



Jagacy V1.2

Jagacy™ is a 3270 screen-scraping library written entirely in Java. It also includes a special 3270 emulator designed to help create screen-scraping applications. Jagacy is faster, easier to use, and more intuitive than HLLAPI. It excels in creating screen-scraping applications reliably and quickly.

Jagacy requires Java 1.3.1 or higher.

Introduction

Screen-scraping allows a user to retrieve information from a mainframe without using a 3270 terminal. It acts as if it is an automated terminal, sending keys and “scraping” information off the mainframe pages. Jagacy can be used in a stand-alone application, in an applet, or in an N-tier client/server environment, where the screen-scraping server is the data/resource tier.

A mainframe page is typically formatted; that is, it is divided into fields, each one preceded with an attribute character:

```
A      AFirst Name:  AJohn
A      ALast Name:   ASmith
A      AStreet:      A55 Main Street
A      ACity:        AAnytown
A      AState:       ANY
A
```

Fields can be protected or unprotected (input fields), intensified, hidden, and numeric or alphanumeric. The attribute character appears as a space on the 3270 screen (field offset = 0). Some pages have no attribute characters: these are known as unformatted pages.

Once a key like the Enter key or one of the PF keys is sent, the keyboard is locked until the mainframe updates the screen. Some mainframe applications unlock the keyboard immediately, some wait until the page to be displayed. Jagacy supports both these modes.

Features

Jagacy supports getting field text, or instead, getting text at specified coordinates. It also supports writing keystrokes and waiting for a specific or general change to occur on the screen, in addition to waiting for the keyboard to unlock. Please refer to the Javadocs for more information.

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

Jagacy also comes with a 3270 emulator (Swing3270). This emulator indicates fields that are protected and unprotected; field number, offset, and length; and row/column coordinates for any character. A Swing3270 window can also be displayed while the screen-scraping program is running. Swing3270 is discussed in detail in a later section.

Properties

Pages change. Fields are added, deleted, and moved. Timeouts increase when mainframes are moved. Jagacy methods support reading field, row/column, and timeout information from properties files (please see the Javadocs for more detail). If one of these changes, the property file can be changed without recompiling code. Jagacy 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 reads the properties in the following order:

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

If a property exists in two places, it is overwritten by the second occurrence in the above order. If you would rather not use properties to specify fields, coordinates, and timeouts, there are methods that allow this too. The only necessary property is `jagacy.host`, and this can be specified on the command line (using the `-D` option) or with `System.setProperty()`.

If the System property `jagacy.properties.dir` is set, Jagacy 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) (note that the `-jar` command line option does not support `classpath`). 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 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>	IBM-3278-2	<u>24x80:</u> IBM-3277-2 IBM-3278-2 IBM-3278-2-E IBM-3279-2 IBM-3279-2-E <u>32x80:</u> IBM-3278-3 IBM-3278-3-E IBM-3279-3 IBM-3279-3-E <u>43x80:</u> IBM-3278-4 IBM-3278-4-E IBM-3279-4 IBM-3279-4-E <u>27x132:</u> IBM-3278-5 IBM-3278-5-E IBM-3279-5 IBM-3279-5-E
<code>jagacy.codepage</code>	INTERNAL	INTERNAL – Default EBCDIC conversion. CP037 – Australian, Canadian, English (U.S.), Netherlands, and Portuguese. CP500 – Swiss, Belgian. CP284 – Spanish. CP285 – English (U.K). CP273 – Austrian/German. CP277 – Danish and Norwegian. CP278 – Finnish and Swedish. CP280 – Italian. CP297 – French. CP871 – Icelandic.

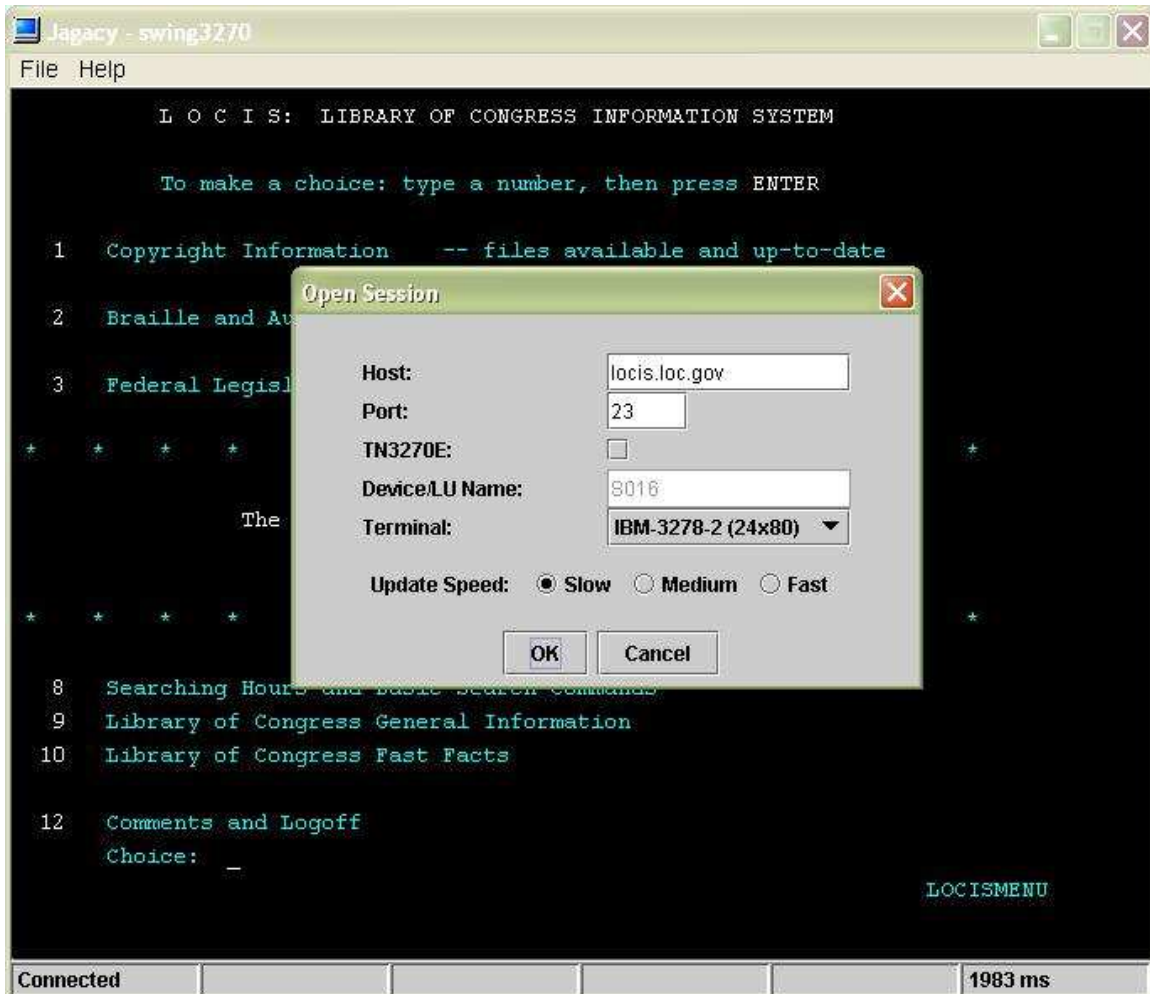
Property	Default Value	Allowed Values
jagacy.sysreq	abort output	abort output interrupt test request
jagacy.attn	break	break interrupt
jagacy.signals	true	true false (useful for Tomcat)
<Session Name>.logLevel	error	trace info debug warn error fatal
<Session Name>.logFile	out	out – System.out err – System.err Any valid file name
<Session Name>.window	false	true or false
<Session Name>.deviceName	empty string	Valid device/LU name

where <Session Name> is the name of the Session specified when a `Session3270` object is constructed.

Properties can be specified in either property file.

Swing3270

Swing3270 is a 3270 emulator tailored to developing and debugging screen-scraping applications:



It can be run from the command line as follows:

```
java -jar jagacy.jar
```

Once the screen opens, select File->Open Session from the menu, and type in the host name. Swing3270 also allows the user to specify the screen update speed (and pausing the screen), to catch intermediate screens that are not normally seen during 3270 operation, but must be considered during screen scraping. Once you hit OK, a session will begin. Use the mouse to click on any character to find out its row, column, field number, field offset, and field length. Most keys have their obvious functionality: other keys (including pausing) can be found in the Help menu.

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

Cyan – Protected,
White – Protected and intensified,
Green – Unprotected,
Red – Unprotected and intensified.

This information (along with the included examples) should serve as a guide for writing screen-scraping applications. For further assistance, please contact support@jagacy.com.

Hints

When writing a screen-scraping application, please keep the following in mind:

- Extend Session3270 and provide logon and logoff routines.
- At startup, you should navigate to a base page in the mainframe application. From this page, any necessary pages should be navigable. This will save time during a server operation.
- Always check to make sure the application is on the correct page, by checking for a string unique to that page. This can be accomplished using Jagacy's wait functions.
- Determine the amount of time it takes for a page to appear and double it. Use the response time at the bottom right of the Swing3270 screen as a guide. This will allow for unforeseen delays in the future.
- Find out from the mainframe application developer any shortcut keys for the application. In the examples, the shortcut "off" is used to navigate back to the main page.
- Always close the session and logoff the userid. This will prevent the userid from being locked. Consider using state variables to indicate what page the application is on, in order to reverse the navigation and logoff. Jagacy automatically handles SIGTERM and SIGINT (Ctrl C) signals (unless `jagacy.signals=false`), and will close the session, and logoff the application, if either occurs.
- Do not run a window in a production environment (set `<Session Name>.window=false`). This may unnecessarily slow down the application.