Tarball (cont.)
 - **File:** config.in Modification for java_home environment variable (this example uses openjdk8 for Linux on a CentOS v7 Linux Distro):
 - Cd to sub directory: **java**
 - MYJAVAHOME="\$JAVA_HOME"
 - In this Example: \$JAVA_HOME is set to: /usr/lib/jvm/java-openjdk
 - Same Steps to Build Java API with Admin rights:
 - ./configure
 - make
 - make check
 - make install

VI. QDBM – Library Setups for Rexx

- Linux Source Tarball (cont.)
 - Copy qdbm.jar file to: `$JAVA_HOME/jre/lib/ext`
 - Or Add Reference to Your `$Classpath` Environment Variable for your Development Environment for Java and Rexx
 - Specifications for the QDBM Java API:
 - <https://fallabs.com/qdbm/jspex.html>
 - Java API Code Examples:
 - <https://fallabs.com/qdbm/jspex.html#examples>

VI.+ QDBM – Win32 Lib Setups for Rexx

- Win32 Binary Package
 - Slightly Older Version: 1.8.75
 - Zip Archive File: **qdbm-1.8.75-win32.zip**
 - Download Link:
 - **<https://docs.huihoo.com/qdbm/>**
 - API DLLs:
 - **jqdbm.dll, libiconv-2.dll, mgwz.dll and qdbm.dll**
 - Jar File for Java API:
 - **qdbm.jar**
 - Use 32 Bit Versions of Java and ooRexx for Binary Compatibility

VII. QDBM – NetRexx Code Sample

- List All Key and Values Entries for a QDBM File Specified on the Command Shell Line
- NetRexx Sample File: listqdbm.nrx
- Compile of NetRexx Source to Java Class File:
 - `sh ./NetRexxC.sh listqdbm.nrx`
- Shell Syntax Example:
 - `java listqdbm womensworldcupchamps.qdb`

VIII. QDBM – NetRexx Sample Run

```
tonyd@dortmund:~/NetRexx/source
File Edit View Search Terminal Help
[tonyd@dortmund source]$ dir list*
listqdbm.class listqdbm.crossref listqdbm.nrx
[tonyd@dortmund source]$ java listqdbm $HOME/qdbm/appl/womensworldcupchamps.qdb

listqdbm.nrx -- NetRexx List QDBM Key Values For QDBM File Entered On Command L
ine
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: December 29, 2007
Date Revised: December 29, 2007

QDBM Filename Argument: /home/tonyd/qdbm/appl/womensworldcupchamps.qdb

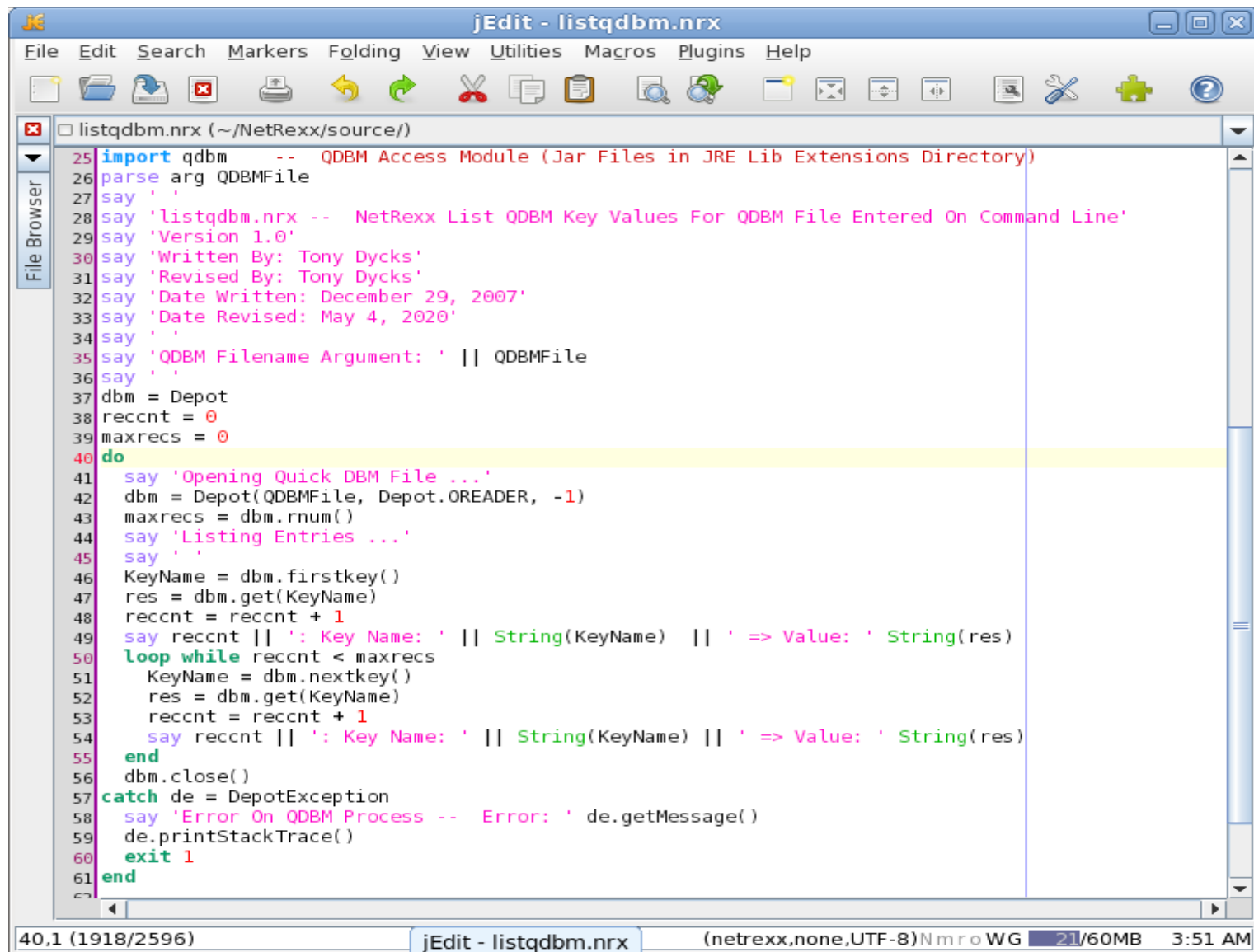
Opening Quick DBM File ...
Listing Entries ...

1: Key Name: 1991 => Value: United States
2: Key Name: 1995 => Value: Norway
3: Key Name: 1999 => Value: United States
4: Key Name: 2003 => Value: Germany
5: Key Name: 2007 => Value: Germany
6: Key Name: 2011 => Value: Japan
7: Key Name: 2015 => Value: United States
8: Key Name: 2019 => Value: United States

>>> End Of Program -- listqdbm.nrx <<<

[tonyd@dortmund source]$
```

IX. QDBM – Net Rexx Sample Code



```
jEdit - listqdbm.nrx
File Edit Search Markers Folding View Utilities Macros Plugins Help
listqdbm.nrx (~/.NetRexx/source/)
25 import qdbm -- QDBM Access Module (Jar Files in JRE Lib Extensions Directory)
26 parse arg QDBMFile
27 say ''
28 say 'listqdbm.nrx -- NetRexx List QDBM Key Values For QDBM File Entered On Command Line'
29 say 'Version 1.0'
30 say 'Written By: Tony Dycks'
31 say 'Revised By: Tony Dycks'
32 say 'Date Written: December 29, 2007'
33 say 'Date Revised: May 4, 2020'
34 say ''
35 say 'QDBM Filename Argument: ' || QDBMFile
36 say ''
37 dbm = Depot
38 reccnt = 0
39 maxrecs = 0
40 do
41 say 'Opening Quick DBM File ...'
42 dbm = Depot(QDBMFile, Depot.OREADER, -1)
43 maxrecs = dbm.rnum()
44 say 'Listing Entries ...'
45 say ''
46 KeyName = dbm.firstkey()
47 res = dbm.get(KeyName)
48 reccnt = reccnt + 1
49 say reccnt || ': Key Name: ' || String(KeyName) || ' => Value: ' String(res)
50 loop while reccnt < maxrecs
51 KeyName = dbm.nextkey()
52 res = dbm.get(KeyName)
53 reccnt = reccnt + 1
54 say reccnt || ': Key Name: ' || String(KeyName) || ' => Value: ' String(res)
55 end
56 dbm.close()
57 catch de = DepotException
58 say 'Error On QDBM Process -- Error: ' de.getMessage()
59 de.printStackTrace()
60 exit 1
61 end
62
```


40,1 (1918/2596) jEdit - listqdbm.nrx (netrexx,none,UTF-8)Nmr o WG 21/60MB 3:51 AM

X. QDBM – BSF400Rexx Code

- List The Version of QDBM Installed Using The Depot Object
- BSF400Rexx Sample File: ListQDBMVer.rex
- Command Shell Syntax Examples:
 - `sh ./rexxj2.sh ListQDBMVer.rex`
 - `rexx ListQDBMVer.rex`

XI. QDBM – BSF4ooRexx Run Output

- Displays Version of QDBM: 1.8.78



```
tonyd@dortmund:~/bsf4ooress/source
File Edit View Search Terminal Help
[tonyd@dortmund source]$ dir *QDBM*
AddQDBMCLEntry.rex  LstQDBMCLEntry.rex  QDBMDepotVer.rex  QDBMFlRecCnt.rex
[tonyd@dortmund source]$ rexx QDBMDepotVer.rex
This Rexx program was invoked by Rexx, JVM loaded by Rexx!

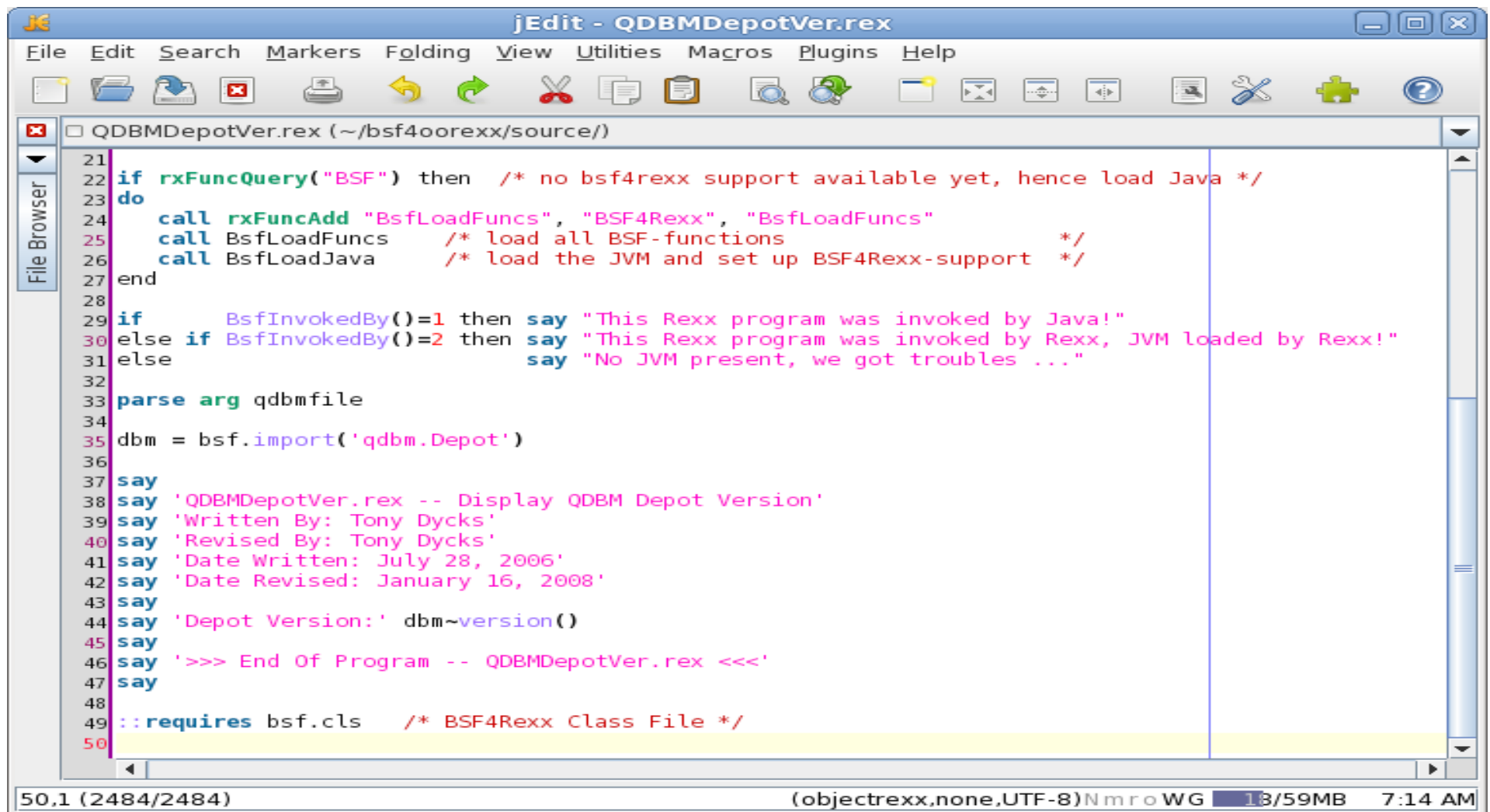
QDBMDepotVer.rex -- Display QDBM Depot Version
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: July 28, 2006
Date Revised: January 16, 2008

Depot Version: 1.8.78

>>> End Of Program -- QDBMDepotVer.rex <<<

[tonyd@dortmund source]$
```


XII. QDBM – QDBMDepotVer.rex Code



```
jEdit - QDBMDepotVer.rex
File Edit Search Markers Folding View Utilities Macros Plugins Help
QDBMDepotVer.rex (~/.bsf4ooorex/source/)
21 if rxFuncQuery("BSF") then /* no bsf4rexx support available yet, hence load Java */
22 do
23   call rxFuncAdd "BsfLoadFuncs", "BSF4Rexx", "BsfLoadFuncs"
24   call BsfLoadFuncs /* load all BSF-functions */
25   call BsfLoadJava /* load the JVM and set up BSF4Rexx-support */
26 end
27
28
29 if BsfInvokedBy()=1 then say "This Rexx program was invoked by Java!"
30 else if BsfInvokedBy()=2 then say "This Rexx program was invoked by Rexx, JVM loaded by Rexx!"
31 else say "No JVM present, we got troubles ..."
32
33 parse arg qdbmfile
34
35 dbm = bsf.import('qdbm.Depot')
36
37 say
38 say 'QDBMDepotVer.rex -- Display QDBM Depot Version'
39 say 'Written By: Tony Dycks'
40 say 'Revised By: Tony Dycks'
41 say 'Date Written: July 28, 2006'
42 say 'Date Revised: January 16, 2008'
43 say
44 say 'Depot Version:' dbm~version()
45 say
46 say '>>> End Of Program -- QDBMDepotVer.rex <<<'
47 say
48
49 ::requires bsf.cls /* BSF4Rexx Class File */
50
```

50,1 (2484/2484) (objectrexx,none,UTF-8)NmroWG 18/59MB 7:14 AM

XIII. QDBM – List of NetRexx Samples

- Additional NetRexx Samples ...
 - **AddQDBMCLEntry.nrx** -- NetRexx Add QDBM Key And Value For File, Key and Value Entered On Command Line
 - **LstQDBMCLEntry.nrx** -- NetRexx List QDBM Value For QDBM File & Key Entered On Command Line
 - **QDBMDepotVer.nrx** -- Display QDBM Depot Database Version

XIV. QDBM – List of BSF4ooRexx Samples

- Additional BSF4ooRexx Samples ...
 - **listqdbm.rex** -- BSF4Rexx List QDBM Key Values For QDBM File Entered On Command Line
 - **AddQDBMCLEntry.rex** -- BSF4ooRexx Add QDBM Key And Value For File, Key and Value Entered On Command Line
 - **LstQDBMCLEntry.rex** -- BSF4ooRexx List QDBM Value For QDBM File & Key Entered On Command Line
 - **QDBMFIRecCnt.rex** -- Display Record Count For QDBM File Command Line Argument

XV. QDBM – Java & Rexx Code Dev Experiences

- Most of the QDBM Depot Read and Write Objects could be coded in both NetRexx and BSF4ooRexx
- This Java Depot File Creation Connection Object Instance Resulted in An Error while attempting the NetRexx and BSF4ooRexx Equivalents:
 - `dbm = new Depot(sQDBMFileNm, Depot.OWRITER | Depot.OCREAT, -1); // Works using Java`

XVI. QDBM – NetRexx Code Dev Experiences

- Attempt to Code Depot Create with NetRexx
 - parse arg qdbmflnm
 - import qdbm.
 - dbm = Depot
 - dbm = null
 - dbm = Depot(qdbmflnm, Depot.OWRITER | Depot.OCREAT, -1) – **Compiles OK; Runtime Error**
 - dbm = Depot(qdbmflnm, Depot.OWRITER, -1) **--OK**

XVII. QDBM – NetRexx Java Exception Message

- Java Run-time Exception (New File Creation):
 - Exception in thread "main"
netrexx.lang.NotLogicException: Boolean must be 0 or 1. Found: 2
 - at netrexx.lang.Rexx.toboolean(Unknown Source)
 - at netrexx.lang.Rexx.OpOr(Unknown Source)
 - Line # References the Depot(...) File Creation Object Instance Source Code

XVIII. QDBM – BSF4ooRex Dev Code

- Attempt to Code Create Depot Instance with BSF4ooRexx
 - parse arg QDBMFile CSVFile
 - /* Import Java Class "qdbm.Depot" */
 - Depot = bsf.import("qdbm.Depot")
 - dbm = Depot~new(QDBMFile, Depot~OWRITER | Depot~OCREAT, -1) **/* Results in Error */**

XVIX. QDBM – BSF4ooRexx Error

- Command:
 - rexx ldqdbm.rex \$HOME/qdbm/appl/horseofyr.qdb
\$HOME/txtfiles/eclipsehorseofyr.txt
- Error Message:
 - This Rexx program was invoked by Rexx, JVM loaded by Rexx!
 - 38 *-* dbm = Depot~new(QDBMFile, Depot~OWRITER | Depot~OCREAT, -1)
 - REX0034E: Error 34 running /home/tonyd/bsf4oorex/source/ldqdbm.rex line 38: Logical value not 0 or 1
 - REX0338E: Error 34.901: Logical value must be exactly "0" or "1"; found "4"

I. Tokyo Cabinet

- Overview
 - Successor to QDBM; Adds Tokyo Tyrant Server
 - Increased File Size for Tokyo Cabinet DBM Objects
 - Faster Processing Speed for TC vs. QDBM
 - Improved API versus QDBM (Transaction Support)
 - TC Supports both 32-Bit & 64-Bit CPU Architectures
 - Added Programming Language APIs
 - Lua (v5.1 Only), Erlang, more...

II. Tokyo Cabinet – Base Library Setup

- RedHat Enterprise & CentOS 6 Linux Distros
 - `sudo yum install tokyocabinet`
 - `sudo yum install tokyocabinet-devel`
- Debian, Mint and Ubuntu Linux Distros
 - `sudo apt-get install tokyocabinet`
 - `sudo apt-get install tokyocabinet-dev`
- Source Tarball Download Link:
 - <https://dbmx.net//tokyocabinet/tokyocabinet-1.4.48.tar.gz>
- **Recommendation:** Avoid the use of the Distro Packages; Build from Source Tarball Instead
- Build steps are quite similar to QDBM Build

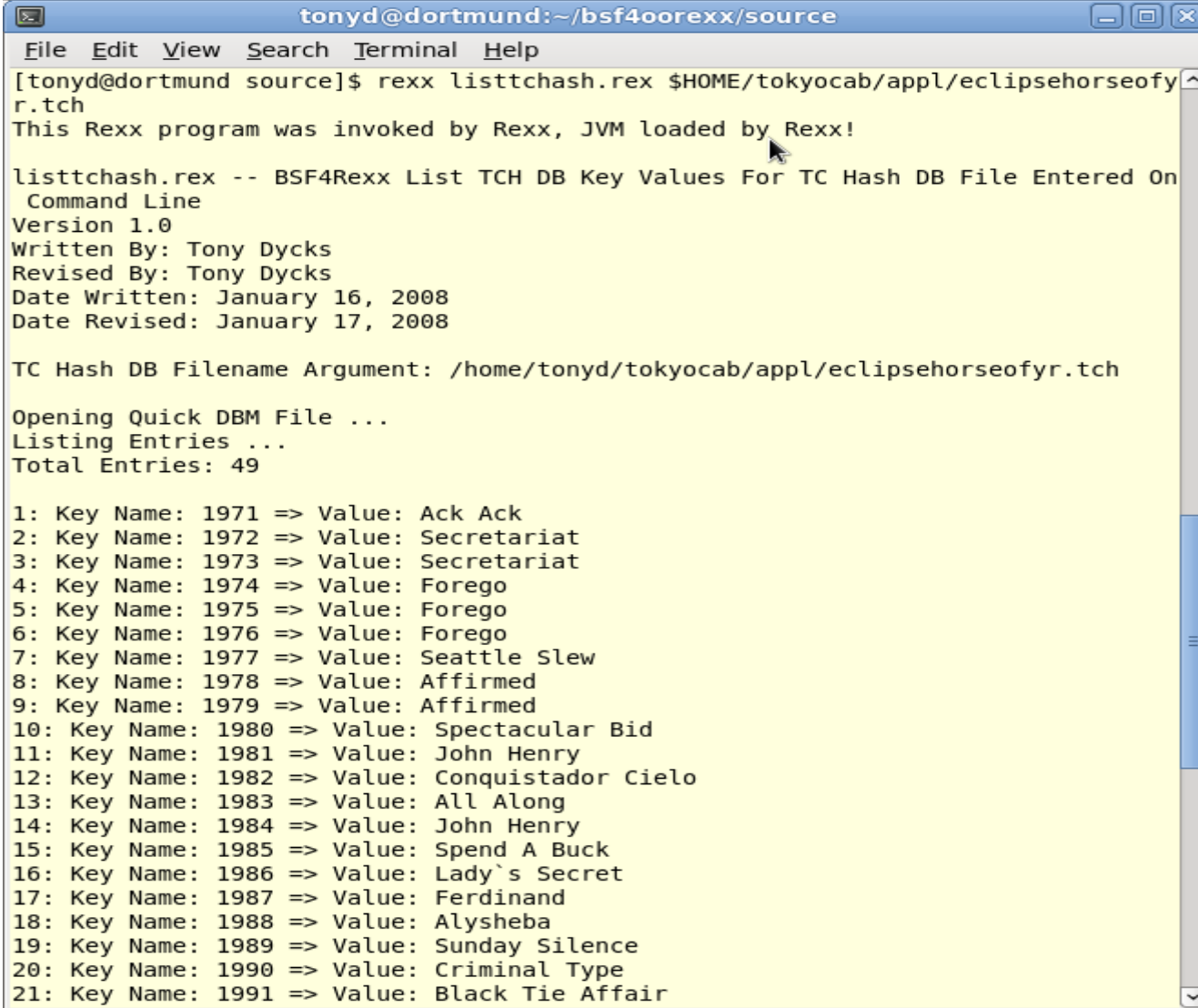
III. Tokyo Cabinet – Java API Setup

- Download The Source Tarball
 - <https://fallabs.com/tokyocabinet/javapkg/tokyocabinet-java-1.24.tar.gz>
- Setup (use Super User **su** vs. **sudo** if possible)
 - Modify config.in to Reflect \$JAVA_HOME Env
 - ./configure
 - make
 - make check
 - make install (Copies to /usr/local/lib)

IV. TC – BSF4ooRexx Code Sample

- List All Key and Values Entries for a TC Hash File Specified on the Command Shell Line
- BSF4ooRexx Sample File: listtchash.rex
- Linux Bash Shell Syntax Example:
 - rexx listtchash.rex
\$HOME/tokyocab/appl/eclipsehorseofyr.tch

V. TC – BSF4ooRexx Run Output



```
tonyd@dortmund:~/bsf4oorexx/source
File Edit View Search Terminal Help
[tonyd@dortmund source]$ rexx listtchash.rex $HOME/tokyocab/appl/eclipsehorseofyr.tch
This Rexx program was invoked by Rexx, JVM loaded by Rexx!

listtchash.rex -- BSF4Rexx List TCH DB Key Values For TC Hash DB File Entered On
Command Line
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 16, 2008
Date Revised: January 17, 2008

TC Hash DB Filename Argument: /home/tonyd/tokyocab/appl/eclipsehorseofyr.tch

Opening Quick DBM File ...
Listing Entries ...
Total Entries: 49

1: Key Name: 1971 => Value: Ack Ack
2: Key Name: 1972 => Value: Secretariat
3: Key Name: 1973 => Value: Secretariat
4: Key Name: 1974 => Value: Forego
5: Key Name: 1975 => Value: Forego
6: Key Name: 1976 => Value: Forego
7: Key Name: 1977 => Value: Seattle Slew
8: Key Name: 1978 => Value: Affirmed
9: Key Name: 1979 => Value: Affirmed
10: Key Name: 1980 => Value: Spectacular Bid
11: Key Name: 1981 => Value: John Henry
12: Key Name: 1982 => Value: Conquistador Cielo
13: Key Name: 1983 => Value: All Along
14: Key Name: 1984 => Value: John Henry
15: Key Name: 1985 => Value: Spend A Buck
16: Key Name: 1986 => Value: Lady`s Secret
17: Key Name: 1987 => Value: Ferdinand
18: Key Name: 1988 => Value: Alysheba
19: Key Name: 1989 => Value: Sunday Silence
20: Key Name: 1990 => Value: Criminal Type
21: Key Name: 1991 => Value: Black Tie Affair
```

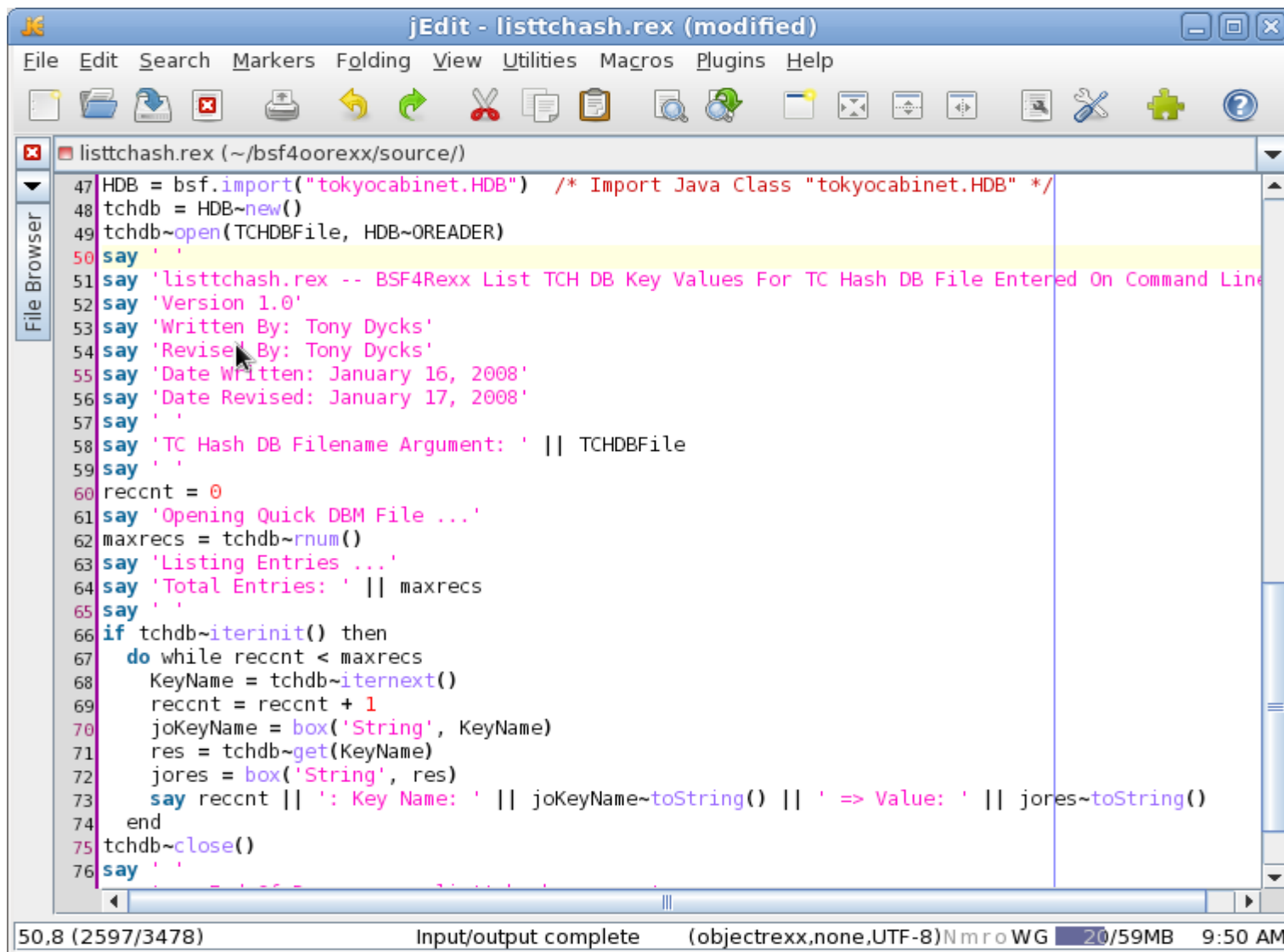
VI. TC – BSF400Rexx Run Output

```
tonyd@dortmund:~/bsf400rexx/source
File Edit View Search Terminal Help
16: Key Name: 1986 => Value: Lady`s Secret
17: Key Name: 1987 => Value: Ferdinand
18: Key Name: 1988 => Value: Alysheba
19: Key Name: 1989 => Value: Sunday Silence
20: Key Name: 1990 => Value: Criminal Type
21: Key Name: 1991 => Value: Black Tie Affair
22: Key Name: 1992 => Value: A.P. Indy
23: Key Name: 1993 => Value: Kotashaan
24: Key Name: 1994 => Value: Holy Bull
25: Key Name: 1995 => Value: Cigar
26: Key Name: 1996 => Value: Cigar
27: Key Name: 1997 => Value: Favorite Trick
28: Key Name: 1998 => Value: Skip Away
29: Key Name: 1999 => Value: Charismatic
30: Key Name: 2000 => Value: Tiznow
31: Key Name: 2001 => Value: Point Given
32: Key Name: 2002 => Value: Mineshaft
33: Key Name: 2003 => Value: Azeri
34: Key Name: 2004 => Value: Ghostzapper
35: Key Name: 2005 => Value: Saint Liam
36: Key Name: 2006 => Value: Invasor
37: Key Name: 2007 => Value: Curlin
38: Key Name: 2008 => Value: Curlin
39: Key Name: 2009 => Value: Rachel Alexandra
40: Key Name: 2010 => Value: Zenyatta
41: Key Name: 2011 => Value: Havre de Grace
42: Key Name: 2012 => Value: Wise Dan
43: Key Name: 2013 => Value: Wise Dan
44: Key Name: 2014 => Value: California Chrome
45: Key Name: 2015 => Value: American Pharoah
46: Key Name: 2016 => Value: California Chrome
47: Key Name: 2017 => Value: Gun Runner
48: Key Name: 2018 => Value: Justify
49: Key Name: 2019 => Value: Bricks and Mortar

>>> End Of Program -- listtchash.rex <<<

[tonyd@dortmund source]$ █
```

VII. TC – BSF4ooRexx Source Code

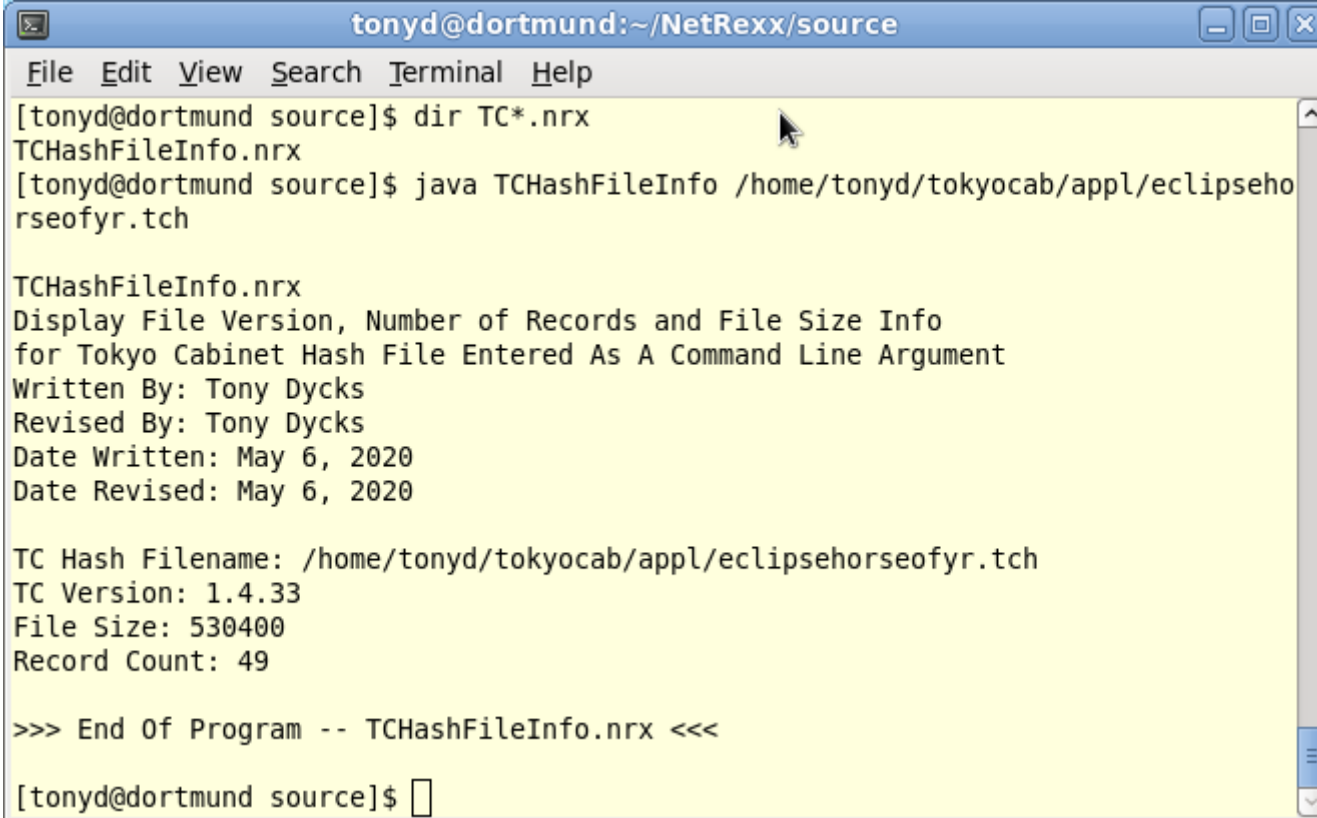


```
jEdit - listtchash.rexx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
listtchash.rexx (~/bsf4ooressource/)
47 HDB = bsf.import("tokyocabinet.HDB") /* Import Java Class "tokyocabinet.HDB" */
48 tchdb = HDB~new()
49 tchdb~open(TCHDBFile, HDB~OREADER)
50 say '
51 say 'listtchash.rexx -- BSF4Rexx List TCH DB Key Values For TC Hash DB File Entered On Command Line
52 say 'Version 1.0'
53 say 'Written By: Tony Dycks'
54 say 'Revised By: Tony Dycks'
55 say 'Date Written: January 16, 2008'
56 say 'Date Revised: January 17, 2008'
57 say '
58 say 'TC Hash DB Filename Argument: ' || TCHDBFile
59 say '
60 reccnt = 0
61 say 'Opening Quick DBM File ...'
62 maxrecs = tchdb~rnum()
63 say 'Listing Entries ...'
64 say 'Total Entries: ' || maxrecs
65 say '
66 if tchdb~iterinit() then
67   do while reccnt < maxrecs
68     KeyName = tchdb~iternext()
69     reccnt = reccnt + 1
70     joKeyName = box('String', KeyName)
71     res = tchdb~get(KeyName)
72     jores = box('String', res)
73     say reccnt || ': Key Name: ' || joKeyName~toString() || ' => Value: ' || jores~toString()
74   end
75 tchdb~close()
76 say '
50,8 (2597/3478) Input/output complete (objectrexx,none,UTF-8)NmroWG 20/59MB 9:50 AM
```

VII. TC – NetRexx Code Sample

- List Info for Tokyo Cabinet Hash DB File Specified on the Command Shell Lin
- NetRexx Sample File: TCHashFileInfo.nrx
- Linux Bash Shell NetRexx Compile Example:
 - `sh ./NetRexxC.sh TCHashFileInfo.nrx`
- Java Bash Shell Run Example:
 - `java TCHashFileInfo
$HOME/tokyocab/appl/eclipsehorseofyr.tch`

VIII. TC – NetRexx Run Output



```
tonyd@dortmund:~/NetRexx/source
File Edit View Search Terminal Help
[tonyd@dortmund source]$ dir TC*.nrx
TCHashFileInfo.nrx
[tonyd@dortmund source]$ java TCHashFileInfo /home/tonyd/tokyocab/appl/eclipsehorseofyr.tch

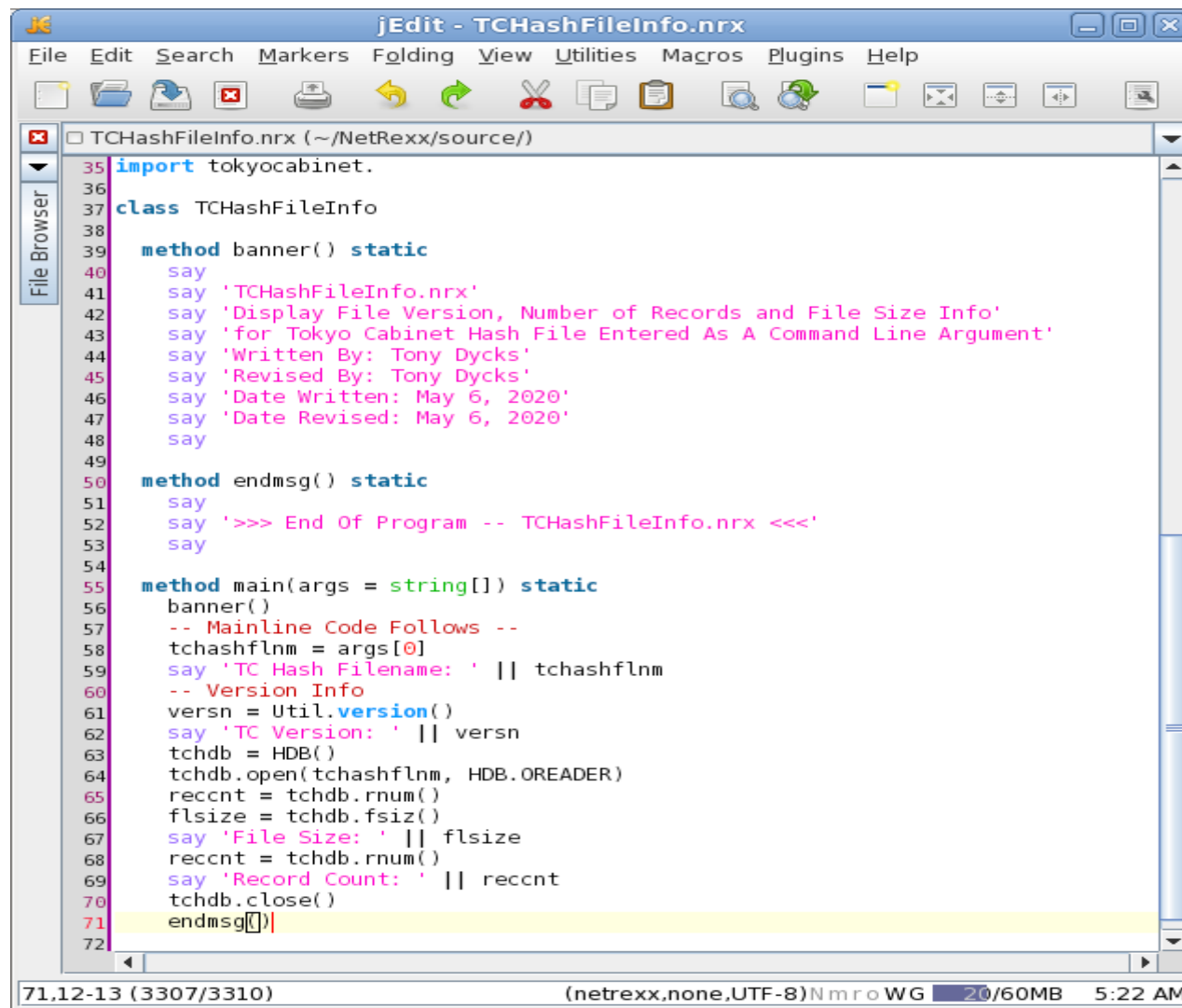
TCHashFileInfo.nrx
Display File Version, Number of Records and File Size Info
for Tokyo Cabinet Hash File Entered As A Command Line Argument
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: May 6, 2020
Date Revised: May 6, 2020

TC Hash Filename: /home/tonyd/tokyocab/appl/eclipsehorseofyr.tch
TC Version: 1.4.33
File Size: 530400
Record Count: 49

>>> End Of Program -- TCHashFileInfo.nrx <<<

[tonyd@dortmund source]$
```

IX. TC – NetRexx Source Code



```
35 import tokyocabinet.
36
37 class TCHashFileInfo
38
39     method banner() static
40         say
41         say 'TCHashFileInfo.nrx'
42         say 'Display File Version, Number of Records and File Size Info'
43         say 'for Tokyo Cabinet Hash File Entered As A Command Line Argument'
44         say 'Written By: Tony Dycks'
45         say 'Revised By: Tony Dycks'
46         say 'Date Written: May 6, 2020'
47         say 'Date Revised: May 6, 2020'
48         say
49
50     method endmsg() static
51         say
52         say '>>> End Of Program -- TCHashFileInfo.nrx <<<'
53         say
54
55     method main(args = string[]) static
56         banner()
57         -- Mainline Code Follows --
58         tchashflnm = args[0]
59         say 'TC Hash Filename: ' || tchashflnm
60         -- Version Info
61         versn = Util.version()
62         say 'TC Version: ' || versn
63         tchdb = HDB()
64         tchdb.open(tchashflnm, HDB.OREADER)
65         reccnt = tchdb.rnum()
66         flsize = tchdb.fsiz()
67         say 'File Size: ' || flsize
68         reccnt = tchdb.rnum()
69         say 'Record Count: ' || reccnt
70         tchdb.close()
71         endmsg()
72
```

71.12-13 (3307/3310) (netrexx.none,UTF-8)Nmro WG 20/60MB 5:22 AM

I. TC & QDBM -- Issues Encountered and Workarounds

- Lubuntu 18.04 LTS & Debian 9 Installation
- Tokyo Cabinet Install Issue
 - Buggy Debian Package Library Installations for **libtokyocabinet** and **libtokyocabinet-dev**
 - Could Print Version Info for Tokyo Cabinet Hash File, but no Record Count or Key/Value Fields
- **Workaround**
 - Build libtokyocabinet Modules Completely from Source Package Download from Fall Labs

II. TC & QDBM -- Issues Encountered and Workarounds

- Linux Distros Build Defaults
- Any Source Based Build to **/usr/local** dirs
- Both QDBM & Tokyo Cabinet
- Java API Setup from Source
 - Could not locate **javac** compiler for **su** login
- **Workaround**
 - From **su** login # ...
 - **export JAVA_HOME**=to base directory of Open JDK Installation (varies widely for different distros)



III. TC & QDBM -- Issues Encountered and Workarounds

- **Python PIP Library Modules**
 - toyko-python
 - pyqdbm
 - PIP Install could not Find Pre-Requirement QDBM & TC Library Headers
- **Workaround (admittedly Quick & Dirty)**
 - Required Pre-Requirement QDBM & TC Library Modules & Headers were Copied from /usr/local/lib to /usr/lib for successful setup of PIP Module

I. QDBM & TC – Raspberry Pi 4 64-Bit Linux OS Installs

- Able to Build QDBM and TC Libraries from Source Package on R Pi4 for ARM 64-Bit OS
 - Manjaro Linux for Raspberry Pi 4 v20.5
- Ubuntu Aarch64 18.04 LTS and 19.10 Binary Install Packages are Available for ARM 64-Bit R Pi 4
 - libqdbm & libqdbm-dev
 - Tokyo Cabinet Libraries required build from source on Ubuntu
- Built Java APIs from Source Package for both QDBM and Tokyo Cabinet

Portability Considerations

- QDBM Portability Findings 
 - QDBM Files Ported To Other PCs Required Reloading of the Data To Function On The Transported PC
- Tokyo Hash File Findings 
 - Tokyo Cabinet Hash Files Could Be Transported to Other PCs Without Requiring Reload of The Data

Findings & Recommendations

- QDBM and Tokyo Cabinet
 - **Platform:** Stay with Linux; Avoid Windows
 - Build QDBM and TC From Most Current Source Packages
 - Ditto for Java APIs; Set **\$JAVA_HOME** for Root User Before Configure
 - Use OpenJDK 7 or 8; Match Bitness of ooRexx Install with JDK
 - Dependent Library Packages can be Installed Using Linux Distro Package Manager (**yum** for CentOS; **apt** for Debian Based Distros)
 - Load data on each PC for Initial Query for QDBM vs. File Migration
 - Use Java or Python Programs for Initial QDBM or TC Hash File Creation
 - NetRexx and BSF4ooRexx programs can be used for Query and Update functionality; Use Java or Other Program API to Create Initial DBM Files

Findings & Recommendations+

- Tokyo Cabinet
 - Requires Development Headers for Bzip2; Ok to Use Linux Distro Install Packages
 - Debian Packages: libbz2 and libbz2-dev
 - RPM Packages: libbz2 and libbz2-devel
 - Tokyo Cabinet Hash Files Can Be Migrated From One Computer to Another readily without Rebuilding the File
 - Use Tokyo Cabinet over QDBM when feasible for Improved File Portability and Access to more Program APIs

Appendix A: List of Technologies

Software	Version(s)	Architecture	Notes
CentOS, Arch, Ubuntu & Debian Linux	6.10 and 7.4, Rolling, 18.04LTS & 9.0-10.4	32 & 64 Bit (i386, i586, i686, armv7+, aarch64)	Also Windows 10 Pro (QDBM Binary)
Java 8 Open JDK	1.8.251 & 1.8.252	32 and 64 Bit (i386, i586 & i686, armv7+, aarch64)	RPM, Deb & Win Packages
NetRexx	3.03 GA or Later	Architecture Independent	Jar Install File v3.03 Zip Archive v3.04 +
ooRexx	4.2 or Later	32 and 64 Bit	RPM & Deb Pkgs
QDBM API	1.8.78 (1.8.75 Win)	32 and 64 Bit	Source (Binary)
Tokyo Cabinet API	1.4.48	32 and 64 Bit	Mix Binary & Source
JEdit	5.3 or later	32 and 64 Bit	Jar File or RPM & Deb Packages
Python 2 (pyqdbm & tokyo-python)	2.7.3 or later	32 and 64 Bit	RPM & Deb Pkgs plus PIP installs for libs

Appendix B: List of References

Software	Author	URL	Foot Notes
QDBM – Quick Database Manager	Mikio Hirabashi	https://dbmx.net/qdbm/	Overview & Links for QDBM Download
QDBM-- Java API Examples	Mikio Hirabashi	https://dbmx.net/qdbm/jspex.html	Specifications of QDBM for Java
QDBM Windows Binary Download	Mikio Hirabashi	https://docs.huihoo.com/qdbm/	Download Binary for QDBM for Windows
Open Object Rexx	Rexx Language Association	https://sourceforge.net/projects/oorexx/	Source Forget Site for ooRexx
BSF4ooRexx	BSF4ooRexx Project Team	https://sourceforge.net/projects/bsf4oorexx/	Source Forge Site for Downloads
Tokyo Cabinet API	Mikio Hirabashi	https://dbmx.net//tokyocabinet/	Overview and Download Links for Tokyo Cabinet
Tokyo Cabinet API – Javadoc	Mikio Hirabashi	https://dbmx.net//tokyocabinet/javadoc/	Javadoc of API

Appendix C: List of References+

Software	Organization	URL	Foot Notes
Net Rexx v3.03GA	NetRexx.org	http://netrexx.org/downloads.nsp	Net Rexx Downloads
Jedit v5.3 Text Edit Program	Jedit.org	http://jedit.org/	Jedit Downloads; Also Available as Linux Package for most Distros

Wrapping It Up

- Questions?
- Zip File of Sample Programs and Data Files Available on Rexx LA Website Presentations Page
- Thanks for Your Time and Interest

