Installation of Catalina

Tuesday, September 02, 2014

This document describes how to install Catalina, a next-generation of MDELite and prototype for an IDE plug-in. Catalina is presently a windows-only tool.

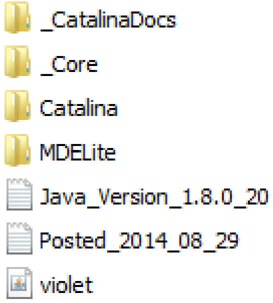
Download

Point your browser to: http://www.cs.utexas.edu/users/schwartz/MDELite/index.html and click on the Catalina download link. You will download a 7zip file; the 7zip tool is free.

Catalina has a dependency Cygwin, meaning that you should install Cygwin. Cygwin is free and provides a bash-like environment for Windows. Just install vanilla (nothing special) Cygwin.

Unzip

Unzipping the download yields a single directory, Categories, whose content looks like:



_CatalinaDocs
_Core
Catalina
MDELite
Java_Version_1.8.0_20
Posted_2014_08_29
violet

- **_CatalinaDocs** is where you'll find this document, and others.
- **_Core** is the framework of Catalina.
- **Catalina** is the framework extension that allows you to build other MDE applications. Catalina was used to bootstrap itself.
- **MDELite** is the Catalina re-implementation of the MDELite tool. MDE was built as a Catalina application.
- **Java_Version** – indicates the version of java that produced this version of Catalina.
- **Posted** – the day that this version was released
- **Violet** is an old version of the violet UML diagram tool. Eventually, this will be replaced with the current version.

Warning: make sure that you use the indicated version of Java. I've noticed that it really DOES make a difference.

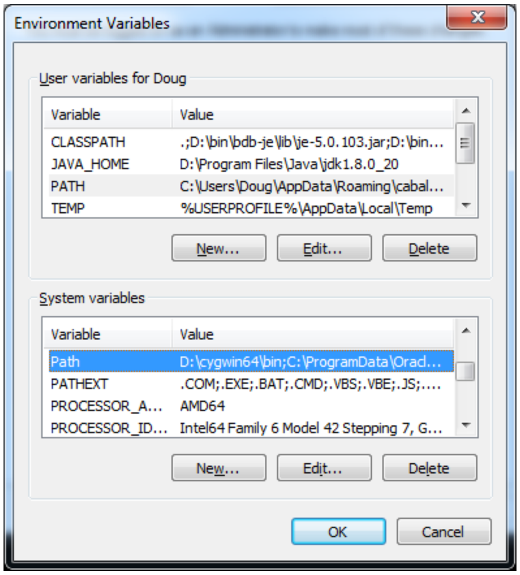
Environment Variables

Catalina tools rely on the following environment variables:

- **CATLIB** – the path to the Categories directory.

- **GVIM_EXE** – the GVIM executable; download GVIM from here.
- **JAVA_EXE** – the java executable.
- **SWIPL_EXE** – the SWI Prolog console; download SWI Prolog here.
- **SWIPL_WIN_EXE** – the SWI Prolog windows application; download SWI Prolog here.
- **BROWSER_EXE** – an internet browser (e.g. “chrome.exe”, “iexplorer.exe”)
- **CLASSPATH** – should include all .jar files in the Categories (**CATLIB**) directory
- **PATH** – should include paths to all 4 of the above executables.

Here is how I make these settings in Windows – I run a batch file, which spawns a command-line window with these settings. You need to set the environment variables to values that represent your environment. Place a shortcut to this batch file on your desktop. Clicking it will launch a DOS window with all Catalina settings ready to go.



Variable	Value
CLASSPATH	.;D:\bin\bdb-je\lib\je-5.0.103.jar;D:\bin...
JAVA_HOME	D:\Program Files\Java\jdk1.8.0_20
PATH	C:\Users\Doug\AppData\Roaming\cabal...
TEMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
Path	D:\cygwin64\bin;C:\ProgramData\Oracl...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64
PROCESSOR_ID...	Intel64 Family 6 Model 42 Stepping 7, G...

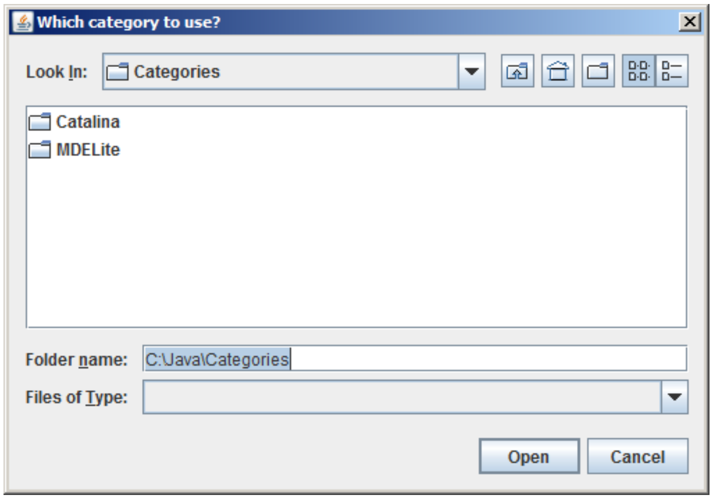
> **Note**: If you don't already have Cygwin configured, you will need to make sure that the path to Cygwin's /bin/ folder is the first item in their system-wide %Path% variable, as shown in the Environment Variables screenshot, and then restart your computer. The batch file (mentioned above) makes these settings automatically when a command-line window is spawned.

CheckEnvironment

Launch a DOS window and run `CatCore.CheckEnvironment`. This program verifies if the Catalina tools can invoke your programs.

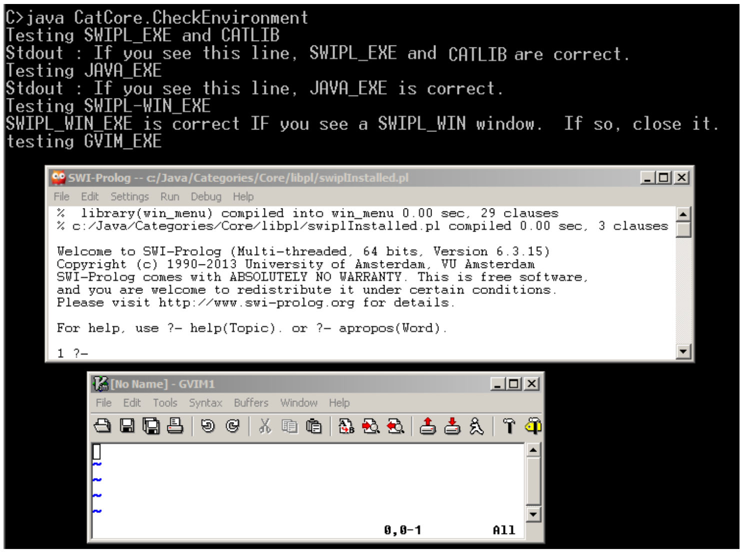
```
> java CatCore.CheckEnvironment
```

Doing so, you’ll be asked what category you will be using. In Catalina, each category is a program that can invoke any number of executables. Doing so yields the dialog below. The directories listed are Catalina applications. Click on **Catalina**.



Folder name: C:\Java\Categories

If everything is correct, you'll see some chatter and a SWI-Prolog and GVim window pop-up, not necessarily in the nice visual alignment below:



```
C>java CatCore.CheckEnvironment
Testing SWIPL_EXE and CATLIB
Stdout : If you see this line, SWIPL_EXE and CATLIB are correct.
Testing JAVA_EXE
Stdout : If you see this line, JAVA_EXE is correct.
Testing SWIPL-WIN_EXE
SWIPL_WIN_EXE is correct IF you see a SWIPL_WIN window.  If so, close it.
testing GVIM_EXE
```

SWI-Prolog -- c:/Java/Categories/Core/libpl/swiplInstalled.pl
File Edit Settings Run Debug Help

```
%  library(win_menu) compiled into win_menu 0.00 sec, 29 clauses
% c:/Java/Categories/Core/libpl/swiplInstalled.pl compiled 0.00 sec, 3 clauses

Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.3.15)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
```

`CheckEnvironment` exercises the setting of every environment variable. For example, the first lines in the above Figure reports the testing of the SWIPL_EXE and CATLIB variables. The output confirms that the settings are correct. Subsequently, JAVA_EXE is tested, along with SWIPL_WIN_EXE, GVIM_EXE, and BROWSER_EXE. Follow the output instructions. If no errors are reported, you're good to go.

Of course, if you don't get the above output, recheck your settings and re-run `CheckEnvironnent`.

> ***Note:*** *here is a batch file that invokes checkEnvironment after Catalina environment variables have been set.*
>
> ***Note****: In the usual Java tradition of "write once, get different results on different platforms", the version of Java that you makes a difference.* `CheckEnvironment` *insists that some executables don't exist, but they plainly do. On a 32-bit platform,* `CheckEnvironment` *under JDK1.7.0_17 will report errors in paths, whereas no*

errors are reported in JDK1.8.0_20. A general rule: if you experience problems with path settings, provide the FULL path to an executable (rather than an abbreviation (ex: "java.exe")) instead of assuming the System environment variable "Path" takes care of everything.

***Note**: it is possible that Catalina requires the use of Cygwin. If so, Cygwin is free and you can download it from here. Let me know if this is the case.*

Violet

I recommend that you create a DOS batch file, too, for invoking violet. Here's the incantation (in batch file format – which says fork a window for the Violet Java UMLeditor):

```
@start java com.horstmann.violet.UMLEditor
```

Place a shortcut to this batch file on your desktop for easy access.

Updates

If you find any deviation from the above, or have questions, or have recommendations, please contact me (Don Batory).