



OPERATING SYSTEM &
HARDWARE INFORMATION



Cross Platform & Architecture Rexx Programs using the OSHI Java API

Rexx Language Association Symposium 2021
Presenter: Tony Dycks – November 9, 2021

Overview



OPERATING SYSTEM &
HARDWARE INFORMATION

- **What is OSHI** – Overview of the OSHI Java Library API
- **Supported OS Platforms** – Windows, Linux, MacOS, UNIX Dialects
- **Rexx and Java Implementations** – NetRexx and BSF4ooRexx
- **OSHI API Features** – OS and Hardware API Functionality
- **OSHI Java Library Requirements** – Required Jar Files for Pre-requisites
- **Rexx Program Requirements** – Rexx Dialects / Console Shells / Edit Programs
- **Java Setup for Examples** – How To Install and Configure Java for Code Samples
- **Java/ooRexx Compatibility** – Keep the Architecture Bitness Consistent
- **OS Architecture Compatibility** – Test Findings on the Different Hardware
- **NetRexx Sample Programs** – Examples of Programs Written in NetRexx
- **BSF4ooRexx Sample Programs** – Examples of Programs Written in BSF4ooRexx
- **Summary of Test Findings** – What Worked; What Didn't Work
- **Presentation References** – List of References for Presentation

What is OSHI?

- **OSHI** – Operating System & Hardware Information API
- **OSHI** is a free JNA-based (native) Operating System and Hardware Information library for Java
- **JNA** – Java Native Access API based
- Cross-platform Implementation
- Functionality
 - Retrieve System Information
 - OS version
 - System Processes
 - Memory and CPU usage
 - Disks and Partitions
 - Devices
 - Sensors
- **Project Repository**
 - **URL:** <https://github.com/oshi/oshi>
 - Current Stable Release: **v5.8**
 - Jar File: **oshi-core-5.8.jar**
 - Additional details available on the **readme.md** file for the GitHub project

Supported OS Platforms

- **Platforms Supported:**
 - **Windows**
 - **Linux**
 - **MacOS**
 - UNIX dialects
 - **AIX**
 - **FreeBSD**
 - **Open BSD**
 - **Solaris**
- **CPU Architectures:**
 - **Intel** (Complete API)
 - **ARM** (Subset of API)
- **Requirements:**
 - **Java Environments**
 - Tested Versions
 - Oracle Java SE 8 Development Kit
 - OpenJDK 8 and 11

Rexx & Java Implementations

- **Rexx** Implementations that Support **Java**:
 - **NetRexx**
 - Versions Demonstrated:
 - 3.03 GA and 3.09 GA
 - **BSF4ooRexx**
 - V641
- **Java** Implementations used in this Presentation
 - Oracle Java SE 8 Development Kit
 - OpenJDK 8 (a.k.a. OpenJDK 1.8 in the Red Hat Package World)
- **CPU Architectures** used in this Presentation
 - Intel 64-Bit
 - Core i5 and i7 Quad core, Various CPU Frequencies
 - Raspberry PI Model 4B 64-Bit
 - Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- **OS Platforms** Programs Demonstrated ...
 - Windows 10 Professional 64-bit (amd64) for Intel i5
 - Debian Linux 10.10 Buster 64-Bit (amd64) for Intel i5
 - OpenSUSE Leap 15.3 64-Bit (aarch64) for Raspberry Pi 4B

OSHI API Features

- **Computer System and Firmware**, Baseboard
- **Operating System and Version/Build**
- **Physical** (core) and **Logical** (hyperthreaded) **CPUs**, processor groups
- **System** and per processor **load**, usage tick counters, interrupts, uptime
- Process Uptime, CPU, Memory Usage, User/Group, Thread details
- **Physical and Virtual memory** Used/Available
- Mounted File Systems (type, usable and total space, options, reads and writes)
- **Disk drives** (model, serial, size, reads and writes) and **Partitions**
- **Network Interfaces** (IPs, bandwidth in/out), network parms, TCP/UDP statistics
- **Battery** state (% capacity, time remaining, power usage stats)
- **USB** Devices
- **Connected displays** (with EDID info), graphics and audio cards
- **Sensors** (temperature, fan speeds, voltage) on some hardware

OSHI Java Library Requirements

- **Java Implementation Requirements**
 - Java Library Prerequisites for OSHI
 - ==> **jna** (Linux) or **jna-platform** (Windows) – Java Native Access
 - ==> **slf4j** – Systems logging facility for Java
 - Versions Used for Presentation
 - ==> **jna** – Version 5.6
 - ==> **slfj** – Version 1.7
 - For Java v8 Implementation deployments ...
 - Download and install the .jar files in Directory:
 - **\$JAVA_HOME/jre/lib/ext**
 - Most Linux Distros can use the **jna** .jar file
 - Windows 10 and FreeBSD require the **jna-platform** .jar file
 - SLF4J Requirement
 - Jar file is required, but there is **no coding requirement to use the logging facility**

Rexx Program Requirements

- Implementations of Rexx that Support Java APIs
 - **NetRexx**
 - Used Version 3.09GA (Well suited for Java 8 along with 3.11GA)
 - **BSF4ooRexx**
 - V641 (Using Rexx 5.0 Beta for this presentation)
- All of the Rexx Program Examples demonstrated are **Console Shell** Based
- There is a **oshi-demo** Java program that utilizes the Swing GUI Library
- **Console Shells** Used:
 - **Command Prompt** (Windows 10 Professional)
 - **Bourne-Again Shell (BASH)** (Raspberry Pi OS and Debian Linux)
 - **Xfce Terminal** (openSUSE 15.3)
- Text Editor or IDE of your Choice (an editor that uses **Rexx** syntax highlighting is a big plus)
- Recommendations: Open Source Text Editors with Rexx Syntax Awareness
 - **JEdit** (Java Based; Cross Platform; Windows, Linux & FreeBSD)
 - **Notepad++** (Windows or WINE emulator in Linux)

Java Setup for Examples

Linux Distros

OpenJDK8 or OpenJDK 1.8 Package

Open JDK Install Commands ...

Debian Family Distros (Pre Version 10)

```
sudo apt-get install openjdk-8-jdk
```

RedHat / Fedora 33-34

```
sudo dnf install java-1.8.0-openjdk
```

openSUSE Leap 15.2 or 15.3

```
sudo zypper install java-1_8_0-openjdk
```

Latest Versions of Debian, Ubuntu & Linux Mint No Longer Offer OpenJDK8*

* - OpenJDK 8 for Debian Linux 10 Available from a Separate Debian Repository

For Windows 10 OS & Linux Distros That No Longer Support OpenJDK8

Opt for **Oracle Java 8 SE Development Kit** Installation

Oracle Account Required to Download Java SE (**Requires Sign-up**)

Download Link URLs:

Java 8 SE JRE:

<https://www.oracle.com/java/technologies/javase-jre8-downloads.html>

Java 8 SE Development Kit:

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

Java / ooRexx Compatibility - I

Recommendation:

Install **matching Architecture Bitness** Versions of ooRexx and Java
To Verify Bitness (Sample output from openSUSE Leap 15.3 aarch64 for R Pi4B) ...

Java JDK – java -version

```
tonyd@Classier:~> /usr/lib64/jvm/java-1.8.0-openjdk-1.8.0/bin/java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (IcedTea 3.19.0) (build 1.8.0_292-b10 suse-3.52.1-aarch64)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
tonyd@Classier:~>
```

ooRexx – rexx -V

```
tonyd@Classier:~> rexx -V
Open Object Rexx Version 5.0.0 r12282
Build date: Jul 27 2021
Addressing mode: 64
Copyright (c) 1995, 2004 IBM Corporation. All rights reserved.
Copyright (c) 2005-2021 Rexx Language Association. All rights reserved.
This program and the accompanying materials are made available under the terms
of the Common Public License v1.0 which accompanies this distribution or at
https://www.oorexx.org/license.html
tonyd@Classier:~>
```

Java / ooRexx Compatibility - II

Testing Findings with OSHI Java Library:

==> Java / ooRexx / BSF4ooRexx / OSHI Technology Stack:

For Java –

OpenJDK 8 or Java SE 8 will work with OSHI; Also OpenJDK 11
I did not test any of the newer versions of Java newer than 8

For ooRexx –

OSHI will work on both Versions 4.2 and 5.0 Beta with BSF4ooRexx

For BSF4ooRexx –

v641 Versions Utilized, which will support Java from Version 6 upward

==> Java / NetRexx / OSHI Technology Stack:

For Java –

OpenJDK 8 or Java SE 8 will work with OSHI

For NetRexx -

Versions 3.03GA, **3.09GA** and 3.11GA were tested with OSHI

OS Architecture Compatibility

OS Platform Architecture Test Findings:

Intel:

- Nearly all of the OSHI APIs Used by the Sample Programs Worked Well For Both Windows 10 Professional and Intel Based Linux Distros
- Also worked well with FreeBSD v12.2 using NetRexx & OpenJDK
- I Was unable to build the ooRexx 5.0 package for FreeBSD
- No issues with CPU Bitness within the Intel family of CPUs
- OSHI Code Worked on both a Celeron and Intel Core i7 CPU

Broadcom ARM for the RPi4 Model B:

- A subset of the OSHI APIs Used by the Sample Programs Worked
- For Both arm7l and aarch64 Linux Architectures on the Raspberry Pi 4
- JNA Error on Version 13.0 of FreeBSD with NetRexx & OpenJDK
(Class Not Found Exception for JNA Library in JNA v5.6-5.8)

NetRexx Program Samples - 1

Program – Description

- 1) **OSHIGetCPUAll.nrx** – Get All CPU Info as a String in Net Rexx using OSHI Java Library
- 2) **OSHIGetCPUIdInfo.nrx** – Get CPU Id Info in Net Rexx using OSHI Java Library
- 3) **OSHIGetOS.nrx** – Get OS Name, Family, Manufacturer and Bitness Using OSHI Java Library
- 4) **OSHIOSFISysDtls.nrx** – Show Details of OS File System Using OSHI Java Library
- 5) **OSHIGetTtlAvailUsedRAM.nrx** – Total, Available & Used RAM Info Using OSHI Java Library

NetRexx Compile to .class file Syntax:

Linux

```
sh ./NetRexxC.sh <NetRexx-program-name>
```

Windows

```
NetRexxC.bat <NetRexx-program-name>
```

Java Run .class File

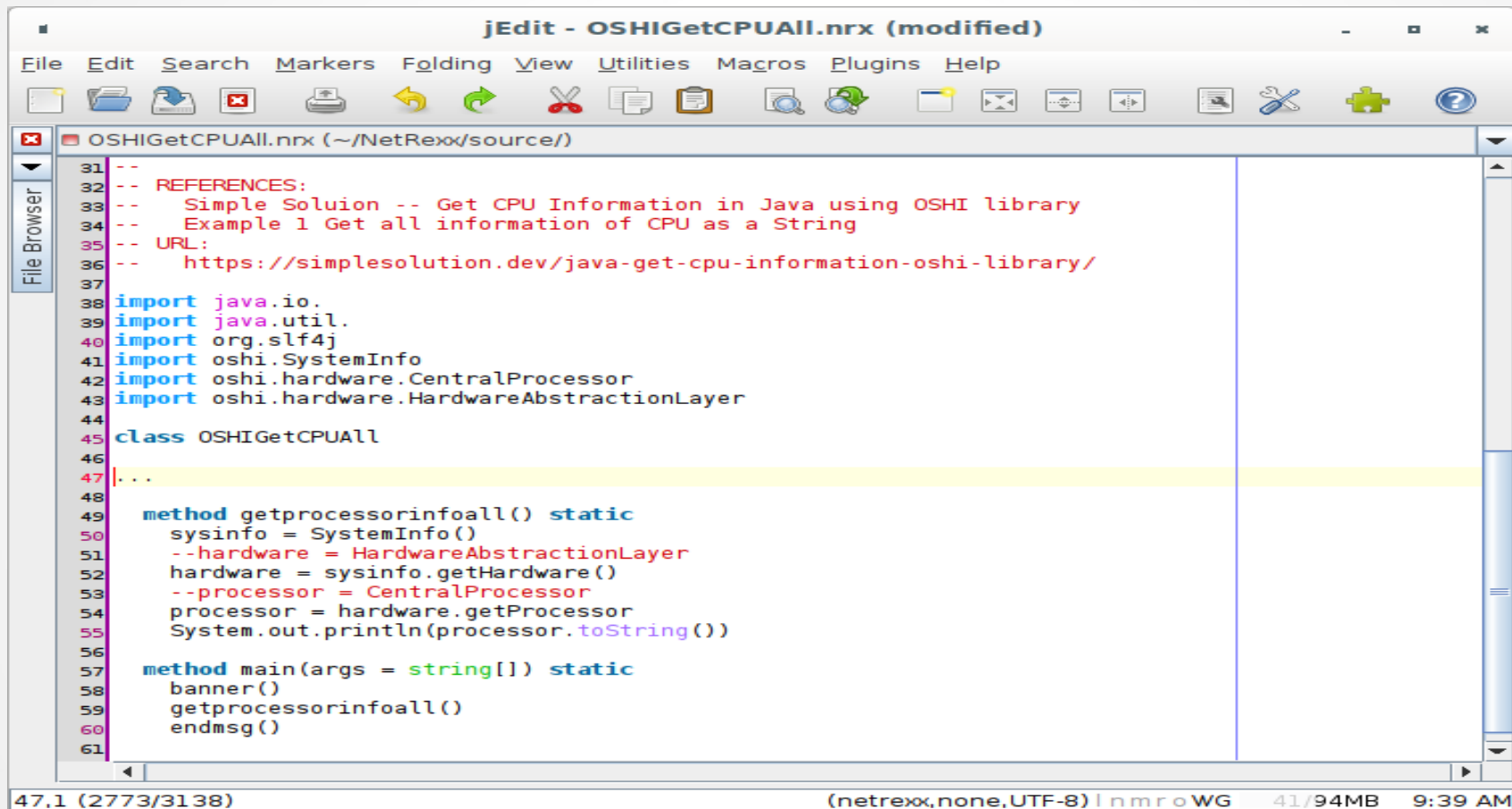
```
java <NetRexx-program-class-file-without-extension>
```

Examples follow ...

NetRexx Program Samples - 1.1

OSHIGetCPUAll.nrx – Get All CPU Info as a String in Net Rexx using OSHI Java Library

NetRexx Source Code Snippet:



```
jEdit - OSHIGetCPUAll.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetCPUAll.nrx (~/NetRexx/source/)
31 --
32 -- REFERENCES:
33 -- Simple Solution -- Get CPU Information in Java using OSHI library
34 -- Example 1 Get all information of CPU as a String
35 -- URL:
36 -- https://simplesolution.dev/java-get-cpu-information-oshi-library/
37
38 import java.io.
39 import java.util.
40 import org.slf4j
41 import oshi.SystemInfo
42 import oshi.hardware.CentralProcessor
43 import oshi.hardware.HardwareAbstractionLayer
44
45 class OSHIGetCPUAll
46
47 |...
48
49 method getprocessorinfoall() static
50     sysinfo = SystemInfo()
51     --hardware = HardwareAbstractionLayer
52     hardware = sysinfo.getHardware()
53     --processor = CentralProcessor
54     processor = hardware.getProcessor
55     System.out.println(processor.toString())
56
57 method main(args = string[]) static
58     banner()
59     getprocessorinfoall()
60     endmsg()
61
47,1 (2773/3138) (netrex,none,UTF-8) | nmroWG 41/94MB 9:39 AM
```

NetRexx Program Samples - 1.2

Sample Compile of **OSHIGetCPUAll.nrx** to .class File for Linux:

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help
tonyd@Expressum:~/NetRexx/source$ dir OSHI*.nrx
OSHIGetCPUAll.nrx      OSHIGetOS.nrx          OSHIOSFlSysDtls.nrx
OSHIGetCPUIdInfo.nrx  OSHIGetTtlAvailUsedRAM.nrx
tonyd@Expressum:~/NetRexx/source$ sh ./NetRexxC.sh OSHIGetCPUAll.nrx
NetRexx portable processor 3.09-GA build 148-20200930-1931
Copyright (c) RexxLA, 2011,2020. All rights reserved.
Parts Copyright (c) IBM Corporation, 1995,2008.
Program OSHIGetCPUAll.nrx
=== class OSHIGetCPUAll ===
  function banner
  function endmsg
  function getprocessorinfoall
  function main(String[])
Compilation of 'OSHIGetCPUAll.nrx' successful
tonyd@Expressum:~/NetRexx/source$
```

NetRexx Program Samples - 1.3

Sample Run of `OSHIGetCPUAll.class` File (Debian Linux 10.10):

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help
tonyd@Expressum:~/NetRexx/source$ java OSHIGetCPUAll

OSHIGetCPUAll.nrx
Verion 1.0
Get All CPU Info as a String in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz
 1 physical CPU package(s)
 2 physical CPU core(s)
 4 logical CPU(s)
Identifier: Intel64 Family 6 Model 42 Stepping 7
ProcessorID: AFC1FBFF002006A7
Microarchitecture: Sandy Bridge (Client)

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

tonyd@Expressum:~/NetRexx/source$ █
```


NetRexx Program Samples - 1.4

Sample Run of `OSHIGetCPUAll.class` File (openSUSE Linux 15.3):

```
Terminal - tonyd@Classier:~/NetRexx/source
File Edit View Terminal Tabs Help
tonyd@Classier:~/NetRexx/source> java OSHIGetCPUAll

OSHIGetCPUAll.nrx
Verion 1.0
Get All CPU Info as a String in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

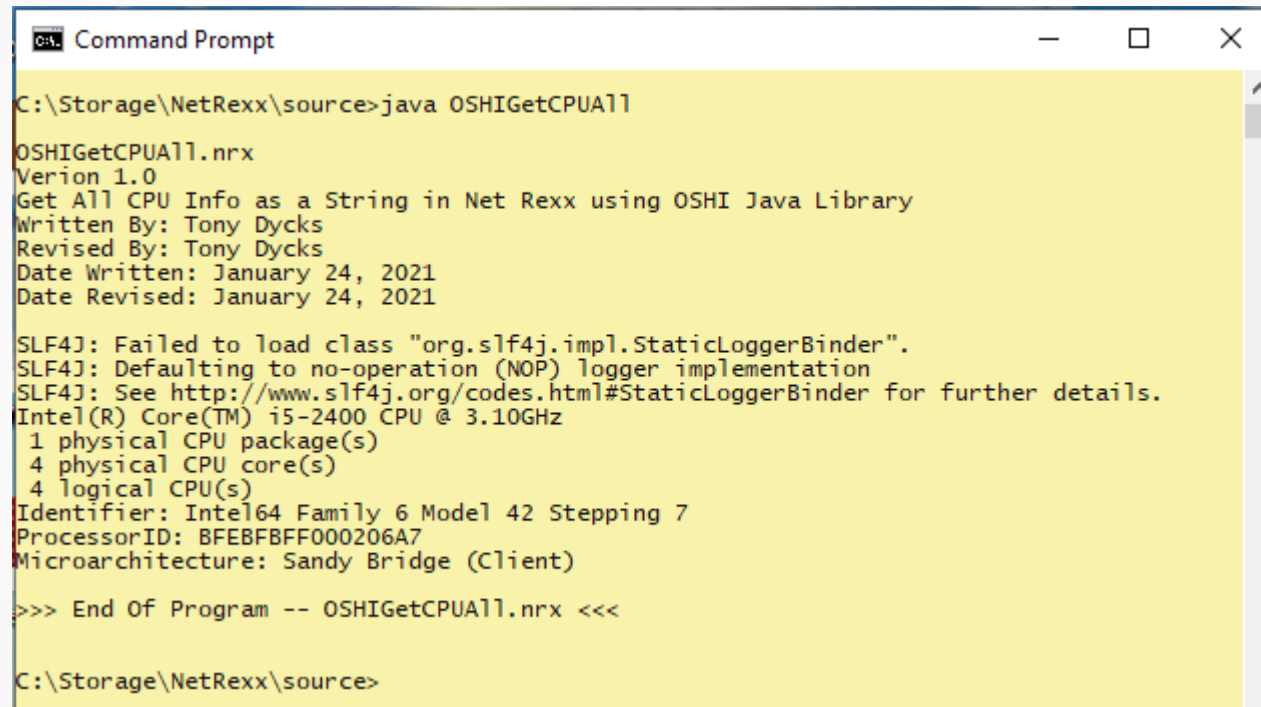
  1 physical CPU package(s)
  1 physical CPU core(s)
  4 logical CPU(s)
Identifier: aarch64 Family 8 Model 0xd08 Stepping r0x0p3
ProcessorID: 4108D08300000000
Microarchitecture: unknown

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

tonyd@Classier:~/NetRexx/source>
```

NetRexx Program Samples - 1.5

Sample Run of `OSHIGetCPUAll.class` File (Windows 10 Professional):



```
Command Prompt
C:\Storage\NetRexx\source>java OSHIGetCPUAll

OSHIGetCPUAll.nrx
Version 1.0
Get All CPU Info as a String in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
 1 physical CPU package(s)
 4 physical CPU core(s)
 4 logical CPU(s)
Identifier: Intel64 Family 6 Model 42 Stepping 7
ProcessorID: BFEFBFF000206A7
Microarchitecture: Sandy Bridge (Client)

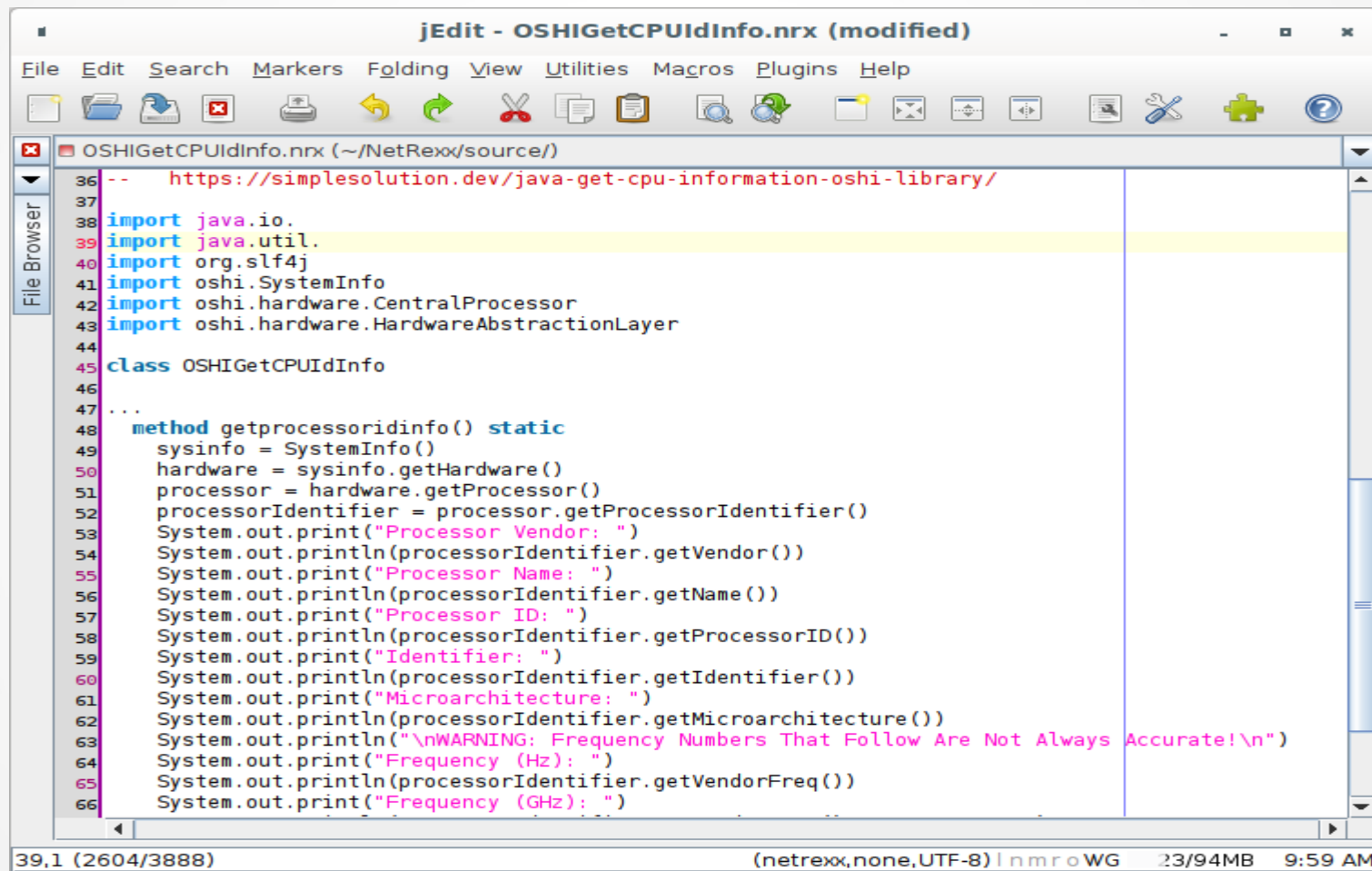
>>> End Of Program -- OSHIGetCPUAll.nrx <<<

C:\Storage\NetRexx\source>
```

NetRexx Program Samples - 2.1

OSHIGetCPUIdInfo.nrx – Get CPU Id Info in Net Rexx using OSHI Java Library

NetRexx Code Snippet:



```
jEdit - OSHIGetCPUIdInfo.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetCPUIdInfo.nrx (~/.NetRexx/source/)
36 -- https://simplsolution.dev/java-get-cpu-information-oshi-library/
37
38 import java.io.
39 import java.util.
40 import org.slf4j
41 import oshi.SystemInfo
42 import oshi.hardware.CentralProcessor
43 import oshi.hardware.HardwareAbstractionLayer
44
45 class OSHIGetCPUIdInfo
46 ...
47 ...
48 method getprocessoridinfo() static
49 sysinfo = SystemInfo()
50 hardware = sysinfo.getHardware()
51 processor = hardware.getProcessor()
52 processorIdentifier = processor.getProcessorIdentifier()
53 System.out.print("Processor Vendor: ")
54 System.out.println(processorIdentifier.getVendor())
55 System.out.print("Processor Name: ")
56 System.out.println(processorIdentifier.getName())
57 System.out.print("Processor ID: ")
58 System.out.println(processorIdentifier.getProcessorID())
59 System.out.print("Identifier: ")
60 System.out.println(processorIdentifier.getIdentifier())
61 System.out.print("Microarchitecture: ")
62 System.out.println(processorIdentifier.getMicroarchitecture())
63 System.out.println("\nWARNING: Frequency Numbers That Follow Are Not Always Accurate!\n")
64 System.out.print("Frequency (Hz): ")
65 System.out.println(processorIdentifier.getVendorFreq())
66 System.out.print("Frequency (GHz): ")
```

39,1 (2604/3888) (netrex,none,UTF-8) | nmr oWG 23/94MB 9:59 AM

NetRexx Sample Program - 2.2

Sample Run of `OSHIGetCPUIdInfo.class` File (Debian Linux 10.10) :

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help
tonyd@Expressum:~/NetRexx/source$ java OSHIGetCPUIdInfo

OSHIGetCPUIdInfo.nrx
Verion 1.0
Get CPU Id Info in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Processor Vendor: GenuineIntel
Processor Name: Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz
Processor ID: AFC1FBFF002006A7
Identifier: Intel64 Family 6 Model 42 Stepping 7
Microarchitecture: Sandy Bridge (Client)

WARNING: Frequency Numbers That Follow Are Not Always Accurate!

Frequency (Hz): 2500000000
Frequency (GHz): 2.5

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

tonyd@Expressum:~/NetRexx/source$
```

NetRexx Sample Program - 2.3

Sample Run of `OSHIGetCPUIdInfo.class` File (openSUSE 15.3) :

```
Terminal - tonyd@Classier:~/NetRexx/source
File Edit View Terminal Tabs Help
OSHIGetCPUIdInfo.nrx
Verion 1.0
Get CPU Id Info in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

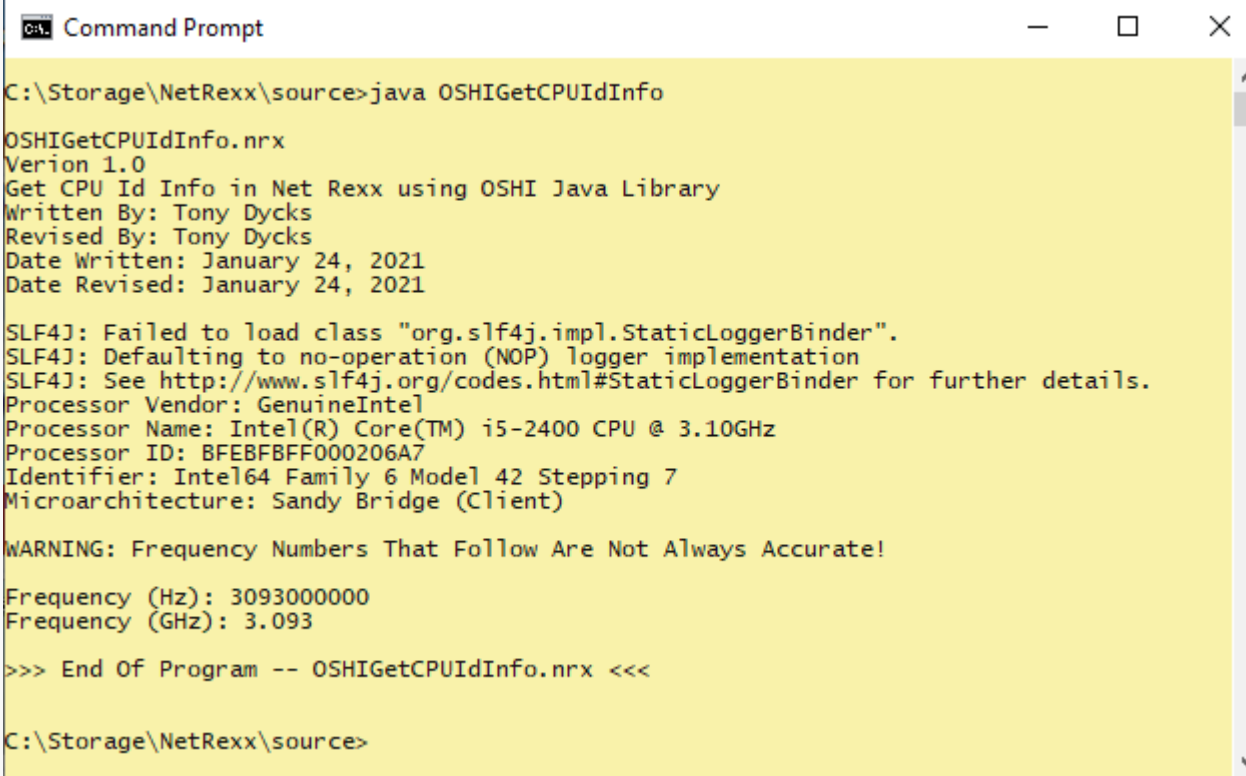
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further detail
s.
Processor Vendor: aarch64
Processor Name:
Processor ID: 4108D08300000000
Identifier: aarch64 Family 8 Model 0xd08 Stepping r0x0p3
Microarchitecture: unknown

WARNING: Frequency Numbers That Follow Are Not Always Accurate!
Frequency (Hz): -1
Frequency (GHZ): -1E-9

>>> End Of Program -- OSHIGetCPUIdInfo.nrx <<<
```

NetRexx Sample Program - 2.4

Sample Run of **OSHIGetCPUIdInfo.class** File (Windows 10 Professional) :



```
Command Prompt
C:\Storage\NetRexx\source>java OSHIGetCPUIdInfo
OSHIGetCPUIdInfo.nrx
Version 1.0
Get CPU Id Info in Net Rexx using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 24, 2021
Date Revised: January 24, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Processor Vendor: GenuineIntel
Processor Name: Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
Processor ID: BFEBFBFF000206A7
Identifier: Intel64 Family 6 Model 42 Stepping 7
Microarchitecture: Sandy Bridge (Client)

WARNING: Frequency Numbers That Follow Are Not Always Accurate!

Frequency (Hz): 3093000000
Frequency (GHz): 3.093

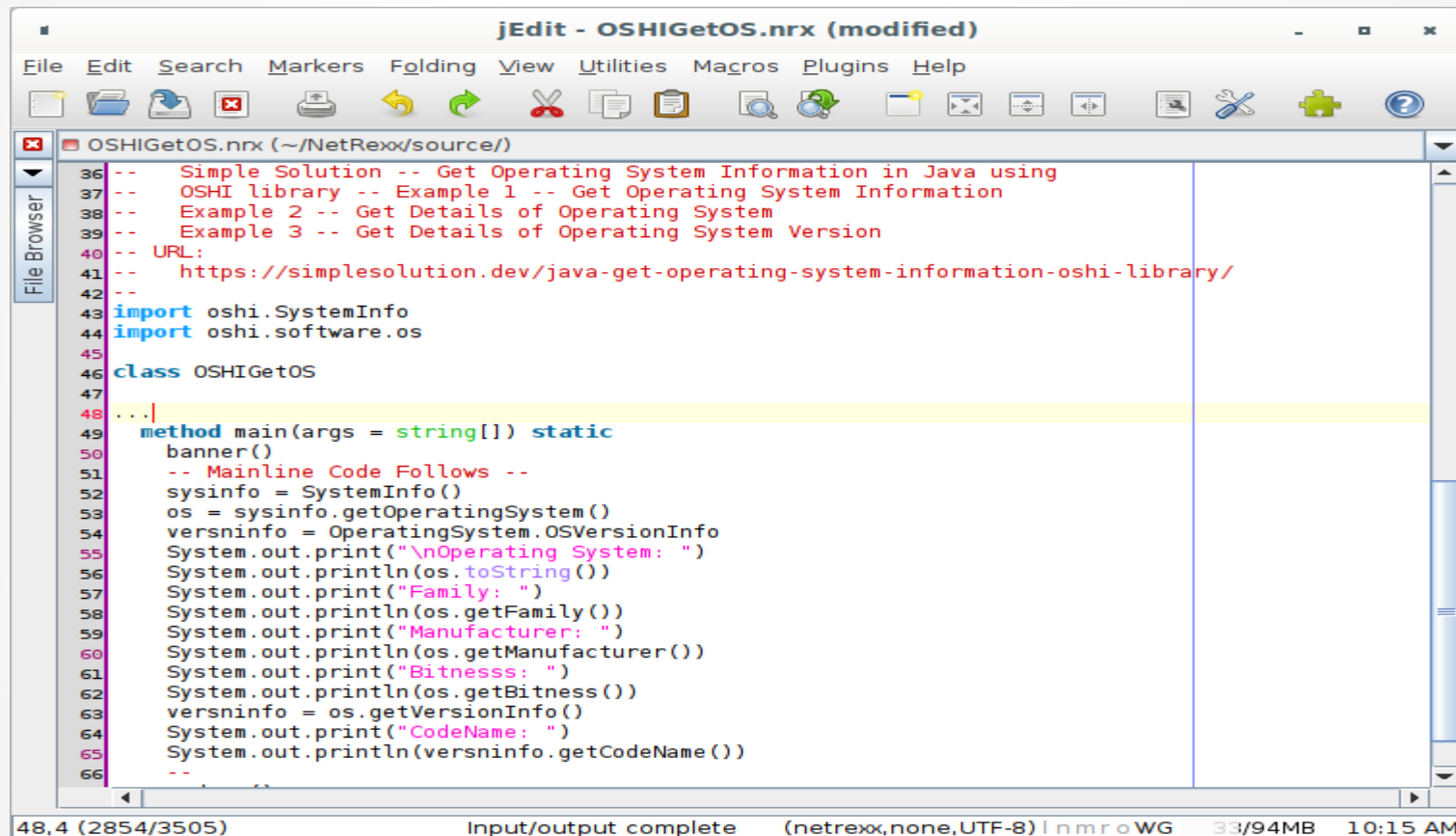
>>> End Of Program -- OSHIGetCPUIdInfo.nrx <<<

C:\Storage\NetRexx\source>
```

NetRexx Sample Programs - 3.1

OSHIGetOS.nrx – Get OS Name, Family, Manufacturer and Bitness Using OSHI Java Library

NetRexx Code Snippet:



```
jEdit - OSHIGetOS.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetOS.nrx (~/.NetRexx/source/)
36 -- Simple Solution -- Get Operating System Information in Java using
37 -- OSHI library -- Example 1 -- Get Operating System Information
38 -- Example 2 -- Get Details of Operating System
39 -- Example 3 -- Get Details of Operating System Version
40 -- URL:
41 -- https://simplsolution.dev/java-get-operating-system-information-oshi-library/
42 --
43 import oshi.SystemInfo
44 import oshi.software.os
45
46 class OSHIGetOS
47
48 ...
49 method main(args = string[]) static
50     banner()
51     -- Mainline Code Follows --
52     sysinfo = SystemInfo()
53     os = sysinfo.getOperatingSystem()
54     versninfo = OperatingSystem.OSVersionInfo
55     System.out.print("\nOperating System: ")
56     System.out.println(os.toString())
57     System.out.print("Family: ")
58     System.out.println(os.getFamily())
59     System.out.print("Manufacturer: ")
60     System.out.println(os.getManufacturer())
61     System.out.print("Bitness: ")
62     System.out.println(os.getBitness())
63     versninfo = os.getVersionInfo()
64     System.out.print("CodeName: ")
65     System.out.println(versninfo.getCodeName())
66     --
```

48,4 (2854/3505) Input/output complete (netrexx,none,UTF-8) | nmroWG 33/94MB 10:15 AM

NetRexx Sample Programs - 3.2

Sample Run of `OSHIGetOS.class` File (Debian Linux 10.10 Buster) :

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help
tonyd@Expressum:~/NetRexx/source$ java OSHIGetOS

OSHIGetOS.nrx
Verion 1.0
Get OS Name, Family and Bitness Using OSHI Java Library
With NetRexx
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 20, 2021
Date Revised: March 20, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Operating System: GNU/Linux Debian GNU/Linux 10 (buster) build 4.19.0-17-amd64
Family: Debian GNU/Linux
Manufacturer: GNU/Linux
Bitness: 64
CodeName: buster

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

tonyd@Expressum:~/NetRexx/source$
```


NetRexx Sample Programs - 3.3

Sample Run of `OSHIGetOS.class` File (Raspberry Pi OS 64-Bit) :

```
Terminal - tonyd@Classier:~/NetRexx/source
File Edit View Terminal Tabs Help

OSHIGetOS.nrx
Verion 1.0
Get OS Name, Family and Bitness Using OSHI Java Library
With NetRexx
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 20, 2021
Date Revised: March 20, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Operating System: GNU/Linux openSUSE Leap 15.3 (unknown) build 5.3.18-59.16-default
Family: openSUSE Leap
Manufacturer: GNU/Linux
Bitness: 64
CodeName: unknown

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

tonyd@Classier:~/NetRexx/source>
```

NetRexx Sample Programs - 3.4

Sample Run of **OSHIGetOS.class** File (Windows 10 Professional) :



```
Command Prompt
C:\Storage\NetRexx\source>java OSHIGetOS
OSHIGetOS.nrx
Verion 1.0
Get OS Name, Family and Bitness Using OSHI Java Library
With NetRexx
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 20, 2021
Date Revised: March 20, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Operating System: Microsoft Windows 10.0 build 19042
Family: Windows
Manufacturer: Microsoft
Bitness: 64
CodeName:

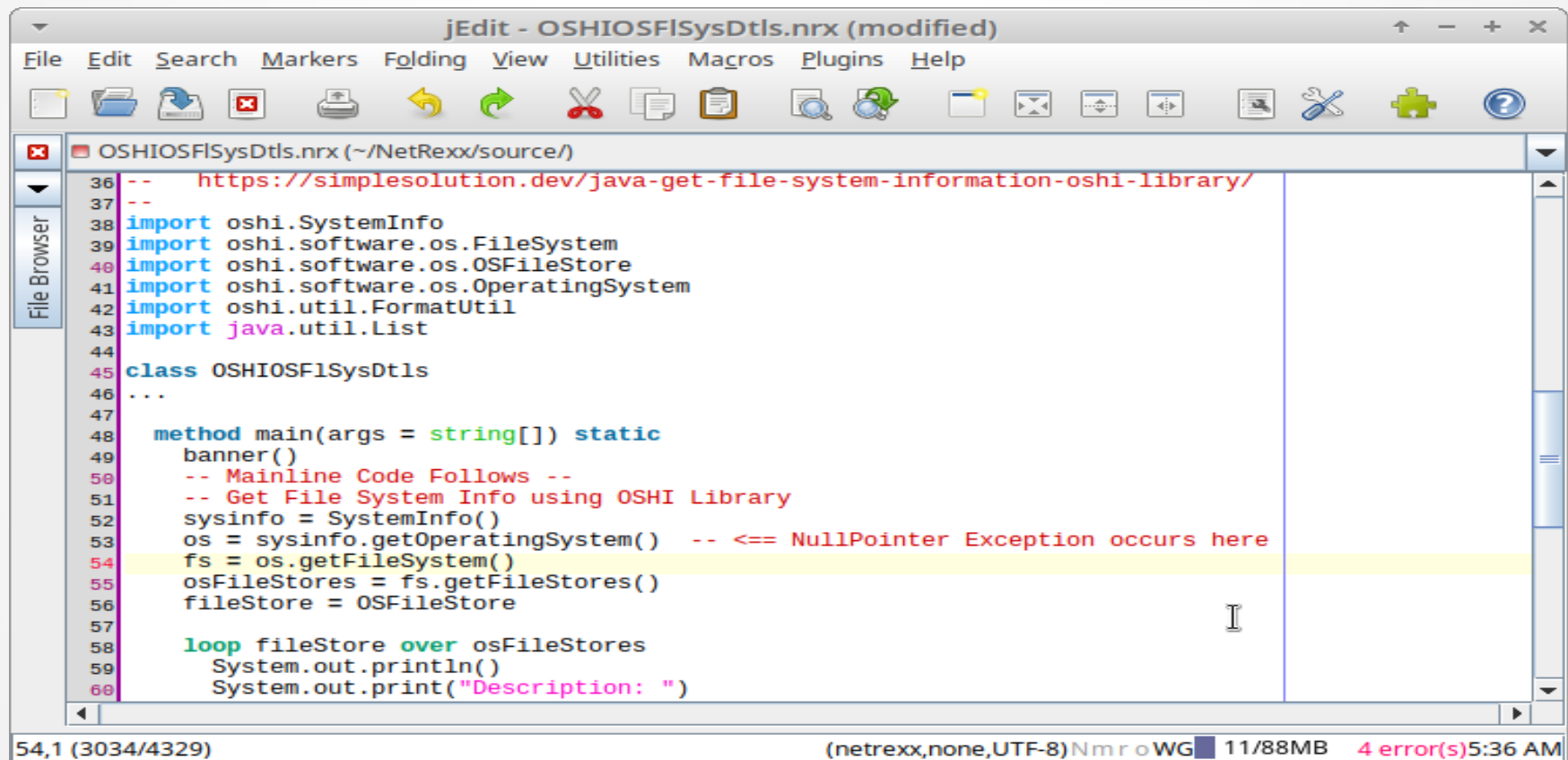
>>> End Of Program -- OSHIGetOS.nrx <<<

C:\Storage\NetRexx\source>
```

NetRexx Sample Programs - 4.1.1

OSHIOSFISysDtIs.nrx – Show Details of OS File System Using OSHI Java Library:

NetRexx Code Snippet (Part 1 of 2):



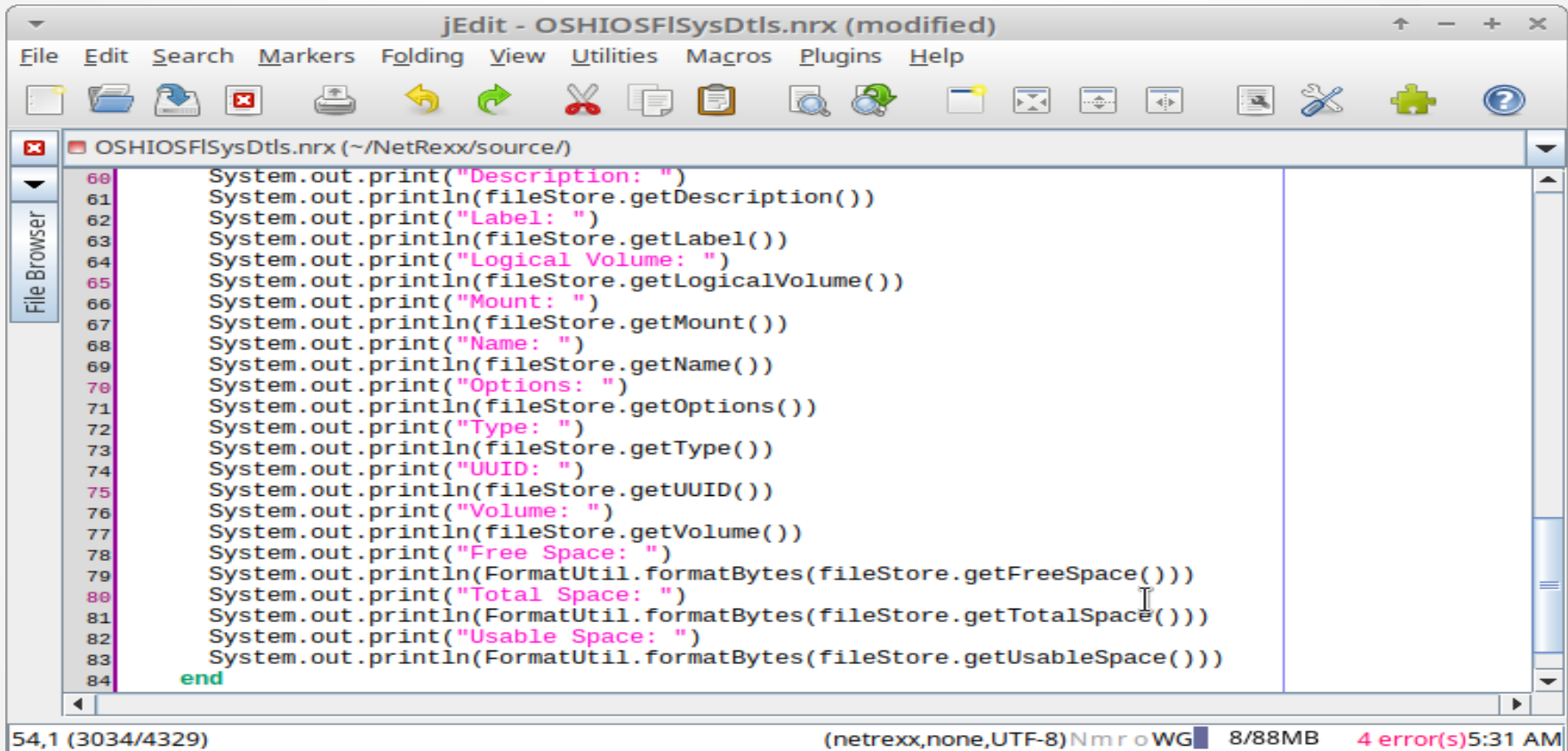
```
jEdit - OSHIOSFISysDtIs.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIOSFISysDtIs.nrx (~/.NetRexx/source/)
36 -- https://simplsolution.dev/java-get-file-system-information-oshi-library/
37 --
38 import oshi.SystemInfo
39 import oshi.software.os.FileSystem
40 import oshi.software.os.OSFileStore
41 import oshi.software.os.OperatingSystem
42 import oshi.util.FormatUtil
43 import java.util.List
44
45 class OSHIOSF1SysDtIs
46 ...
47
48 method main(args = string[]) static
49     banner()
50     -- Mainline Code Follows --
51     -- Get File System Info using OSHI Library
52     sysinfo = SystemInfo()
53     os = sysinfo.getOperatingSystem() -- <== NullPointerException occurs here
54     fs = os.getFileSystem()
55     osFileStores = fs.getFileStores()
56     fileStore = OSFileStore
57
58     loop fileStore over osFileStores
59         System.out.println()
60         System.out.print("Description: ")
```

54,1 (3034/4329) (netrex,none,UTF-8)Nm r oWG 11/88MB 4 error(s)5:36 AM

NetRexx Sample Programs - 4.1.2

OSHIOSFISysDtls.nrx – Show Details of OS File System Using OSHI Java Library:

NetRexx Code Snippet (Part 2 of 2):



```
jEdit - OSHIOSFISysDtls.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIOSFISysDtls.nrx (~/.NetRexx/source/)
60 System.out.print("Description: ")
61 System.out.println(fileStore.getDescription())
62 System.out.print("Label: ")
63 System.out.println(fileStore.getLabel())
64 System.out.print("Logical Volume: ")
65 System.out.println(fileStore.getLogicalVolume())
66 System.out.print("Mount: ")
67 System.out.println(fileStore.getMount())
68 System.out.print("Name: ")
69 System.out.println(fileStore.getName())
70 System.out.print("Options: ")
71 System.out.println(fileStore.getOptions())
72 System.out.print("Type: ")
73 System.out.println(fileStore.getType())
74 System.out.print("UUID: ")
75 System.out.println(fileStore.getUUID())
76 System.out.print("Volume: ")
77 System.out.println(fileStore.getVolume())
78 System.out.print("Free Space: ")
79 System.out.println(FormatUtil.formatBytes(fileStore.getFreeSpace()))
80 System.out.print("Total Space: ")
81 System.out.println(FormatUtil.formatBytes(fileStore.getTotalSpace()))
82 System.out.print("Usable Space: ")
83 System.out.println(FormatUtil.formatBytes(fileStore.getUsableSpace()))
84 end
54,1 (3034/4329) (netrex,none,UTF-8)Nmr o WG 8/88MB 4 error(s)5:31 AM
```

NetRexx Sample Programs - 4.2

Sample Run of `OSHIOSFISysDtls.class` File (Debian Linux 10.10) :

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help

Description: Local Disk
Label:
Logical Volume:
Mount: /media/tonyd/A6A8-AAEE
Name: /dev/sdb1
Options: rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmtask=0022,dmask=0022,codepage=437,ioccharset=ascii,shortname=mixed,showexec,utf8,flush,errors=remount-ro
Type: vfat
UUID: a6a8-aaee
Volume: /dev/sdb1
Free Space: 11.8 GiB
Total Space: 57.3 GiB
Usable Space: 11.8 GiB

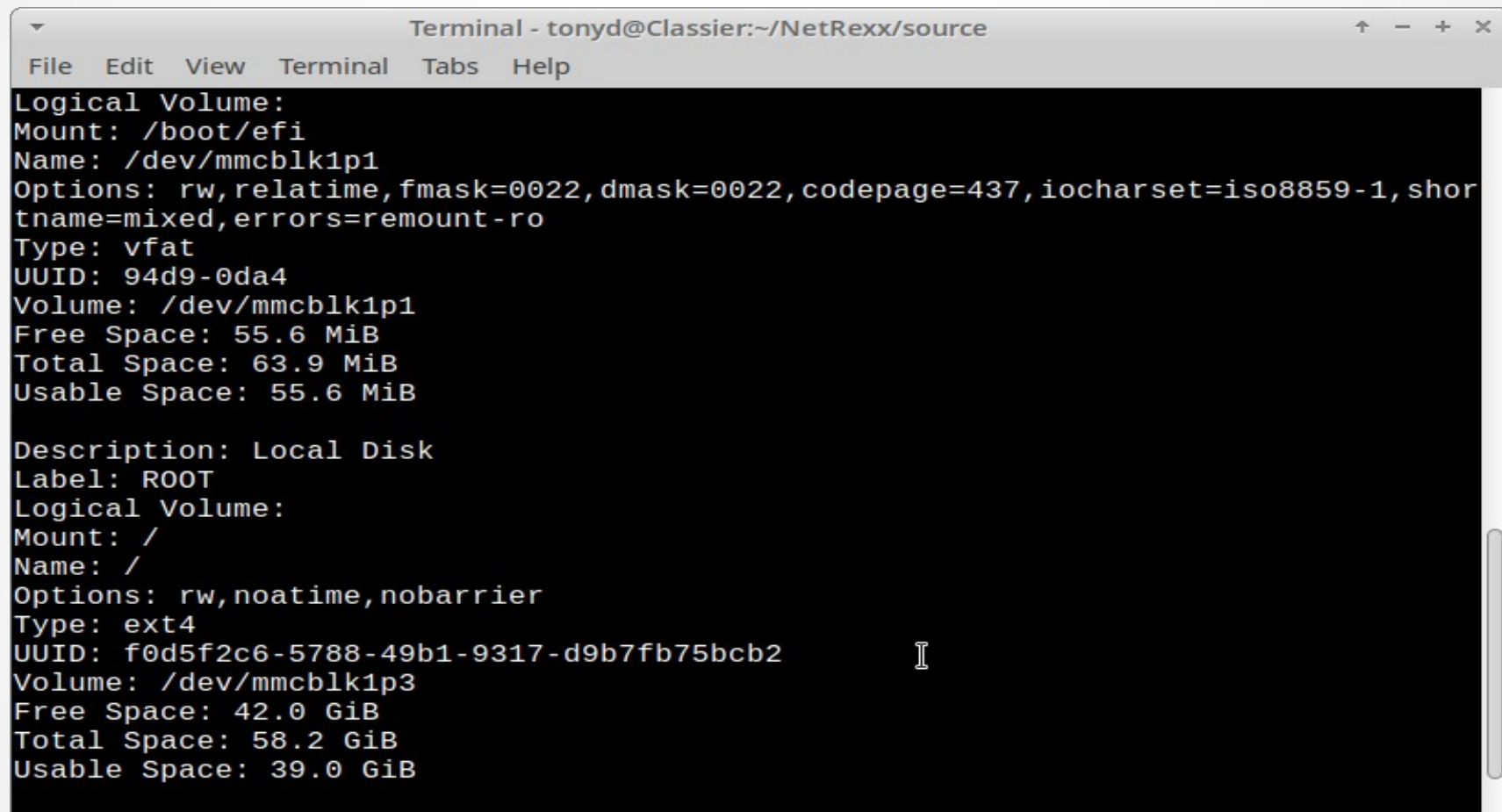
Description: Local Disk
Label:
Logical Volume:
Mount: /
Name: /
Options: rw,noatime
Type: ext4
UUID: 98b6f954-d09d-46ea-ad8b-45e7e4b1beb8
Volume: /dev/sda1
Free Space: 657.1 GiB
Total Space: 678.0 GiB
Usable Space: 622.6 GiB

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

tonyd@Expressum:~/NetRexx/source$
```

NetRexx Sample Programs - 4.3

Sample Run of `OSHIOSFISysDtls.class` File (openSUSE15.3) :



```
Terminal - tonyd@Classier:~/NetRexx/source
File Edit View Terminal Tabs Help
Logical Volume:
Mount: /boot/efi
Name: /dev/mmcblk1p1
Options: rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=iso8859-1,shortname=mixed,errors=remount-ro
Type: vfat
UUID: 94d9-0da4
Volume: /dev/mmcblk1p1
Free Space: 55.6 MiB
Total Space: 63.9 MiB
Usable Space: 55.6 MiB

Description: Local Disk
Label: ROOT
Logical Volume:
Mount: /
Name: /
Options: rw,noatime,nobarrier
Type: ext4
UUID: f0d5f2c6-5788-49b1-9317-d9b7fb75bcb2
Volume: /dev/mmcblk1p3
Free Space: 42.0 GiB
Total Space: 58.2 GiB
Usable Space: 39.0 GiB
```

NetRexx Sample Programs - 4.4

Sample Run of `OSHIOSFISysDtIs.class` File (Windows 10 Professional 64 Bit) :

```
Command Prompt
Description: CD-ROM
Label:
Logical Volume:
Mount: E:\
Name: CD-ROM Disc (E:)
Options: rw
Type:
UUID: be475d3f-4555-11ea-a12e-e83935410934
Volume: \\?\Volume{be475d3f-4555-11ea-a12e-e83935410934}\
Free Space: 0 bytes
Total Space: 0 bytes
Usable Space: 0 bytes

Description: CD-ROM
Label:
Logical Volume:
Mount: D:\
Name: CD-ROM Disc (D:)
Options: rw
Type:
UUID: e99820b3-37cd-11ea-a12b-806e6f6e6963
Volume: \\?\Volume{e99820b3-37cd-11ea-a12b-806e6f6e6963}\
Free Space: 0 bytes
Total Space: 0 bytes
Usable Space: 0 bytes

Description: Fixed drive
Label: Windows
Logical Volume:
Mount: C:\
Name: Local Fixed Disk (C:)
Options: rw,reparse,sparse,trans,journald,quota,casess,oids,casepn,efs,streams,unicode
,acls,fcomp
Type: NTFS
UUID: 9cf0c889-0000-0000-0000-f01500000000
Volume: \\?\Volume{9cf0c889-0000-0000-0000-f01500000000}\
Free Space: 205.6 GiB
Total Space: 464.9 GiB
Usable Space: 205.6 GiB

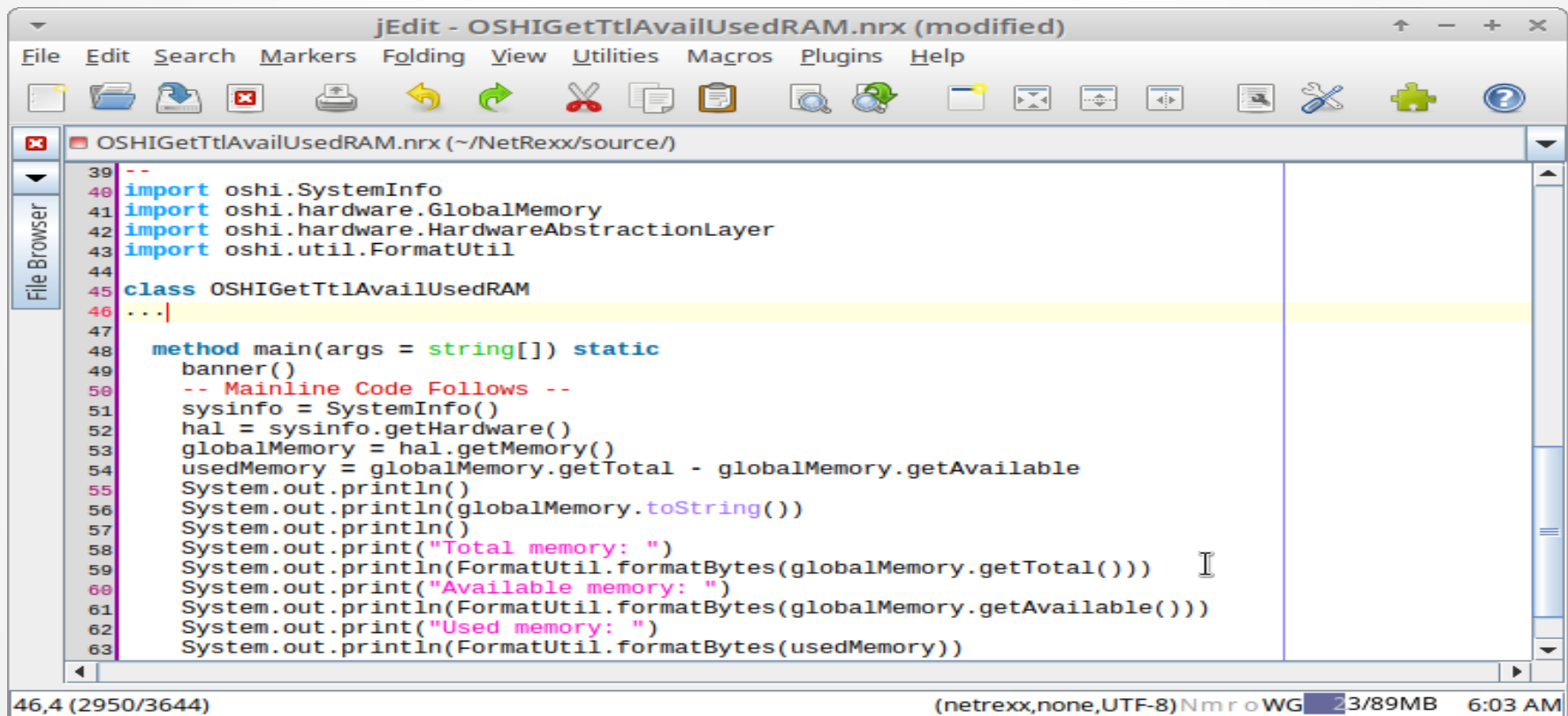
>>> End Of Program -- OSHIOSFISysDtIs.nrx <<<

C:\Storage\NetRexx\source>
```

NetRexx Sample Programs - 5.1

OSHIGetTtlAvailUsedRAM.nrx – Total, Available & Used RAM Info Using OSHI Java Library

NetRexx Code Snippet



```
jEdit - OSHIGetTtlAvailUsedRAM.nrx (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetTtlAvailUsedRAM.nrx (~/.NetRexx/source/)
39 --
40 import oshi.SystemInfo
41 import oshi.hardware.GlobalMemory
42 import oshi.hardware.HardwareAbstractionLayer
43 import oshi.util.FormatUtil
44
45 class OSHIGetTtlAvailUsedRAM
46 ...|
47
48 method main(args = string[]) static
49 banner()
50 -- Mainline Code Follows --
51 sysinfo = SystemInfo()
52 hal = sysinfo.getHardware()
53 globalMemory = hal.getMemory()
54 usedMemory = globalMemory.getTotal - globalMemory.getAvailable
55 System.out.println()
56 System.out.println(globalMemory.toString())
57 System.out.println()
58 System.out.print("Total memory: ")
59 System.out.println(FormatUtil.formatBytes(globalMemory.getTotal()))
60 System.out.print("Available memory: ")
61 System.out.println(FormatUtil.formatBytes(globalMemory.getAvailable()))
62 System.out.print("Used memory: ")
63 System.out.println(FormatUtil.formatBytes(usedMemory))
46,4 (2950/3644) (netrex,none,UTF-8)Nm r o WG 23/89MB 6:03 AM
```


NetRexx Sample Programs - 5.2

Sample Run of `OSHIGetTtlAvailUsedRAM.class` File (Debian Linux 10.10) :

```
tonyd@Expressum: ~/NetRexx/source
File Edit View Search Terminal Help
tonyd@Expressum:~/NetRexx/source$ java OSHIGetTtlAvailUsedRAM

OSHIGetTtlAvailUsedRAM.nrx
Version 1.0
Get Total, Available & Used RAM Info Using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 13, 2021
Date Revised: March 14, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Available: 4.3 GiB/5.7 GiB

Total memory: 5.7 GiB
Available memory: 4.3 GiB
Used memory: 1.4 GiB

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

tonyd@Expressum:~/NetRexx/source$ █
```

NetRexx Sample Programs - 5.3

Sample Run of `OSHIGetTtlAvailUsedRAM.class` File (openSUSE 15.3) :

```
Terminal - tonyd@Classier:~/NetRexx/source
File Edit View Terminal Tabs Help
tonyd@Classier:~/NetRexx/source> java OSHIGetTtlAvailUsedRAM

OSHIGetTtlAvailUsedRAM.nrx
Version 1.0
Get Total, Available & Used RAM Info Using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 13, 2021
Date Revised: March 14, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

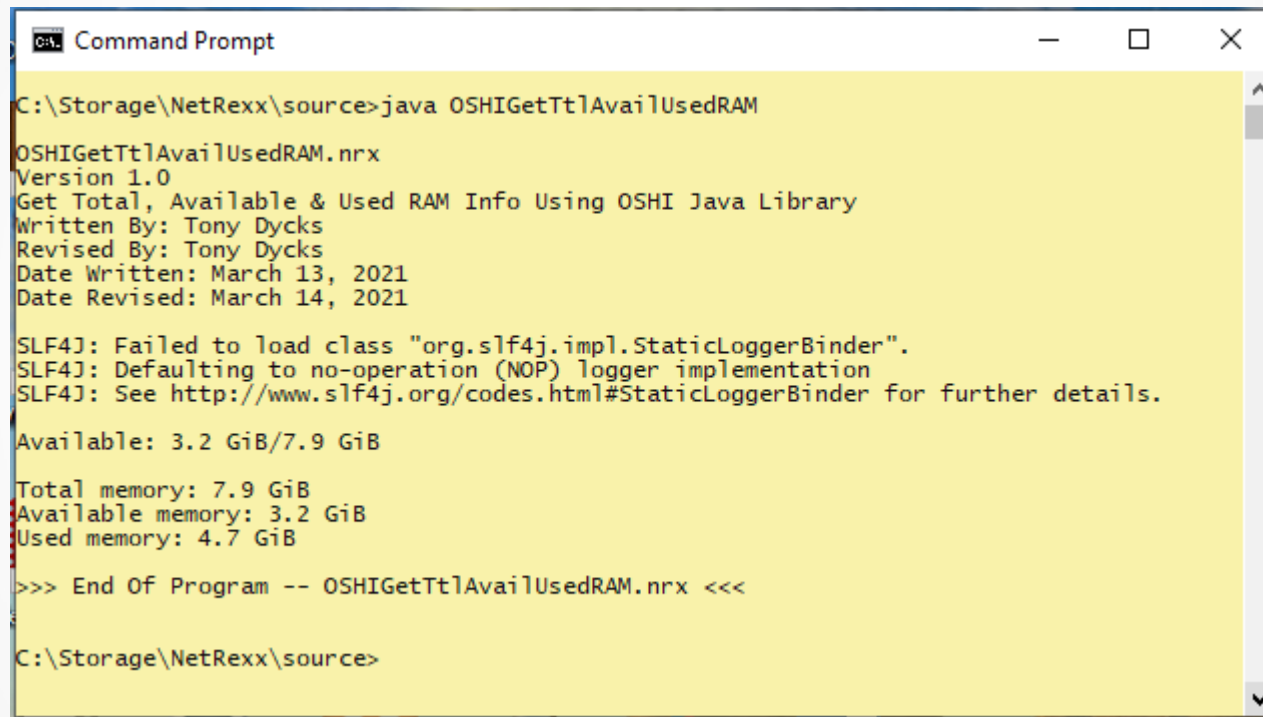
Available: 6.9 GiB/7.7 GiB

Total memory: 7.7 GiB
Available memory: 6.9 GiB
Used memory: 814.1 MiB

>>> End Of Program -- OSHIGetTtlAvailUsedRAM.nrx <<<
tonyd@Classier:~/NetRexx/source> █
```

NetRexx Sample Programs - 5.4

Sample Run of `OSHIGetTtlAvailUsedRAM.class` File (Windows 10 Professional) :



```
Command Prompt
C:\Storage\NetRexx\source>java OSHIGetTtlAvailUsedRAM
OSHIGetTtlAvailUsedRAM.nrx
Version 1.0
Get Total, Available & Used RAM Info Using OSHI Java Library
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: March 13, 2021
Date Revised: March 14, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Available: 3.2 GiB/7.9 GiB

Total memory: 7.9 GiB
Available memory: 3.2 GiB
Used memory: 4.7 GiB

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

C:\Storage\NetRexx\source>
```

BSF4ooRexx Sample Programs - 1

Program – Description

- 1) **OSHIGetCPUAll.rex** – Get All CPU Info as a String in Net Rexx using OSHI Java Library
- 2) **OSHIGetCPUIdInfo.rex** - Get CPU Id Info in Net Rexx using OSHI Java Library
- 3) **OSHIGetCPUNumbers.rex** – Get Number of Physical & Logical CPUs using OSHI Java Library
- 4) **OSHIOFISysDtls.rex** – Show Details of OS File System Using OSHI Java Library
- 5) **OSHIGetOS.rex** – Get OS Name, Family, Manufacturer and Bitness Using OSHI Java Library

BSF4ooRexx Run Syntax

Linux

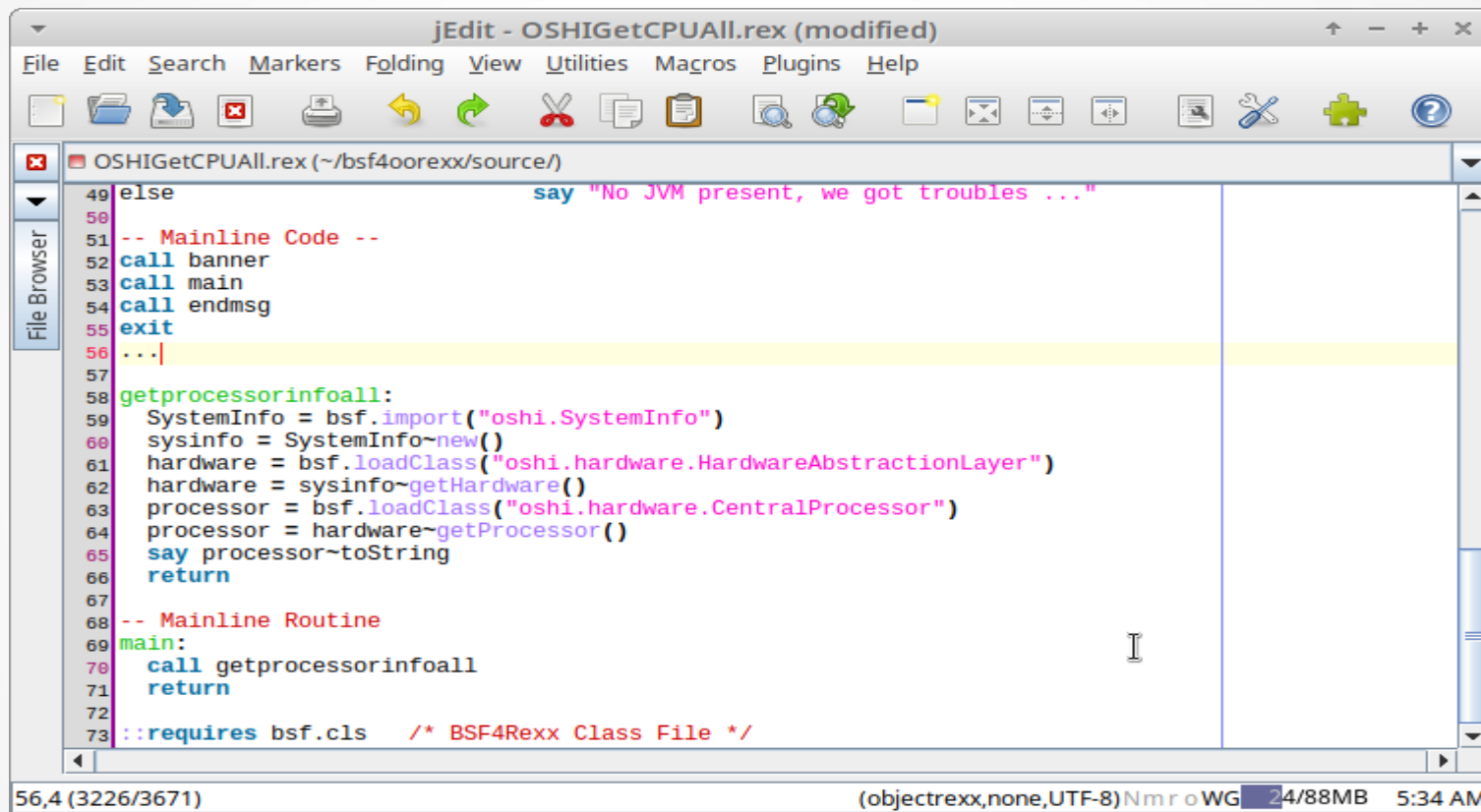
```
$sh ./rexxj2.sh <BSF4ooRexx-source-program>
```

Windows

```
>rexxj2.cmd <BSF4ooRexx-source-program>
```

BSF4ooRexx Sample Programs - 1.1

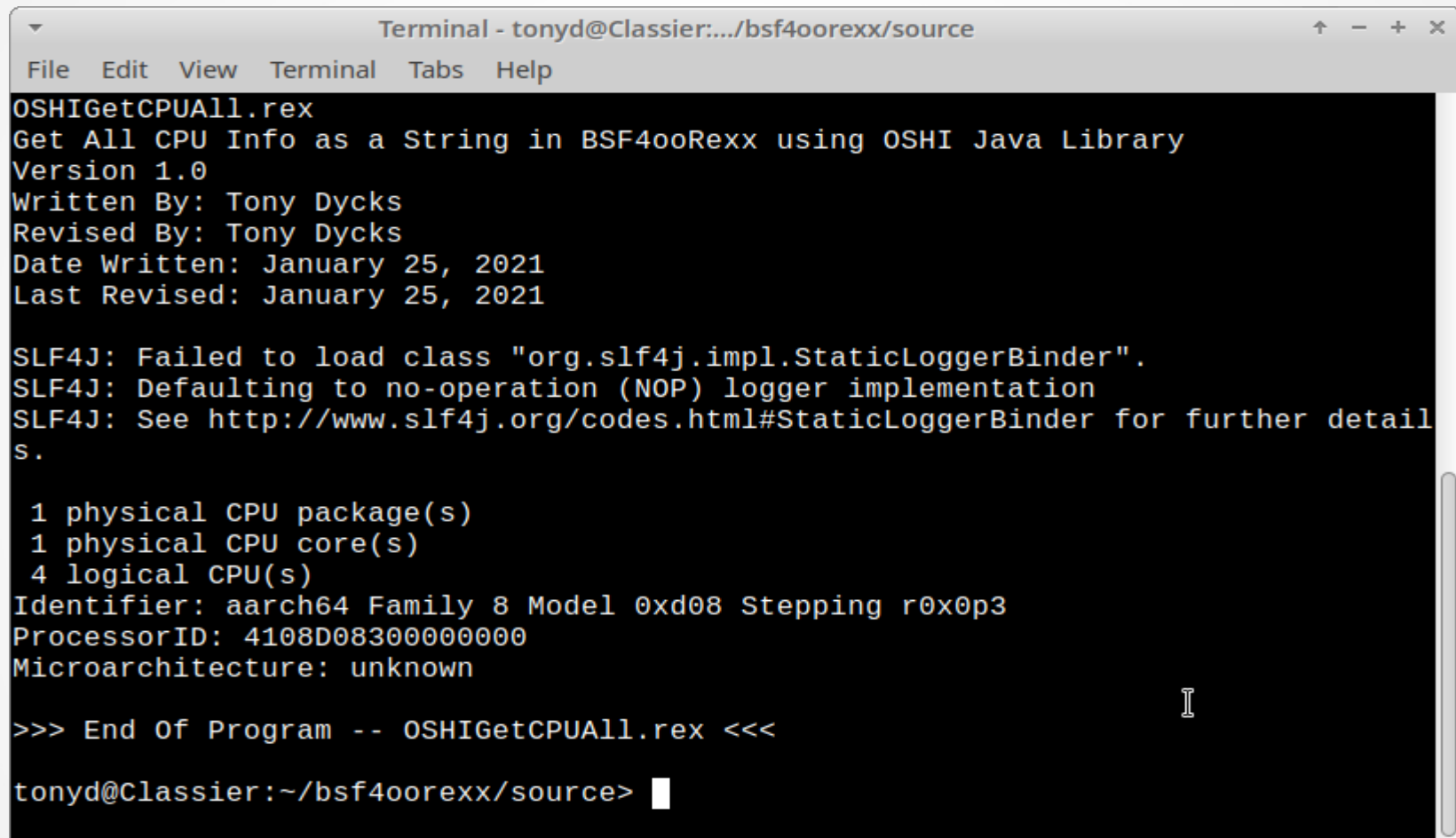
OSHIGetCPUAll.rex BSF4ooRexx Code Snippet:



```
jEdit - OSHIGetCPUAll.rex (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetCPUAll.rex (~/bsf4ooorex/source/)
49 else say "No JVM present, we got troubles ..."
50
51 -- Mainline Code --
52 call banner
53 call main
54 call endmsg
55 exit
56 ...
57
58 getprocessorinfoall:
59   SystemInfo = bsf.import("oshi.SystemInfo")
60   sysinfo = SystemInfo~new()
61   hardware = bsf.loadClass("oshi.hardware.HardwareAbstractionLayer")
62   hardware = sysinfo~getHardware()
63   processor = bsf.loadClass("oshi.hardware.CentralProcessor")
64   processor = hardware~getProcessor()
65   say processor~toString
66   return
67
68 -- Mainline Routine
69 main:
70   call getprocessorinfoall
71   return
72
73 ::requires bsf.cls /* BSF4Rexx Class File */
56,4 (3226/3671) (objectrexx,none,UTF-8)Nmr oWG 24/88MB 5:34 AM
```

BSF4ooRexx Sample Programs - 1.2

Sample Run of `OSHIGetCPUAll.rex` Program (openSUSE 15.3)



```
Terminal - tonyd@Classier:~/bsf4ooorex/source
File Edit View Terminal Tabs Help
OSHIGetCPUAll.rex
Get All CPU Info as a String in BSF4ooRexx using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

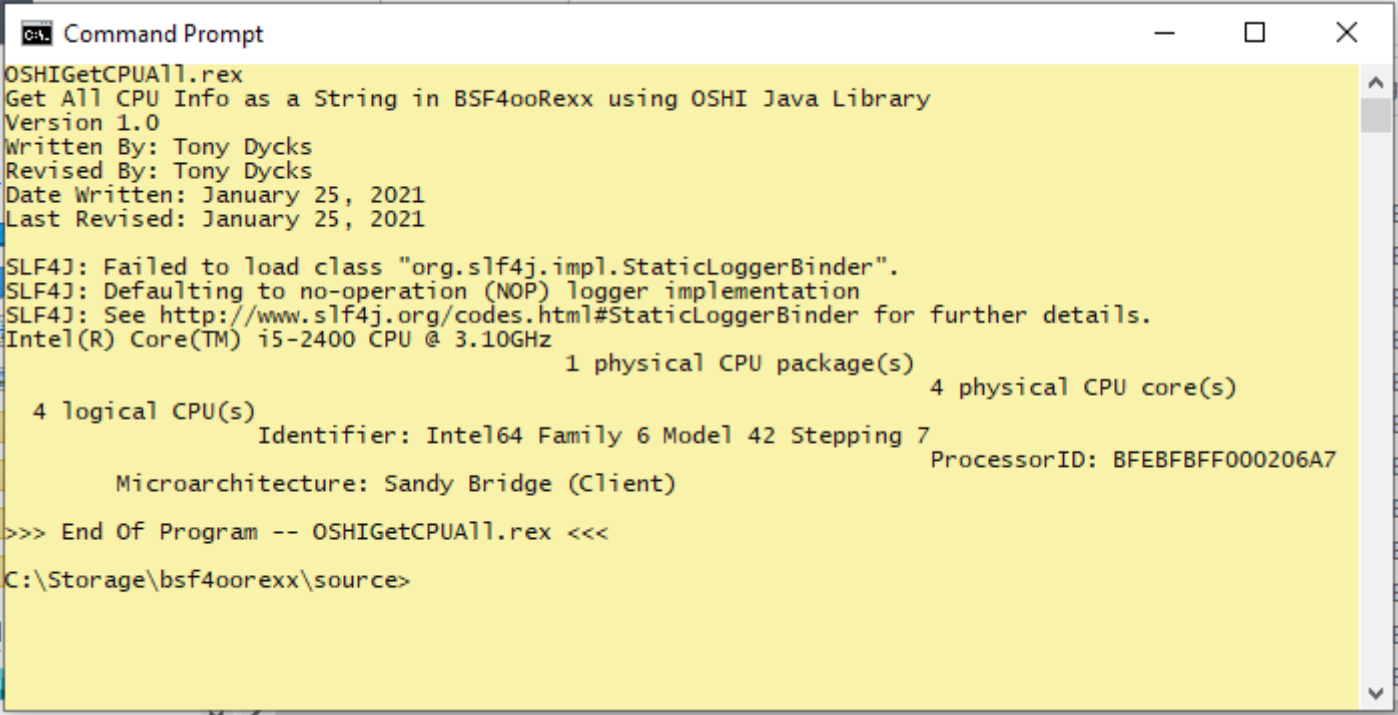
 1 physical CPU package(s)
 1 physical CPU core(s)
 4 logical CPU(s)
Identifier: aarch64 Family 8 Model 0xd08 Stepping r0x0p3
ProcessorID: 4108D08300000000
Microarchitecture: unknown

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

tonyd@Classier:~/bsf4ooorex/source>
```

BSF4ooRexx Sample Programs - 1.3

Sample Run of **OSHIGetCPUAll.rex** Program (Windows 10 Pro):



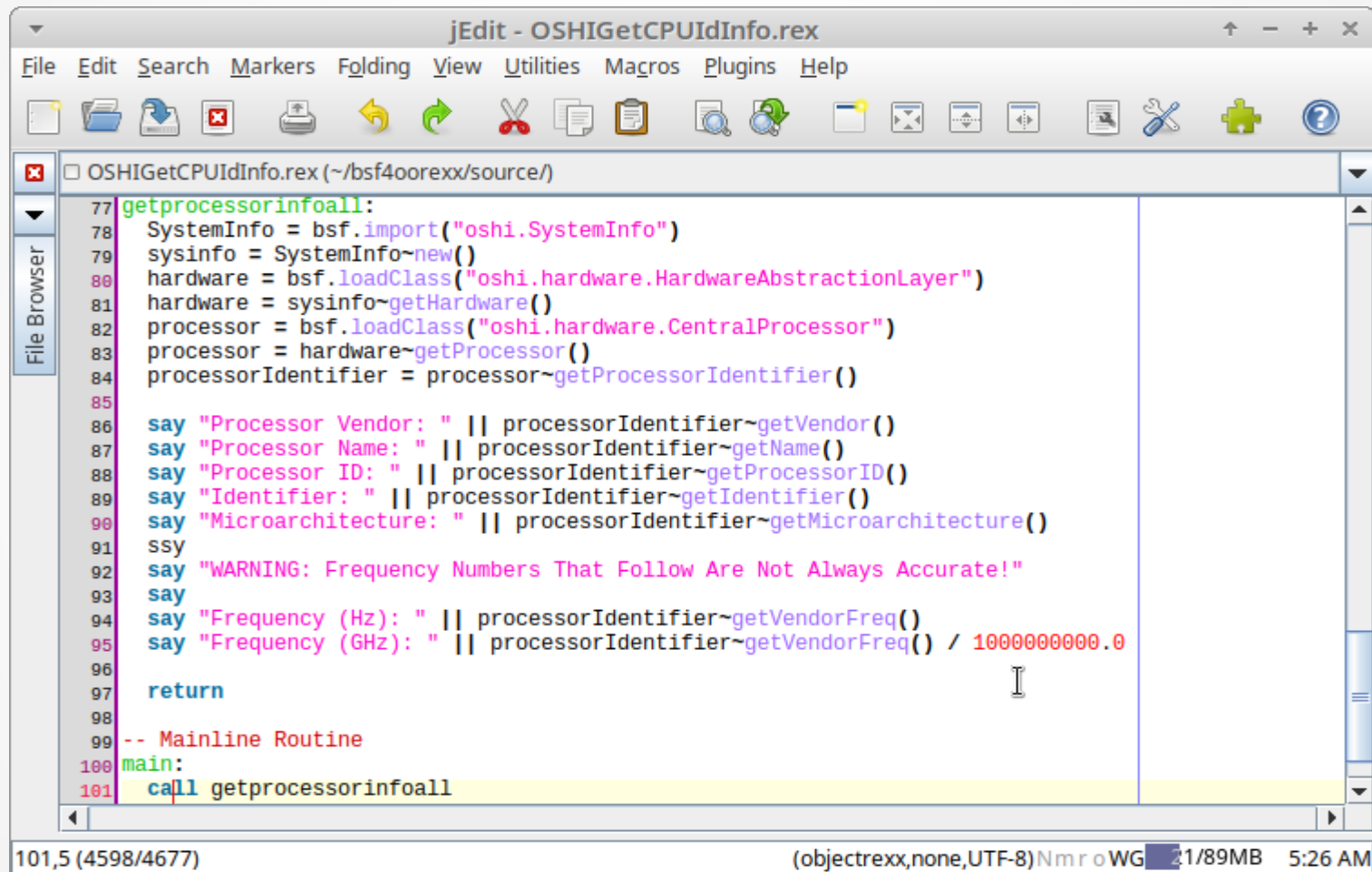
```
OSHIGetCPUAll.rex
Get All CPU Info as a String in BSF4ooRexx using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
           1 physical CPU package(s)
           4 physical CPU core(s)
4 logical CPU(s)
   Identifier: Intel64 Family 6 Model 42 Stepping 7
   Microarchitecture: Sandy Bridge (Client)
   ProcessorID: BFEBFBFF000206A7

>>> End Of Program -- OSHIGetCPUAll.rex <<<
C:\Storage\bsf4ooRexx\source>
```

BSF4ooRexx Sample Programs - 2.1

OSHIGetCPUIdInfo.rexx BSF4ooRexx Code Snippet:



```
jEdit - OSHIGetCPUIdInfo.rexx
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetCPUIdInfo.rexx (~/bsf4ooREXX/source/)
77 getprocessorinfoall:
78   SystemInfo = bsf.import("oshi.SystemInfo")
79   sysinfo = SystemInfo~new()
80   hardware = bsf.loadClass("oshi.hardware.HardwareAbstractionLayer")
81   hardware = sysinfo~getHardware()
82   processor = bsf.loadClass("oshi.hardware.CentralProcessor")
83   processor = hardware~getProcessor()
84   processorIdentifier = processor~getProcessorIdentifier()
85
86   say "Processor Vendor: " || processorIdentifier~getVendor()
87   say "Processor Name: " || processorIdentifier~getName()
88   say "Processor ID: " || processorIdentifier~getProcessorID()
89   say "Identifier: " || processorIdentifier~getIdentifier()
90   say "Microarchitecture: " || processorIdentifier~getMicroarchitecture()
91   ssy
92   say "WARNING: Frequency Numbers That Follow Are Not Always Accurate!"
93   say
94   say "Frequency (Hz): " || processorIdentifier~getVendorFreq()
95   say "Frequency (GHz): " || processorIdentifier~getVendorFreq() / 1000000000.0
96
97   return
98
99 -- Mainline Routine
100 main:
101   call getprocessorinfoall
101,5 (4598/4677) (objectrexx,none,UTF-8)Nmr oWG 21/89MB 5:26 AM
```


BSF4ooRexx Sample Programs - 2.2

Sample Run of **OSHIGetCPUIdInfo.rex** Program (openSUSE 15.3)

```
Terminal - tonyd@Classier:~/bsf4oorexx/source
File Edit View Terminal Tabs Help
Get CPU Id Info in BSF4ooRexx using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Processor Vendor: aarch64
Processor Name:
Processor ID: 4108D08300000000
Identifier: aarch64 Family 8 Model 0xd08 Stepping r0x0p3
Microarchitecture: unknown

WARNING: Frequency Numbers That Follow Are Not Always Accurate!

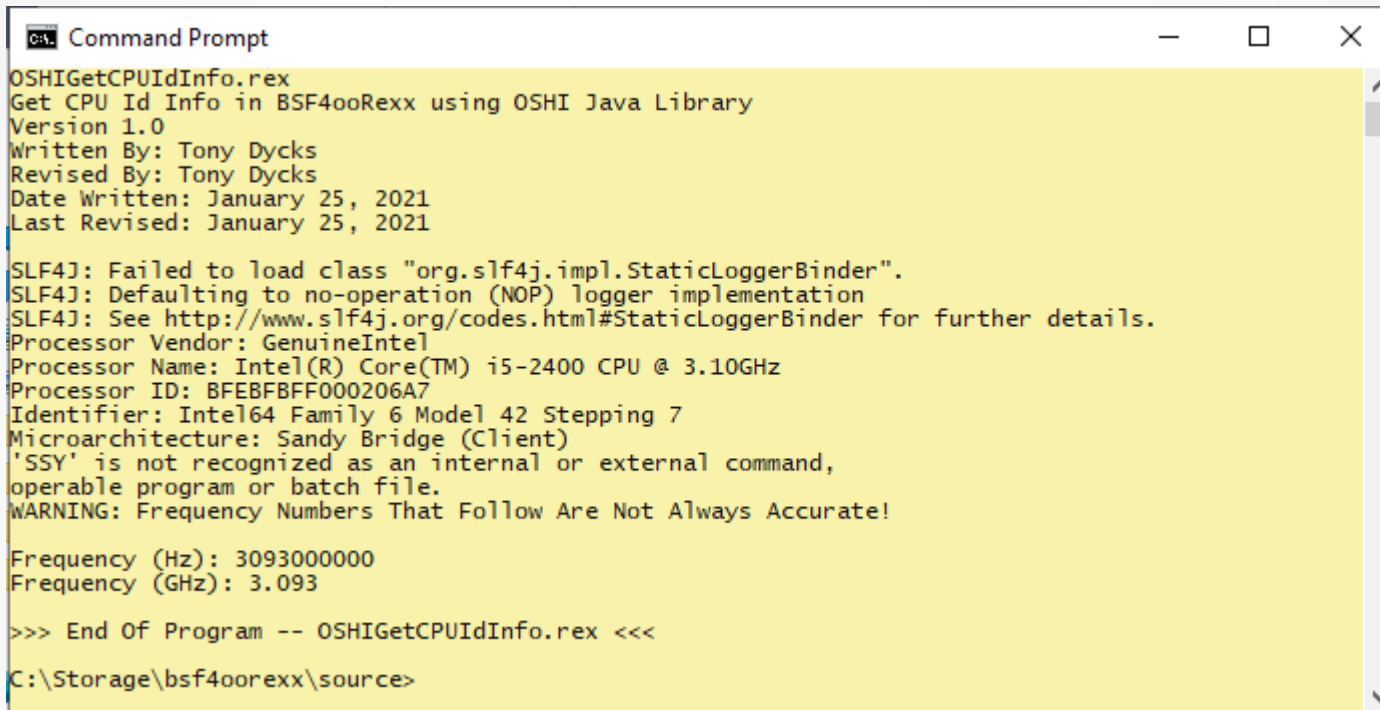
Frequency (Hz): -1
Frequency (GHZ): -0.000000001

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

tonyd@Classier:~/bsf4oorexx/source>
```

BSF4ooRexx Sample Programs - 2.3

Sample Run of **OSHIGetCPUIdInfo.rex** Program (Windows 10 Pro):



```
Command Prompt
OSHIGetCPUIdInfo.rex
Get CPU Id Info in BSF4ooRexx using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Processor Vendor: GenuineIntel
Processor Name: Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
Processor ID: BFEBFBFF000206A7
Identifier: Intel64 Family 6 Model 42 Stepping 7
Microarchitecture: Sandy Bridge (Client)
'SSY' is not recognized as an internal or external command,
operable program or batch file.
WARNING: Frequency Numbers That Follow Are Not Always Accurate!

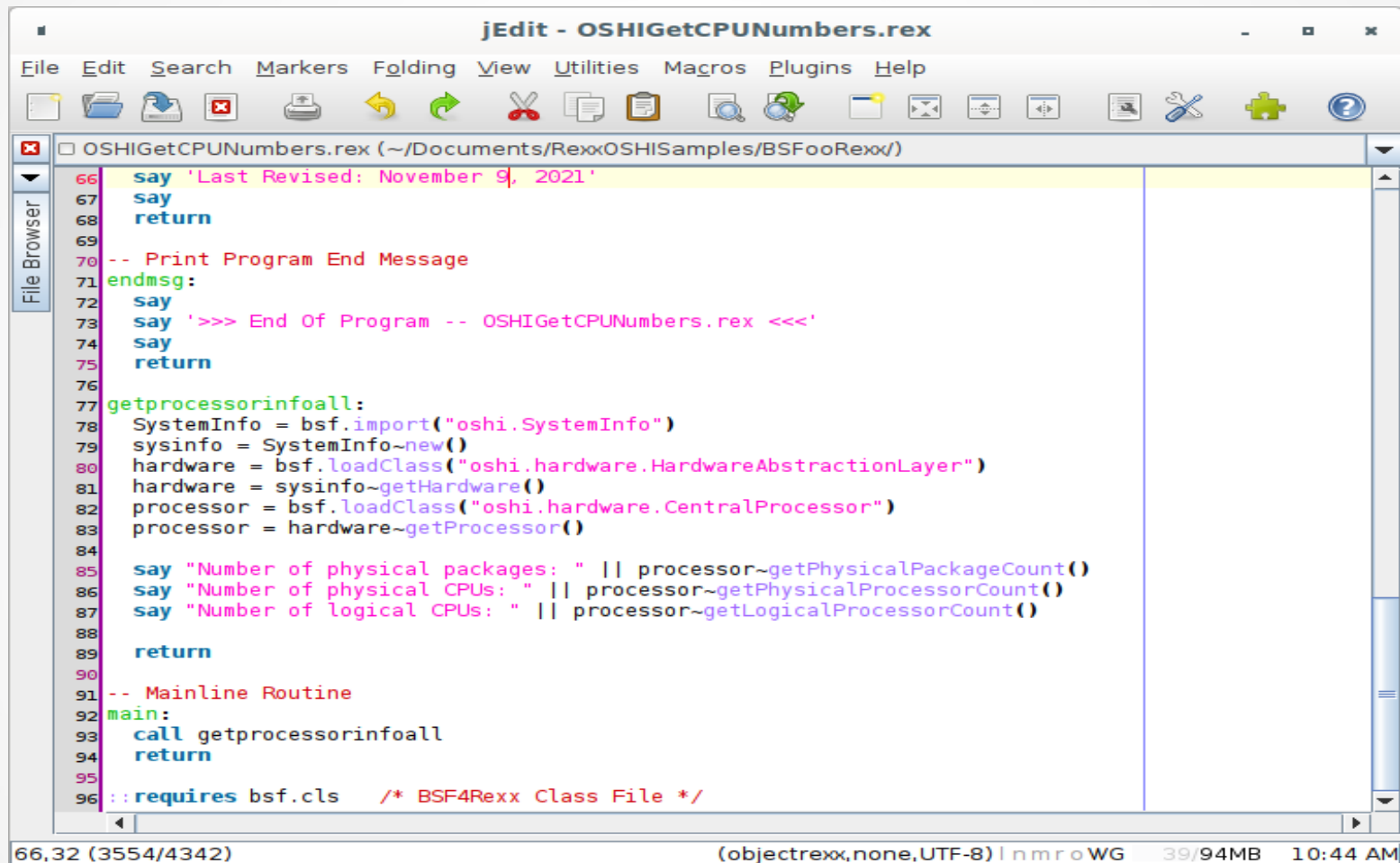
Frequency (Hz): 3093000000
Frequency (GHz): 3.093

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

C:\Storage\bsf4ooRexx\source>
```

BSF4ooRexx Sample Programs - 3.1

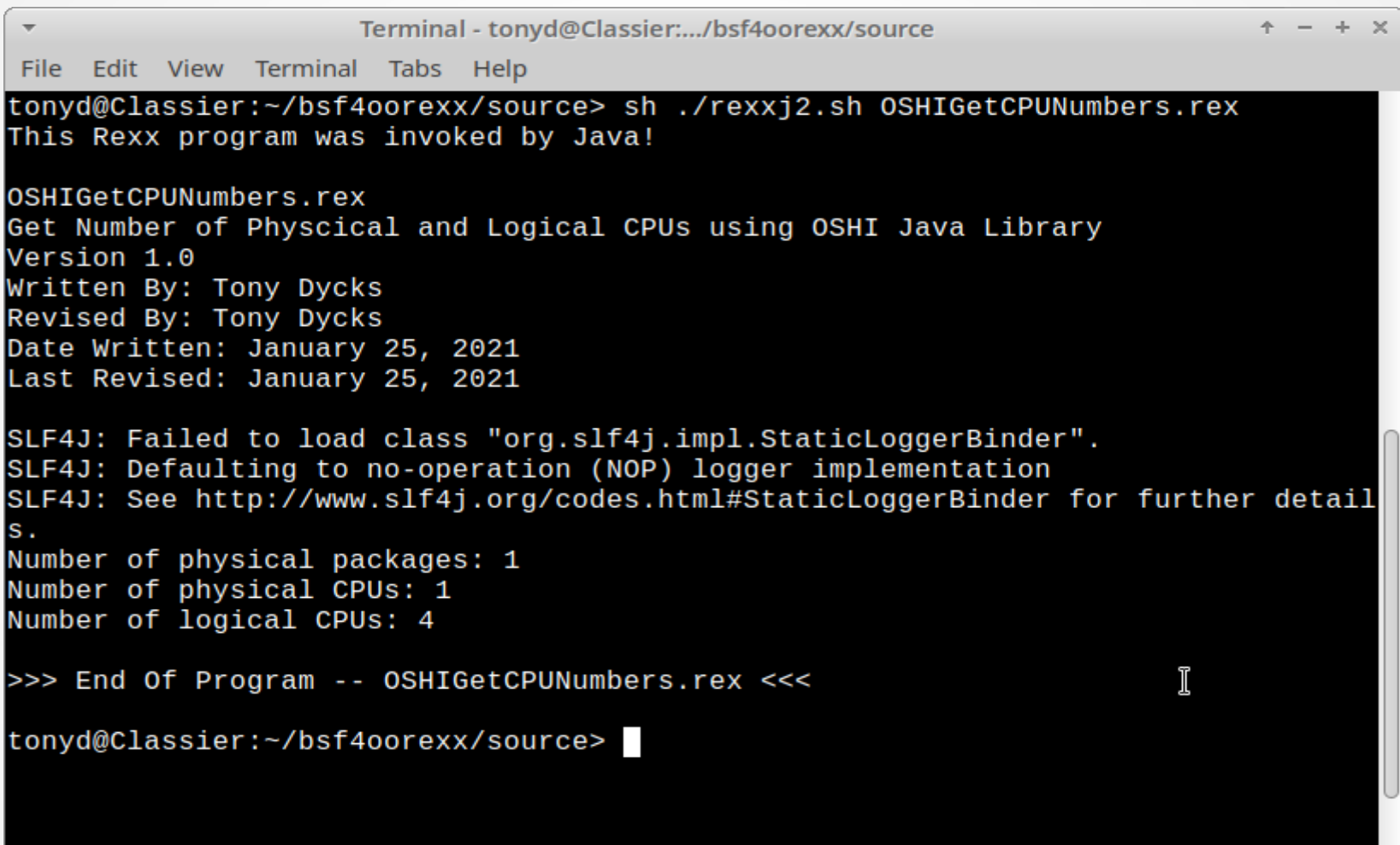
OSHIGetCPUNumbers.rex BSF4ooRexx Code Snippet (openSUSE 15.3)



```
jEdit - OSHIGetCPUNumbers.rex
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetCPUNumbers.rex (~/Documents/RexxOSHIExamples/BSF4ooRexx/)
66 say 'Last Revised: November 9, 2021'
67 say
68 return
69
70 -- Print Program End Message
71 endmsg:
72 say
73 say '>>> End Of Program -- OSHIGetCPUNumbers.rex <<<'
74 say
75 return
76
77 getprocessorinfoall:
78 SystemInfo = bsf.import("oshi.SystemInfo")
79 sysinfo = SystemInfo~new()
80 hardware = bsf.loadClass("oshi.hardware.HardwareAbstractionLayer")
81 hardware = sysinfo~getHardware()
82 processor = bsf.loadClass("oshi.hardware.CentralProcessor")
83 processor = hardware~getProcessor()
84
85 say "Number of physical packages: " || processor~getPhysicalPackageCount()
86 say "Number of physical CPUs: " || processor~getPhysicalProcessorCount()
87 say "Number of logical CPUs: " || processor~getLogicalProcessorCount()
88
89 return
90
91 -- Mainline Routine
92 main:
93 call getprocessorinfoall
94 return
95
96 ::requires bsf.cls /* BSF4Rexx Class File */
66,32 (3554/4342) (objectrexx,none,UTF-8) | n m r o W G 39/94MB 10:44 AM
```

BSF4ooRexx Sample Programs - 3.2

Sample Run of **OSHIGetCPUNumbers.rex** Program (openSUSE 15.3)



```
Terminal - tonyd@Classier:~/bsf4ooorexx/source
File Edit View Terminal Tabs Help
tonyd@Classier:~/bsf4ooorexx/source> sh ./rexxj2.sh OSHIGetCPUNumbers.rex
This Rexx program was invoked by Java!

OSHIGetCPUNumbers.rex
Get Number of Physical and Logical CPUs using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

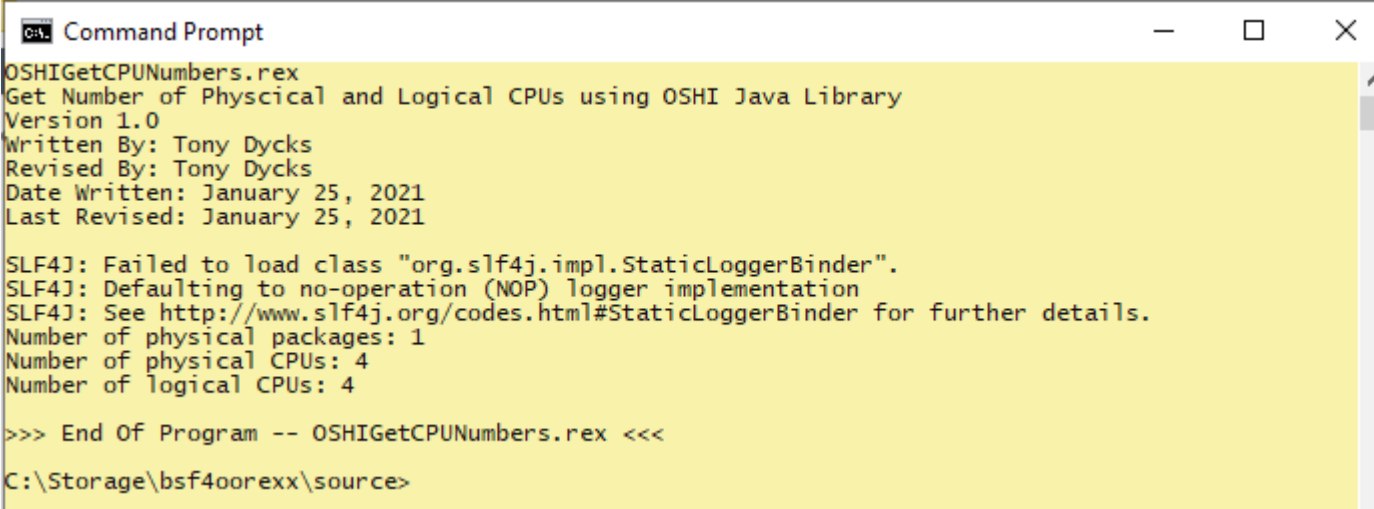
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Number of physical packages: 1
Number of physical CPUs: 1
Number of logical CPUs: 4

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

tonyd@Classier:~/bsf4ooorexx/source> █
```

BSF4ooRexx Sample Programs - 3.3

Sample Run of **OSHIGetCPUNumbers.rex** Program (Windows 10 Pro):



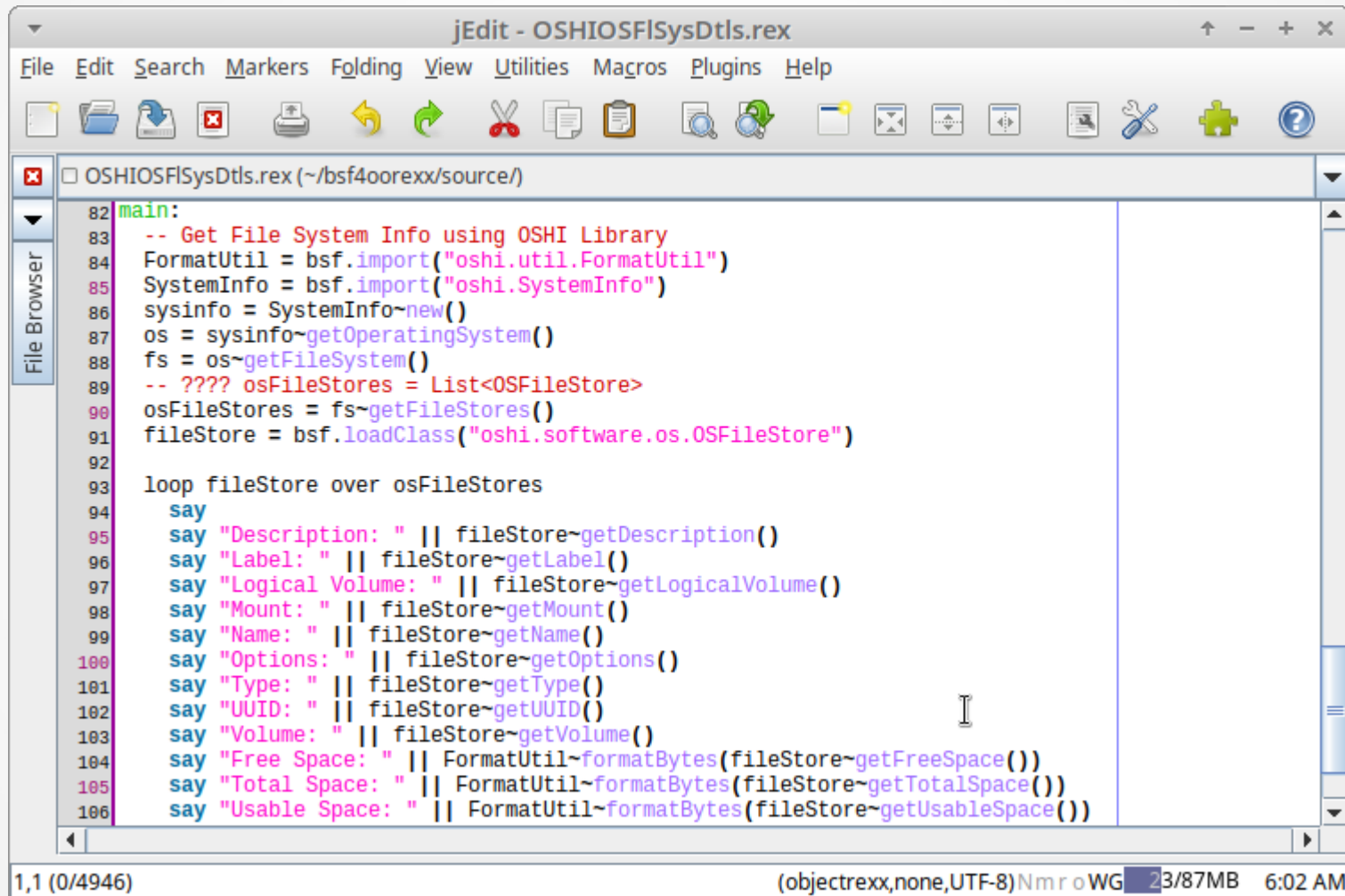
```
Command Prompt
OSHIGetCPUNumbers.rex
Get Number of Physical and Logical CPUs using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: January 25, 2021
Last Revised: January 25, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Number of physical packages: 1
Number of physical CPUs: 4
Number of logical CPUs: 4

>>> End Of Program -- OSHIGetCPUNumbers.rex <<<
C:\Storage\bsf4ooorex\source>
```

BSF4ooRexx Sample Programs - 4.1

OSHIOSFLSysDtIs.rex BSF4ooRexx Code Snippet (openSUSE 15.3)



```
jEdit - OSHIOSFLSysDtIs.rex
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIOSFLSysDtIs.rex (~/bsf4ooorex/source/)
82 main:
83 -- Get File System Info using OSHI Library
84 FormatUtil = bsf.import("oshi.util.FormatUtil")
85 SystemInfo = bsf.import("oshi.SystemInfo")
86 sysinfo = SystemInfo~new()
87 os = sysinfo~getOperatingSystem()
88 fs = os~getFileSystem()
89 -- ??? osFileStores = List<OSFileStore>
90 osFileStores = fs~getFileStores()
91 fileStore = bsf.loadClass("oshi.software.os.OSFileStore")
92
93 loop fileStore over osFileStores
94   say
95   say "Description: " || fileStore~getDescription()
96   say "Label: " || fileStore~getLabel()
97   say "Logical Volume: " || fileStore~getLogicalVolume()
98   say "Mount: " || fileStore~getMount()
99   say "Name: " || fileStore~getName()
100  say "Options: " || fileStore~getOptions()
101  say "Type: " || fileStore~getType()
102  say "UUID: " || fileStore~getUUID()
103  say "Volume: " || fileStore~getVolume()
104  say "Free Space: " || FormatUtil~formatBytes(fileStore~getFreeSpace())
105  say "Total Space: " || FormatUtil~formatBytes(fileStore~getTotalSpace())
106  say "Usable Space: " || FormatUtil~formatBytes(fileStore~getUsableSpace())
```

1,1 (0/4946) (objectrexx,none,UTF-8)Nmr oWG 23/87MB 6:02 AM

BSF4ooRexx Sample Programs - 4.2

Sample Run of **OSHIOSFISysDtIs.rex** Program (openSUSE 15.3)

```
Terminal - tonyd@Classier:~/bsf4oorexx/source
File Edit View Terminal Tabs Help
Label: ROOT
Logical Volume:
Mount: /
Name: /
Options: rw,noatime,nobarrier
Type: ext4
UUID: f0d5f2c6-5788-49b1-9317-d9b7fb75bcb2
Volume: /dev/mmcblk1p3
Free Space: 41.9 GiB
Total Space: 58.2 GiB
Usable Space: 38.9 GiB

Description: Local Disk
Label: EFI
Logical Volume:
Mount: /boot/efi
Name: /dev/mmcblk1p1
Options: rw,relatime,mask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
Type: vfat
UUID: 94d9-0da4
Volume: /dev/mmcblk1p1
Free Space: 55.6 MiB
Total Space: 63.9 MiB
Usable Space: 55.6 MiB
```

BSF4ooRexx Sample Programs - 4.3

Sample Run of **OSHIOSFISysDtIs.rex** Program (Windows 10 Pro)

```
Select Command Prompt
Description: Fixed drive
Label: Windows
Logical Volume:
Mount: C:\
Name: Local Fixed Disk (C:)
Options: rw,reparse,sparse,trans,journaled,quota,casess,oids,casepn,efs,streams,unicode,acls,fcom
Type: NTFS
UUID: 9cf0c889-0000-0000-0000-f01500000000
Volume: \\?\Volume{9cf0c889-0000-0000-0000-f01500000000}\
Free Space: 205.7 GiB
Total Space: 464.9 GiB
Usable Space: 205.7 GiB

Description: CD-ROM
Label:
Logical Volume:
Mount: D:\
Name: CD-ROM Disc (D:)
Options: rw
Type:
UUID: e99820b3-37cd-11ea-a12b-806e6f6e6963
Volume: \\?\Volume{e99820b3-37cd-11ea-a12b-806e6f6e6963}\
Free Space: 0 bytes
Total Space: 0 bytes
Usable Space: 0 bytes

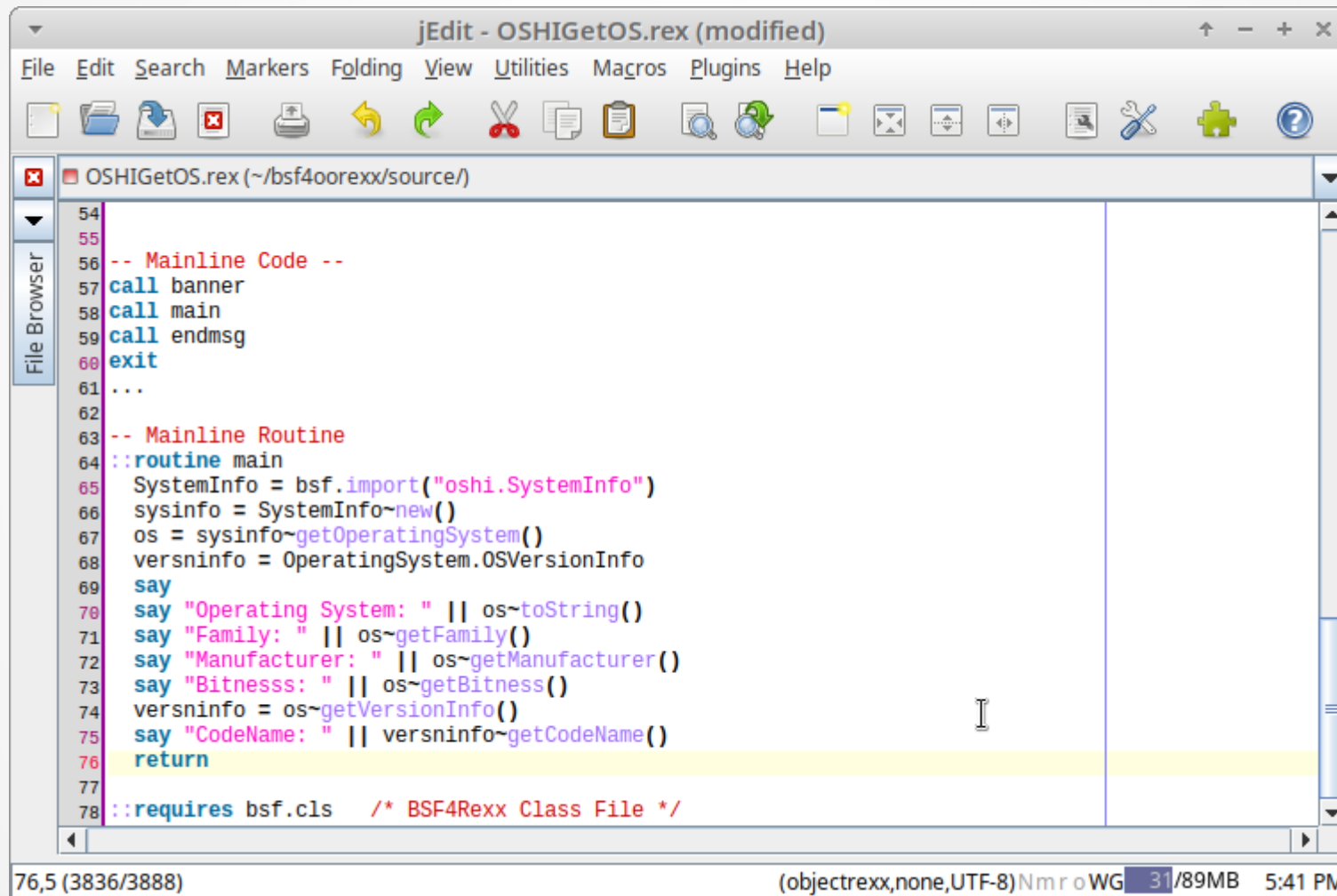
Description: CD-ROM
Label:
Logical Volume:
Mount: E:\
Name: CD-ROM Disc (E:)
Options: rw
Type:
UUID: be475d3f-4555-11ea-a12e-e83935410934
Volume: \\?\Volume{be475d3f-4555-11ea-a12e-e83935410934}\
Free Space: 0 bytes
Total Space: 0 bytes
Usable Space: 0 bytes

Description: Removable drive
Label:
Logical Volume:
Mount: F:\
Name: Removable Disk (F:)
Options: rw,casepn,efs,unicode
Type: FAT32
UUID: 7a6272cd-46a3-11ea-a149-806e6f6e6963
Volume: \\?\Volume{7a6272cd-46a3-11ea-a149-806e6f6e6963}\
Free Space: 11.8 GiB
Total Space: 57.3 GiB
Usable Space: 11.8 GiB

>>> End Of Program -- OSHIOSFISysDtIs.rex <<<
C:\Storage\bsf4ooorex\source>
```


BSF4ooRexx Sample Programs - 5.1

OSHIGetOS.rex BSF4ooRexx Code Snippet (openSUSE 15.3)



```
jEdit - OSHIGetOS.rex (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
OSHIGetOS.rex (~/.bsf4ooRexx/source/)
54
55
56 -- Mainline Code --
57 call banner
58 call main
59 call endmsg
60 exit
61 ...
62
63 -- Mainline Routine
64 ::routine main
65 SystemInfo = bsf.import("oshi.SystemInfo")
66 sysinfo = SystemInfo~new()
67 os = sysinfo~getOperatingSystem()
68 versninfo = OperatingSystem.OSVersionInfo
69 say
70 say "Operating System: " || os~toString()
71 say "Family: " || os~getFamily()
72 say "Manufacturer: " || os~getManufacturer()
73 say "Bitness: " || os~getBitness()
74 versninfo = os~getVersionInfo()
75 say "CodeName: " || versninfo~getCodeName()
76 return
77
78 ::requires bsf.cls /* BSF4Rexx Class File */
76,5 (3836/3888) (objectrexx,none,UTF-8)Nmr oWG 31/89MB 5:41 PM
```

BSF4ooRexx Sample Programs - 5.2

Sample Run of **OSHIGetOS.rex** Program (openSUSE 15.3)

```
Terminal - tonyd@Classier:~/bsf4oorexx/source
File Edit View Terminal Tabs Help
OSHIGetOS.rex
Get OS Name, Family, Manufacturer and Bitness Using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: August 2, 2021
Last Revised: August 2, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

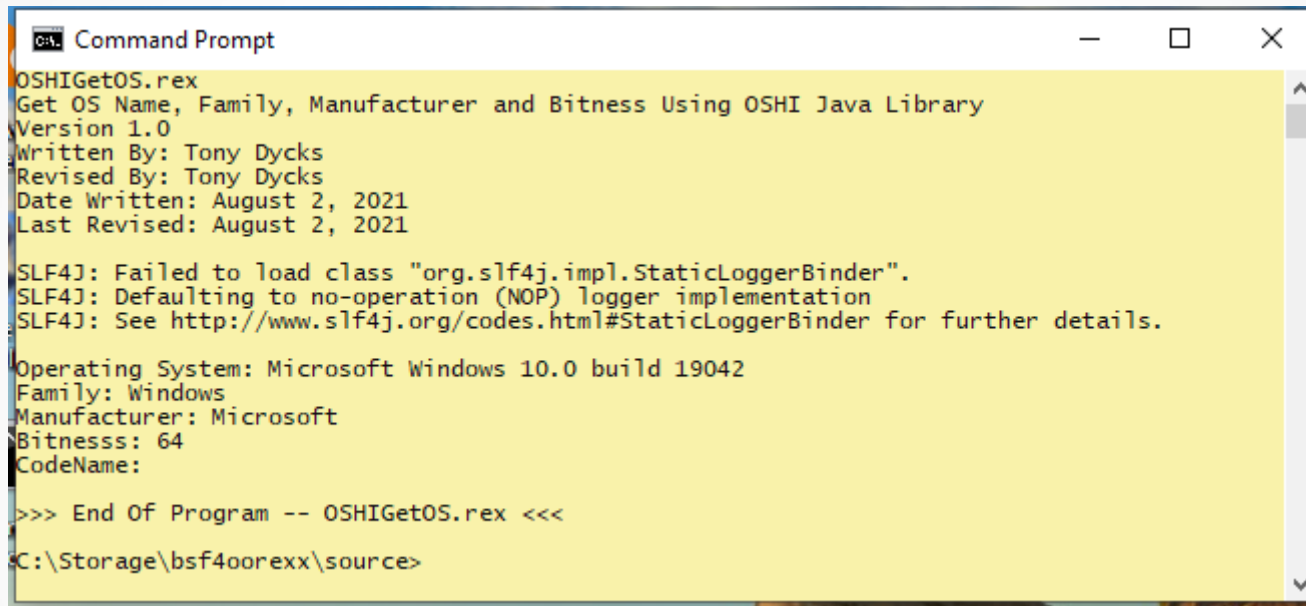
Operating System: GNU/Linux openSUSE Leap 15.3 (unknown) build 5.3.18-59.16-default
Family: openSUSE Leap
Manufacturer: GNU/Linux
Bitness: 64
CodeName: unknown

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

tonyd@Classier:~/bsf4oorexx/source
```

BSF4ooRexx Sample Programs - 5.3

Sample Run of **OSHIGetOS.rex** Program (Windows 10 Pro)



```
Command Prompt
OSHIGetOS.rex
Get OS Name, Family, Manufacturer and Bitness Using OSHI Java Library
Version 1.0
Written By: Tony Dycks
Revised By: Tony Dycks
Date Written: August 2, 2021
Last Revised: August 2, 2021

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Operating System: Microsoft Windows 10.0 build 19042
Family: Windows
Manufacturer: Microsoft
Bitness: 64
CodeName:

>>> End Of Program -- OSHIGetOS.rex <<<
C:\Storage\bsf4ooorex\source>
```

Summary of Test Findings

- In OSHI CPU Id Info Programs, Frequency of Processor was a Negative False Value on the Raspberry Pi OS Linux Distros
- Intel CPUs Displayed Correct Frequencies
- Micro Architecture Info was Unavailable for the Raspberry Pi 4
- Intel Processors displayed Micro Architecture Info
- Physical and Logical CPU Info was Accurate for Most Cases
- OS Code Name Property Was Unavailable in Windows 10
- File System Details Were Also Accurate and Informative
- Rexx Programs Could Be Run on All Platforms Without Any Code Modifications to Cater to the OS or Hardware
- Jna Jar Worked For Most Linux Distros; Jna-platform Jar Was Required for Windows 10
- Zip Archive of Sample Rexx Programs Will Be Included on the Rexx LA Presentation Site
- JNA v5.8 Currently Does Not Support FreeBSD 13.0 on the RPi4

List of Java Technologies Used



OPERATING SYSTEM &
HARDWARE INFORMATION

Technology	Version	Function	URL
oshi-core-5.8.jar	5.8	OS & HW Java API	https://github.com/oshi/oshi
jna-5.6.jar	5.6	Java Native Access API	https://github.com/java-native-access/jna
slf4j-1.7.25.jar	1.7.25	System Logging Facility	http://www.slf4j.org/
JEdit Text Edit	5.1-5.6	Java Based Editor with Rexx Syntax Awareness	http://www.jedit.org
Java Open JDK	8 (1.8)	Java Development Kit	https://openjdk.java.net/

List of Rexx Technologies



Technology	Version	Function	URL
NetRexx	3.09GA	NetRexx Compiler & Runtime	http://www.netrex.org
Open Object Rexx	5.0 Beta	Rexx Interpreter	https://www.oorex.org/download.html
BSF4ooRexx	V641 (July 2021 Release)	Bean Scripting Facility for ooRexx	https://sourceforge.net/projects/bsf4oorex
Rexx Language Association	N/A	ooRexx & NetRexx Documentation & Presentations	https://www.rexxla.org
Oracle Java SE 8	8 (1.8)	Java SE Development	Refer to Slide #9 for Download URLs

Zip File of Rexx Programs / OSHI Java URLs



OPERATING SYSTEM &
HARDWARE INFORMATION

Description	File / URL
Simple Solution – OSHI Java Tutorial	https://simplsolution.dev/java-oshi-tutorial/
Zip Archive of Sample Rexx Programs	RexxOSHISamples.zip
Simple Solution – Get File System Information in Java using OSHI library	https://simplsolution.dev/java-get-file-system-information-oshi-library/
Simple Solution – Get CPU Information in Java using OSHI library	https://simplsolution.dev/java-get-cpu-information-oshi-library/
Simple Solution – Get Operating System Information in Java using OSHI library	https://simplsolution.dev/java-get-operating-system-information-oshi-library/

End of Presentation



OPERATING SYSTEM &
HARDWARE INFORMATION

Questions? Comments?

“ “

” ”