Display JIT-compiled code by a Java process on Windows

How to see the JIT-compiled assembly code on Windows 10

The first steps are described here https://stackoverflow.com/questions/1503479/how-to-see-jit-compiled-code-in-jvm

and here : https://dropzone.nfshost.com/hsdis/

Files were mirrored and steps saved just in case

- **Install Cygwin**. At the Select Packages screen, add the following packages (expand the All/Devel category, Search for the package's name, and double-click Skip in its row of the table):
    - gcc-core
    - make
    - mingw64-i686-gcc-core (only needed for hsdis-i386.dll)
    - mingw64-x86_64-gcc-core (only needed for hsdis-amd64.dll)

    Run the Cygwin (or Cygwin64) Terminal.
    This can be done using the Desktop or Start Menu icon created by the installer, and will create your Cygwin home directory (C:\cygwin\home\username or C:\cygwin64\home\username, by default).
- **Download GNU binutils** to your Cygwin home directory, and extract it with:

binutils-2.35.1.tar.xz

```
> tar xvf binutils-2.35.1.tar.xz
```

- **Download hsdis** to your Cygwin home directory, and extract it with:

jdk15u-jdk-15.0.1+9.tar.bz2

```
> tar xvf jdk15u-jdk-15.0.1+9.tar.bz2
```

- **Change to the hsdis directory:**

```
cd jdk15u-jdk-15.0.1+9/src/utils/hsdis
```

- **Build hsdis-amd64.dll:**

```
make OS=Linux MINGW=x86_64-w64-mingw32 BINUTILS=~/binutils-2.35.1
```

or hsdis-i386.dll:

```
make OS=Linux MINGW=i686-w64-mingw32 BINUTILS=~/binutils-2.35.1 DLDFLAGS='-shared -static-libgcc'
```

Once built, the DLL can be installed by **copying it from build/Linux-amd64 or build/Linux-i586 to your JRE's bin directory (wherever exists `java.dll` (use Windows search)) .**

The following VM options should display the native asm code :

`-XX:+UnlockDiagnosticVMOptions -XX:+PrintAssembly`

Filtering the compiled methods can be done with :

`-XX:+UnlockDiagnosticVMOptions -XX:CompileCommand=print,*FilteredClass.filterMethod`

Bonus tip:

`-XX:PrintAssemblyOptions=options`

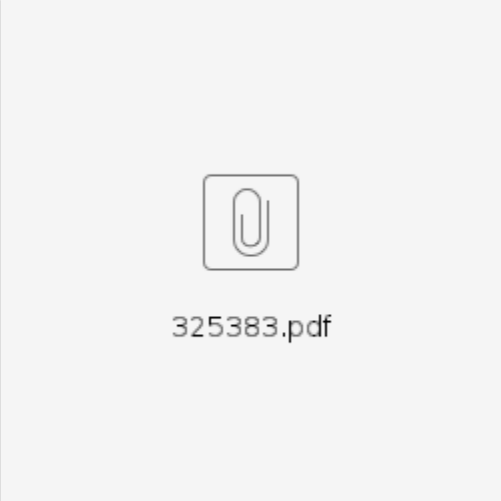
can be used to pass the same options accepted by

`objdump's -M/--disassembler-options.`

For example, if you prefer Intel syntax to AT&T, specify

`-XX:PrintAssemblyOptions=intel-mnemonic.`

Reference document for instructions with Intel syntax



Here's the built DLLs as of 16/Nov/2021 (Win 10):

hsdis-amd64.dll

hsdis-i386.dll