



Jagacy VT

Jagacy VT is a feature rich VT100/VT220 terminal emulator written entirely in Java. It supports SSL, ANSI and XTERM protocols.

Jagacy VT requires Java 1.3.1 or higher (Java 1.5.0 or higher for SSL support).

1. Introduction

Jagacy VT is a full-featured terminal emulator that is 100% Java. It can be configured by a system administrator using the properties files `jagacy.properties` and `<userid>.properties`. These are loaded from the directory indicated by the environment variable `JAGACY_PROPERTIES_DIR`. Jagacy VT is discussed in a later section.

2. Features

Jagacy VT is configured using properties. Property values that begin and/or end with spaces can be quoted. Properties are discussed further in the next section.

3. Properties

Pages change: text is added, deleted, and moved. Timeouts change when mainframes are moved or network equipment is replaced.

Jagacy VT reads the property files from the current working directory (unless `jagacy.properties.dir` is set, see below). If one of the files does not exist, it skips it. Jagacy VT reads the properties in the following order:

- 1) `jagacy.properties`,
- 2) `<userid>.properties`,
- 3) System properties

If a property exists in two places, it is overwritten by the second occurrence in the above order.

If the System property `jagacy.properties.dir` is set, Jagacy VT will read the properties files from the specified directory. If the property is set to `classpath`, the CLASSPATH will be searched for the properties file(s) (for the -jar command line option use `jagacy.class.path`). This property can only be specified on the command line or with `System.setProperty()`. If this property is set, at least one of the properties files must exist in the specified directory.

In addition to method properties, Jagacy VT supports the following properties:

Property	Default Value	Allowed Values
<code>jagacy.host</code>	none	Any host name or IP address.
<code>jagacy.port</code>	23	Any valid port number.
<code>jagacy.terminal</code>	VT100	VT100 DEC-VT100 ANSI XTERM
<code>jagacy.class.path</code>	empty string	%CLASSPATH% (for Windows) "\$CLASSPATH" (for Linux/Mac OS X)
<code>jagacy.ssl</code>	false	true false
<code>jagacy.ssl.keyFile</code>	empty string	Valid key file directory and name
<code>jagacy.ssl.keyPassword</code>	empty string	Valid key password.
<code><userid>.logLevel</code> (activated if <code>log4j=false</code>)	error	trace info debug warn error fatal
<code><userid>.logFile</code> (activated if <code>log4j=false</code>)	out	out – System.out err – System.err Any valid file name
<code><userid>.log4j</code>	false	true (Jagacy must be in the classpath before log4j) false
<code><userid>.window</code>	false	true or false
<code><userid>.answerback</code>	empty string	Valid answerback string.
<code><userid>.autowrap</code>	true	true or false
<code><userid>.newlineMode</code>	true	true or false
<code><userid>.mainTextReceiveTimeout</code>	1000	Amount of time (in milliseconds) to wait for text to echo.
<code><userid>.trailingTextReceiveTimeout</code>	10	Amount of time (in milliseconds) to wait

Property	Default Value	Allowed Values
		for trailing characters after text is echoed.
<userid>.tabAfterLabel	true	true if writeAfterLabel should use TAB; false otherwise.

where <userid> is the name of the current user. Properties can be specified in either property file.

4. Jagacy VT

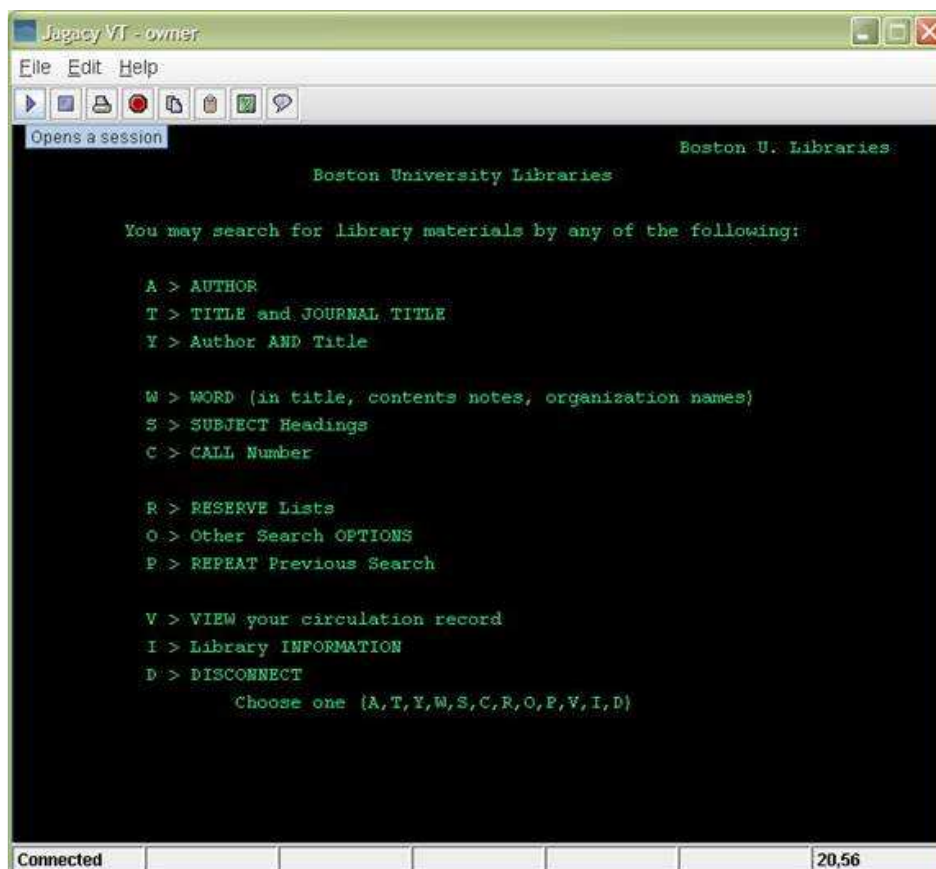
Jagacy VT is a feature rich VT100/VT220ANSI/XTERM emulator. It can be run from the command line as follows:

```
jagacyVT
```

for Windows, and:

```
jagacyVT.sh
```

for Linux and Mac OS X:



Once the screen opens, select `File->Open Session` from the menu, and type in the host name. Once you hit OK, a session will begin. Most keys have their obvious functionality: other keys can be found in the `Help` menu.

During emulation, the following colors indicate the type of field:

Green – Normal,
Light Green – Bold.