

# VisualVM

# Agenda

- Using VisualVM
  - > Monitoring & troubleshooting
  - > VisualVM features
  - > Extending VisualVM with plugins
- Developing plugins for VisualVM
  - > API overview
  - > Do it yourself! - how to write a VisualVM plugin

# What's VisualVM?

- A new Integrated and Extensible Troubleshooting Tool for the Java Platform
- Integrates existing JDK Management, Monitoring and Troubleshooting tools and adds support for lightweight CPU and Memory profiling
- Extensible through VisualVM Plugins Center
- Production and development time tool
- Audience: developers, administrators, performance and sustaining engineers, etc.
- <https://visualvm.dev.java.net>

# VisualVM System Requirement

- VisualVM itself must be run with JDK 6.
- It can display information, however, on any Java application that is running on JDK 1.4.2 or higher

# Using VisualVM

# Demo

# VisualVM features

- can monitor JVM from 1.4.2 up
- 4 technologies
  - > Jvmstat, Attach API, JMX, Serviceability Agent (SA)
- VisualVM uses all of them
- Local and remote application
- CPU and Memory Profiling
  - > stripped down version of NetBeans Profiler
- File viewer
  - > binary HPROF heap dump
  - > NetBeans Profiler CPU and Memory snapshots
  - > Thread dump, Core dumps
- Application snapshots
  - > offline analysis

# VisualVM feature matrix

Localhost	1.4.2	5.0	6	7
Jvmstat	X	X	X	X
JMX		X <sup>1</sup>	X	X
Attach			X	X
SA <sup>2</sup>		X	X	X

- 1) With `-Dcom.sun.management.jmxremote`
- 2) Linux and Solaris only

# VisualVM feature matrix

Remote	1.4.2	5.0	6	7
Jvmstat <sup>1</sup>	X	X	X	X
JMX <sup>2</sup>		X	X	X
Attach				
SA				

- 1) jstatd must be running on remote host
- 2) with `-Dcom.sun.management.jmxremote.port=portno`

# VisualVM feature matrix

Functions	Jvmstat	JMX	Attach	SA
detect app	X	X		
monitoring	X	X		
threads		X		
profiler			X	
threadump		X	X	X
heapdump		X <sup>1</sup>	X	
invoke GC		X		
OOME		X <sup>1</sup>	X	
sys prop		X	X	X
JVM args	X	X		X
main class	X			X

# Demo

# Extending VisualVM with plugins

- Standard
  - > MBeans
  - > JConsole
- Beta
  - > GlassFish
- Experimental
  - > JavaDB, BTrace, GChisto, VisualGC
- 3<sup>rd</sup> party
  - > Terracotta

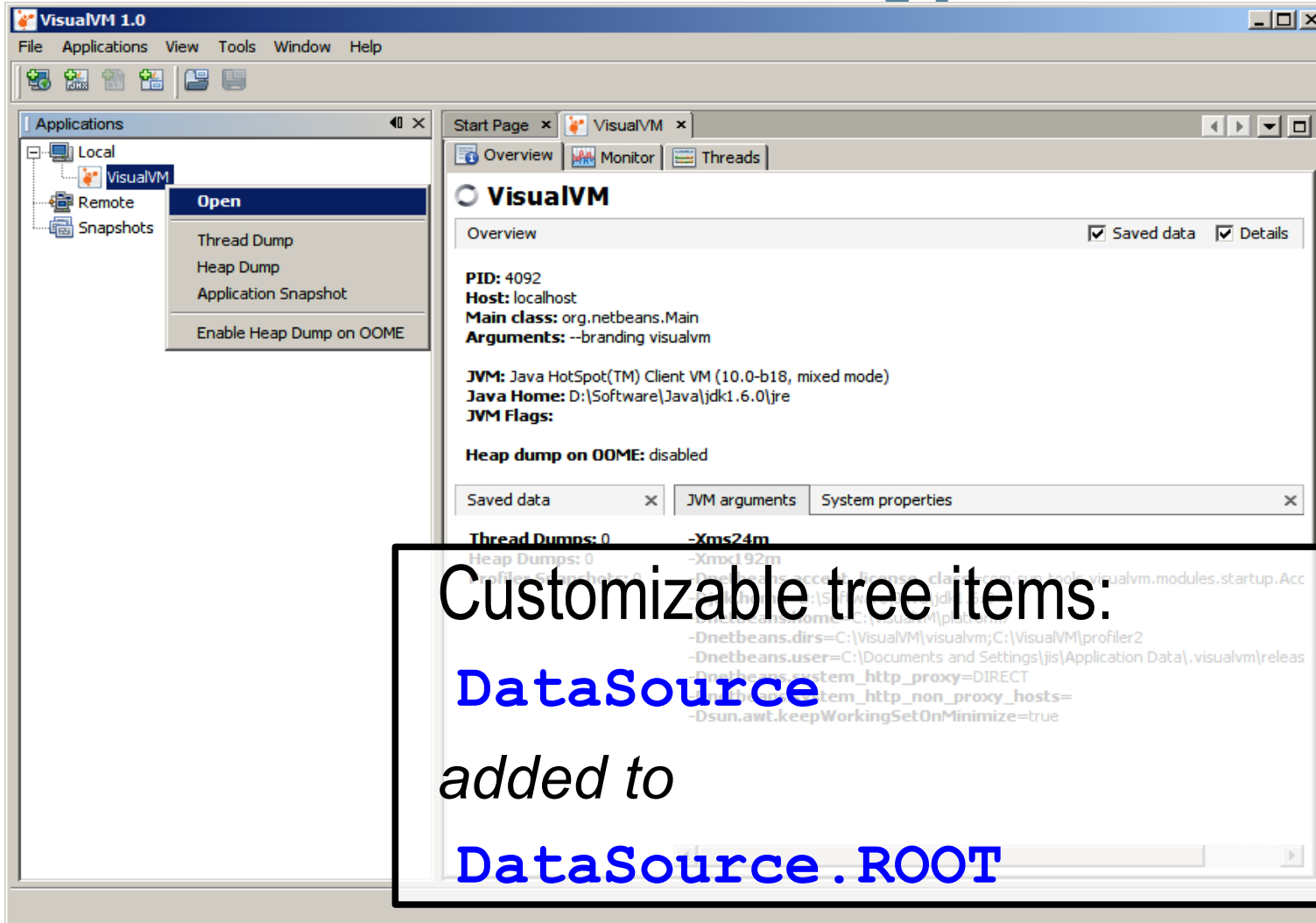
# Demo

# Developing Plug-in's for VisualVM

# API overview

- DataSource
  - > Application
  - > Host
  - > ThreadDump, HeapDump, ...
- Models for DataSource
  - > attaching functionality to DataSource
- Tools API
  - > Jvmstat
  - > JMX
  - > Attach
  - > SA
- JVM abstraction

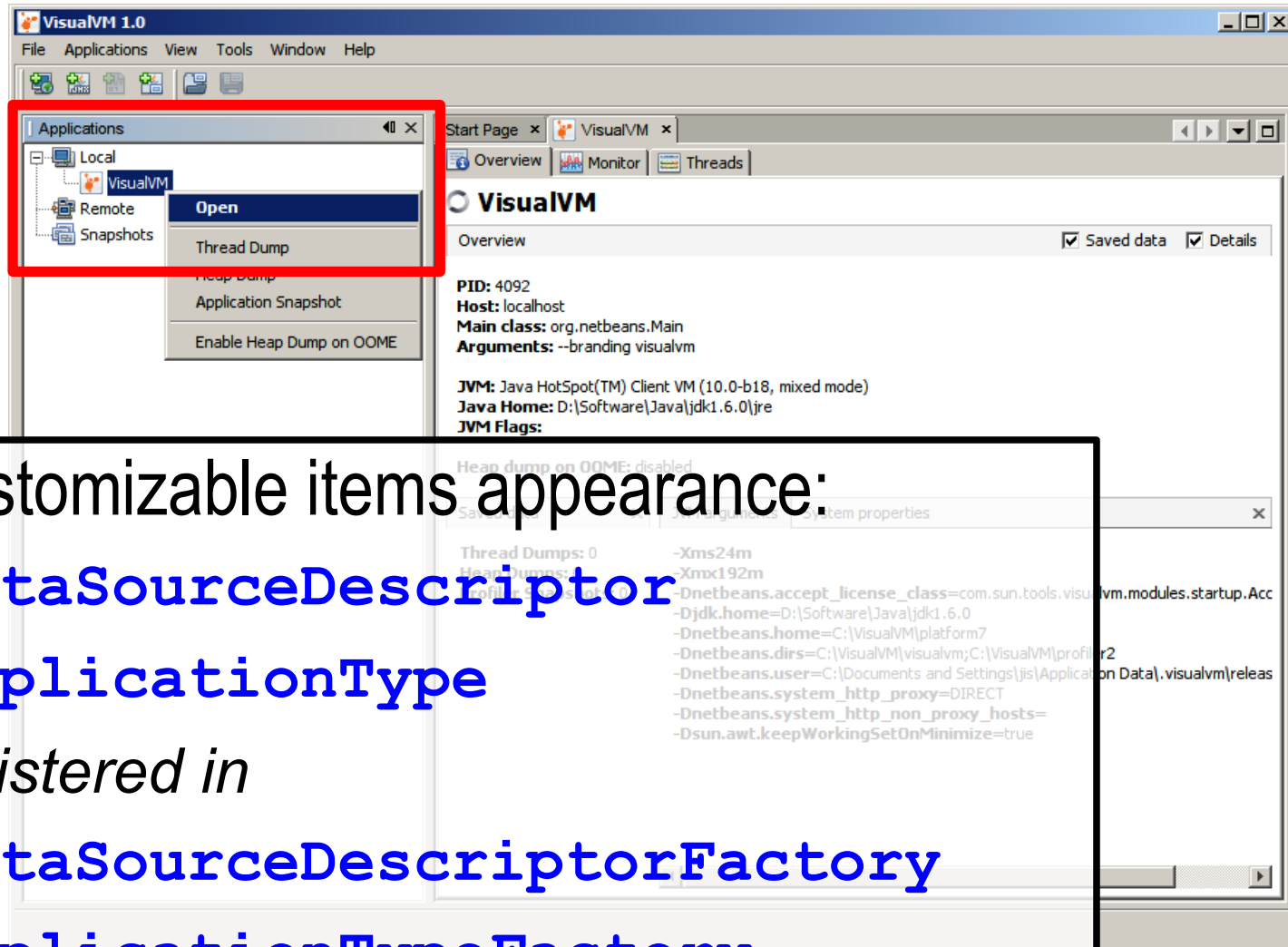
# VisualVM 1.0 API Entrypoints



The screenshot shows the VisualVM 1.0 application window. On the left, the 'Applications' tree is expanded to show 'VisualVM'. A context menu is open over 'VisualVM', listing several actions: 'Open', 'Thread Dump', 'Heap Dump', 'Application Snapshot', and 'Enable Heap Dump on OOME'. The main pane shows the 'Overview' tab for the selected application, displaying details such as PID (4092), Host (localhost), Main class (org.netbeans.Main), and JVM arguments. A text box is overlaid on the bottom right of the screenshot, containing the text: 'Customizable tree items: DataSource added to DataSource.ROOT'. The text 'DataSource' and 'DataSource.ROOT' are highlighted in blue.

Customizable tree items:  
**DataSource**  
*added to*  
**DataSource.ROOT**

# VisualVM 1.0 API Entrypoints



Customizable items appearance:

**DataSourceDescriptor**

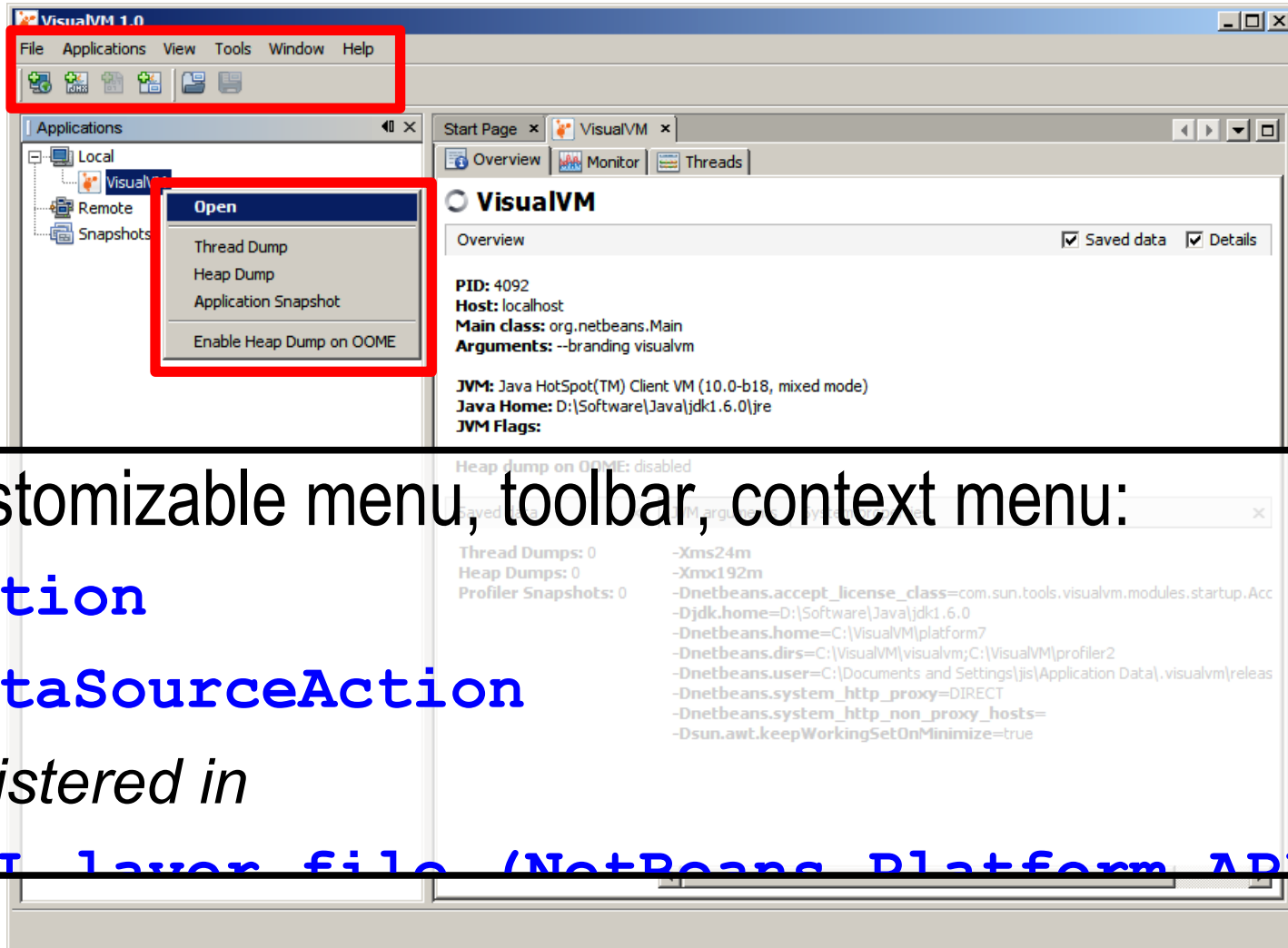
**ApplicationType**

*registered in*

**DataSourceDescriptorFactory**

**ApplicationTypeFactory**

# VisualVM 1.0 API Entrypoints



Customizable menu, toolbar, context menu:

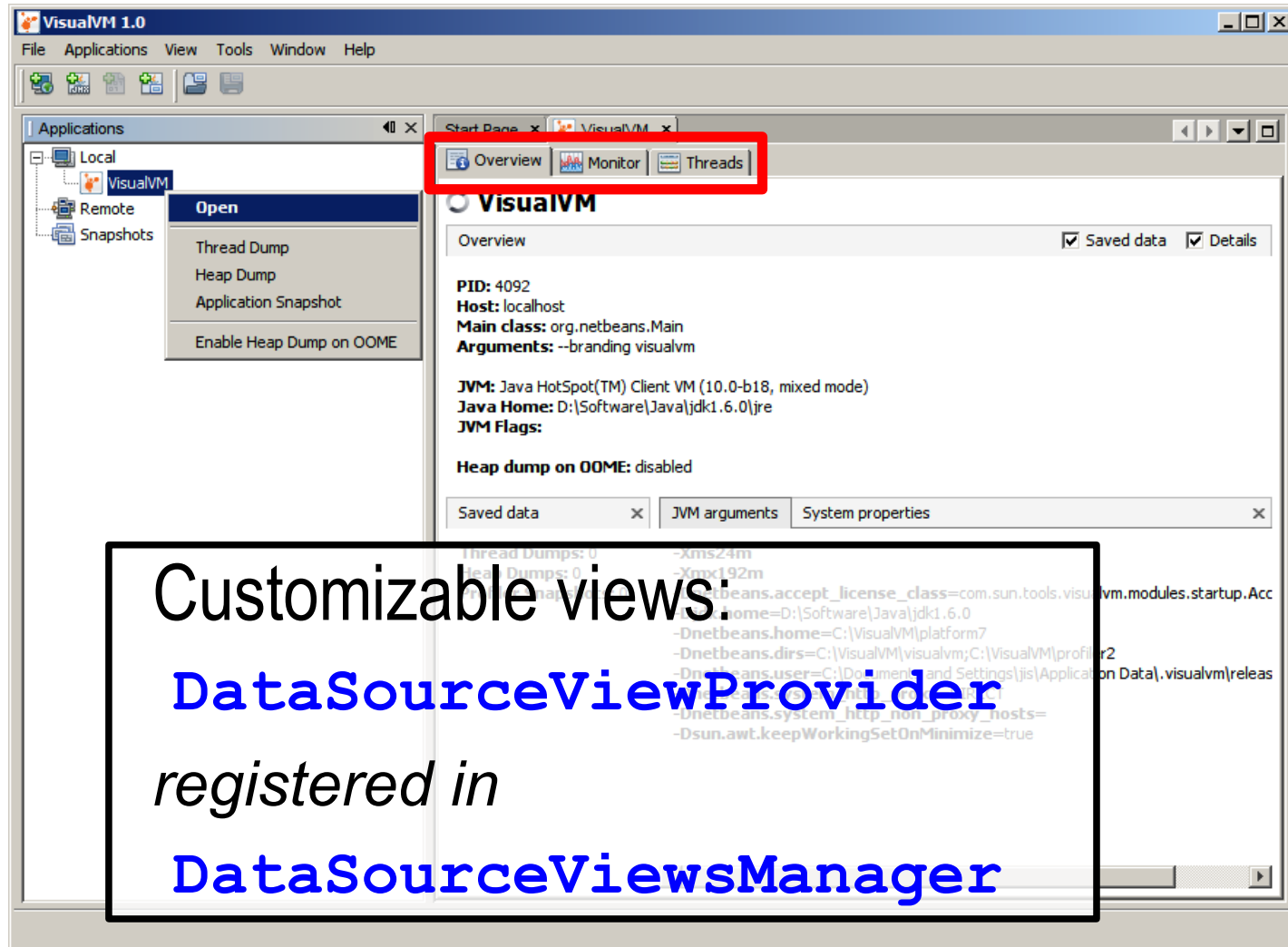
**Action**

**DataSourceAction**

*registered in*

**YMI layer file (NetBeans Platform API)**

# VisualVM 1.0 API Entrypoints



The screenshot shows the VisualVM 1.0 application window. The 'Overview' tab is selected in the top navigation bar, which is highlighted with a red box. The main content area displays the following information:

- PID:** 4092
- Host:** localhost
- Main class:** org.netbeans.Main
- Arguments:** --branding visualvm
- JVM:** Java HotSpot(TM) Client VM (10.0-b18, mixed mode)
- Java Home:** D:\Software\Java\jdk1.6.0\jre
- JVM Flags:**
- Heap dump on OOME:** disabled

At the bottom of the window, there are tabs for 'Saved data', 'JVM arguments', and 'System properties'. The 'JVM arguments' tab is active, showing the following arguments:

```

-Xms24m
-Xmx192m
-Dnetbeans.accept_license_class=com.sun.tools.visualvm.modules.startup.AcceptLicenseDialog
-Djava.home=D:\Software\Java\jdk1.6.0
-Dnetbeans.home=C:\VisualVM\platform7
-Dnetbeans.dirs=C:\VisualVM\visualvm;C:\VisualVM\profiles2
-Dnetbeans.user=C:\Documents and Settings\jis\Application Data\visualvm\releases
-Dnetbeans.system_http_proxy_hosts=
-Dsun.awt.keepWorkingSetOnMinimize=true
    
```

Customizable views:  
**DataSourceViewProvider**  
*registered in*  
**DataSourceViewsManager**

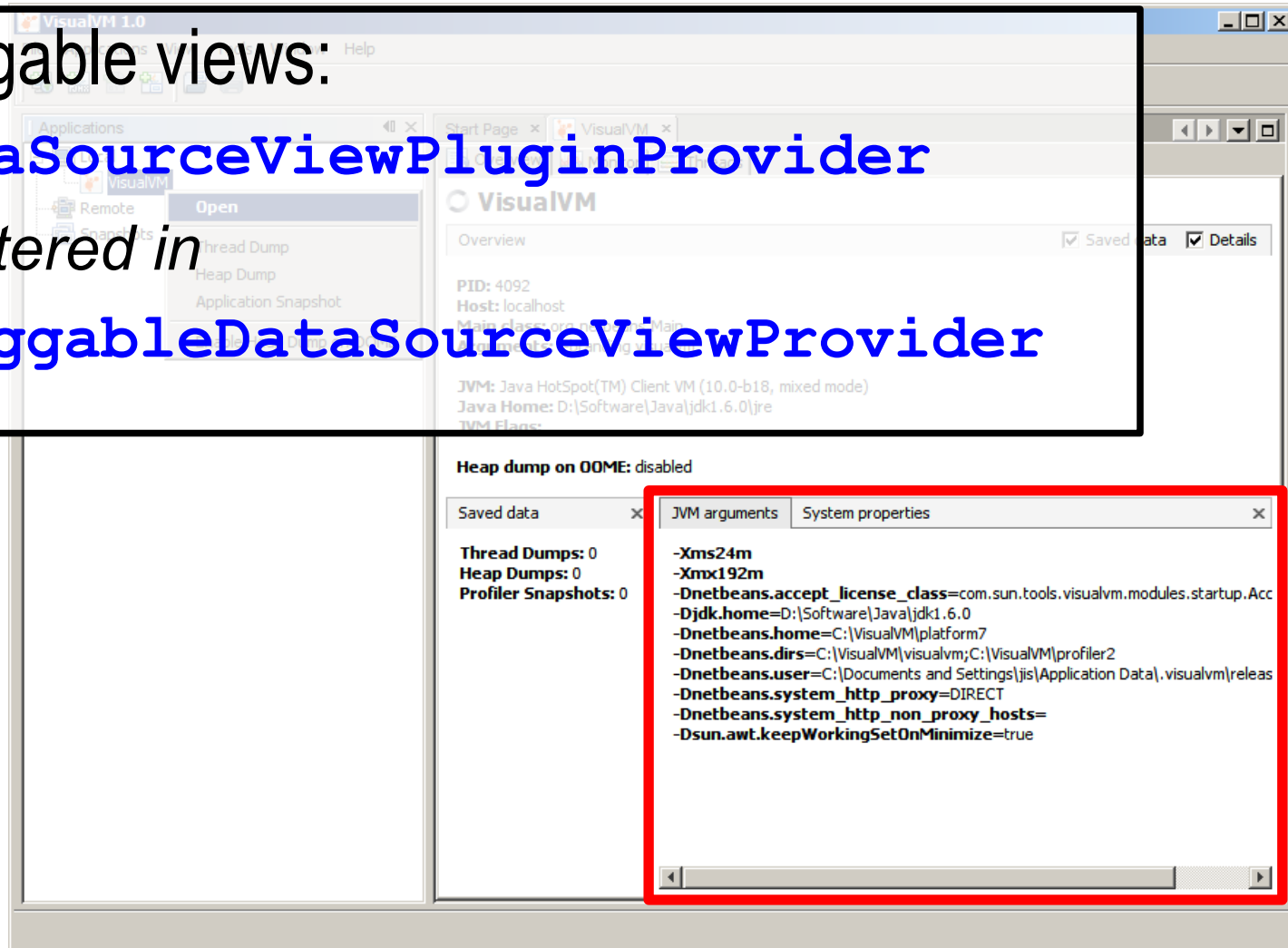
# VisualVM 1.0 API Entrypoints

Pluggable views:

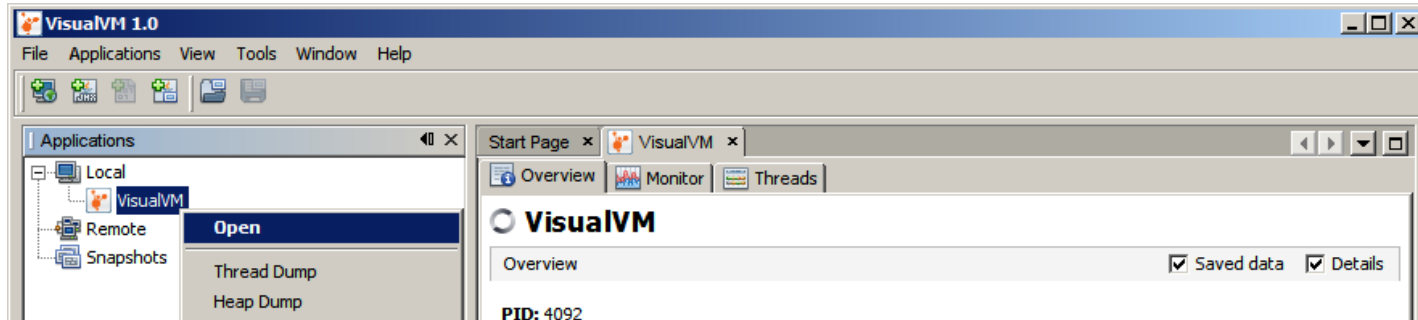
**DataSourceViewPluginProvider**

*registered in*

**PluggableDataSourceViewProvider**



# VisualVM 1.0 API Entrypoints



And much more:

**Jvm, JvmFactory**

**JmxModel, JmxModelFactory**

**JvmMXBeans, JvmMXBeansFactory**

**JvmstatModel, JvmstatFactory**

**JvmJvmstatModel, JvmJvmstatFactory**

**AttachModel, AttachModelFactory**

# Demo

**Thank You!**