12/29/2016

Jagacy Software
www.jagacy.com

Jagacy 3270

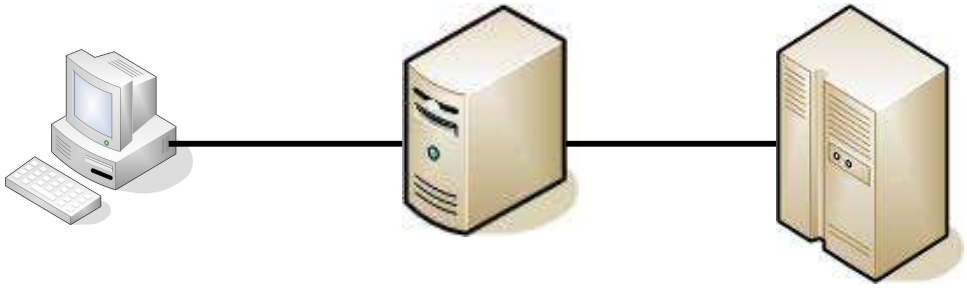
Jagacy 3270 is a feature rich 3270 screen-scraping library written entirely in Java. It supports Internationalization, SSL, TN3270E, and over thirty languages. It also includes a terminal emulator designed to help create screen-scraping applications. Developers can also develop their own custom terminal emulators (with automated logon and logoff). Jagacy 3270 screen-scraping is faster, easier to use, and more intuitive than HLLAPI. It excels in creating applications reliably and quickly.

Jagacy 3270 requires Java 1.6.0 or higher.

1. 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 3270 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 access tier:

N-Tier Architecture



Presentation | Functionality/ Data Access | Data

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 3270 supports both these modes.

2. Features

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

3. Properties

Pages change: fields are added, deleted, and moved. Timeouts change when mainframes are moved or network equipment is replaced. Different languages must be supported for international applications. Jagacy 3270 provides methods for 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 3270 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 3270 reads the properties in the following order:

1) jagacy.properties,
2) <SessionName>.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.

If the System property `jagacy.properties.dir` is set, Jagacy 3270 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 3270 supports the following properties:

Property	**Default Value**	**Allowed Values**
jagacy.host	None	Any host name or IP address.
jagacy.port	23	Any valid port number.
jagacy.useProxy	false	true or false
jagacy.proxy.host	localhost	Any host name or IP address.
jagacy.proxy.port	1080	Any valid port number.
jagacy.terminal	IBM-3278-2	24x80: IBM-3277-2 IBM-3278-2 IBM-3278-2-E IBM-3279-2 IBM-3279-2-E 32x80: IBM-3278-3 IBM-3278-3-E IBM-3279-3 IBM-3279-3-E 43x80: IBM-3278-4 IBM-3278-4-E IBM-3279-4 IBM-3279-4-E 27x132: IBM-3278-5 IBM-3278-5-E IBM-3279-5 IBM-3279-5-E
jagacy.dbcs	false	true or false
jagacy.codepage	INTERNAL	INTERNAL – Default EBCDIC conversion. CP037 – Australian, Canadian, English (U.S.), Netherlands, and Portuguese. CP273 – Austrian/German. CP277 – Danish and Norwegian. CP278 – Finnish and Swedish. CP280 – Italian. CP284 – Spanish. CP285 – English (U.K.), Irish. CP297 – French. CP500 – Swiss, Belgian.

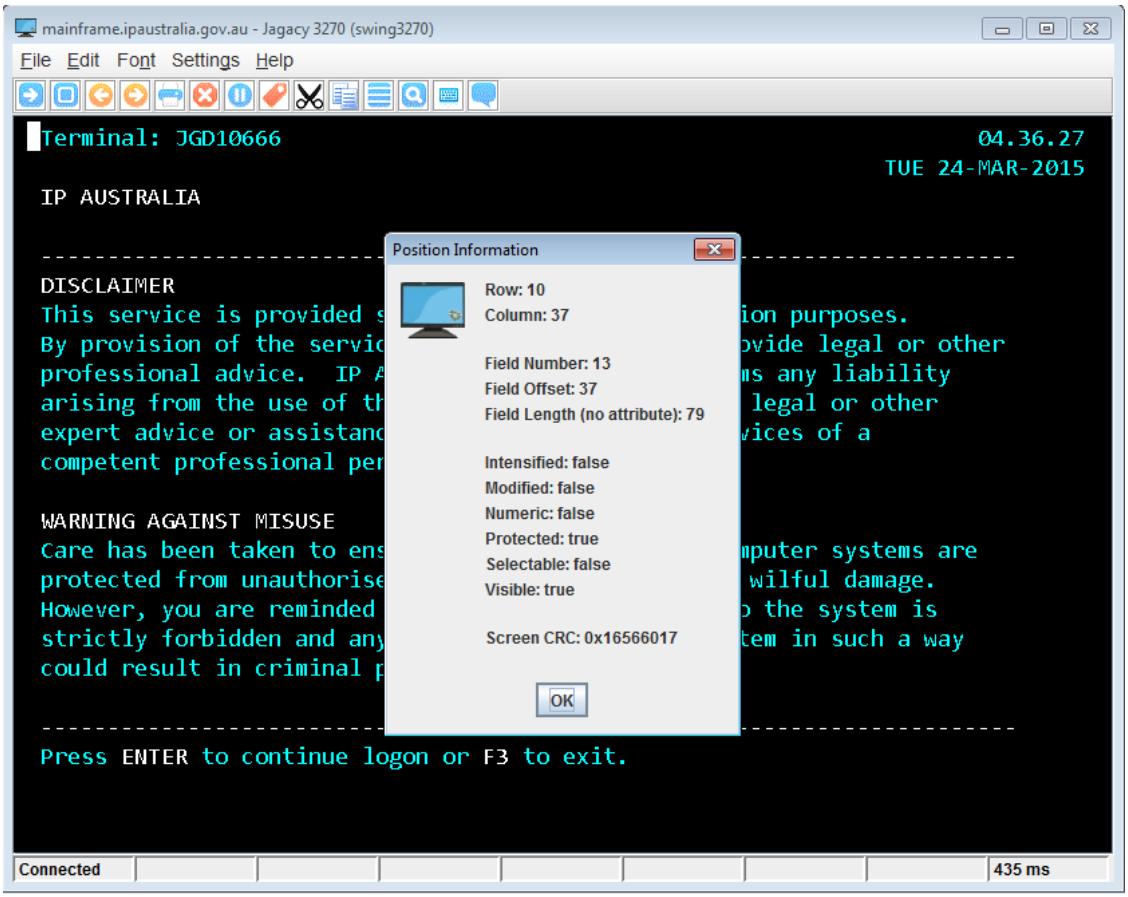
Property	Default Value	Allowed Values
		CP838 – Thai. CP855 – Cyrillic. CP857 – Turkish. CP869 – Greek. CP870 – Multilingual Latin-2. CP871 – Icelandic. CP875 – Greek 2. CP918 – Urdu. CP930 (Japanese DBCS). CP933 (Korean DBCS). CP935 (Simplified Chinese DBCS). CP937 (Traditional Chinese DBCS). CP939 (Japanese DBCS) CP1025 – Cyrillic 2. CP1026 – Latin-5, Turkish. CP1047 – Latin-1. CP1140 – Australian, Canadian, English (U.S.), Netherlands, and Portuguese (with €). CP1141 – Austrian/German (with €). CP1142 – Danish and Norwegian (with €). CP1143 – Finnish and Swedish (with €). CP1144 – Italian (with €). CP1145 – Spanish (with €). CP1146 – English (U.K.), Irish (with €). CP1147 – French (with €). CP1148 – Swiss, Belgian (with €). CP1149 – Icelandic (with €).
jagacy.sysreq	abort output	abort output interrupt test request
jagacy.attn	break	break interrupt
jagacy.signals	true	true false (useful for Tomcat and MQ series)
jagacy.connect.timeout	10000	Timeout in milliseconds.
jagacy.rfc2355	true	true or false
jagacy.autoReconnect	false	true or false
jagacy.strictLU	false	true or false
jagacy.class.path	empty string	%CLASSPATH% (for Windows) $CLASSPATH (for Linux/Mac OS X)
jagacy.ssl	false	true or false
jagacy.ssl.keyFile	empty string	Valid key file directory and name, or :java: for .keystore
jagacy.ssl.keyPassword	empty string	Valid key file password.
jagacy.ssl.provider	None	A valid provider class.

Property	Default Value	Allowed Values
jagacy.ssl.context	SSLv3	TLS, SSLv2, SSLv3
jagacy.ssl.receiveTimeout	10000	Timeout in milliseconds.
<SessionName>.logLevel (activated if log4j=false)	error	all trace watch debug info warn error fatal off
<SessionName>.logFile (activated if log4j=false)	out	out – System.out err – System.err Any valid file name
<SessionName>.logFile.options (activated if log4j=false)	empty string	append=true;date=true
<SessionName>.window	false	true or false
<SessionName>.deviceName	empty string	Valid device/LU name
<SessionName>.log4j	false	true or false
<SessionName>.numericAcceptAny	false	true or false
<SessionName>.showHiddenFields	false	true or false

where <SessionName> is the name of the Session specified when a `Session3270` object is constructed. Properties can be specified in either property file.

4. Swing 3270

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



It can be run from the command line as follows:

```
swing3270.exe
```

for Windows, and:

```
swing3270.sh
```

for Linux and Mac OS X.

Swing 3270 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. Use the mouse to right click on any character to find out its row, column, field number, field offset, field length, and screen CRC.

During emulation (when extended attributes are turned off), the following colors indicate the type of field:

Cyan – Protected,
White – Protected and intensified,
Green – Unprotected,
Red – Unprotected and intensified,
Blue – Protected or unprotected numeric.

Clicking the mouse button moves the cursor to the indicated position, while dragging the mouse marks text for copying.

5. Writing a Screen Scraper

There are three approaches to writing a screen scraper:

a. Searching for unique strings on a screen and writing values at the cursor position,
b. Using row/column positions,
c. Using fields.

In the src subdirectory of the installation directory, the two examples correspond to approaches b. and c. listed above. Furthermore, both approaches can use properties or hard-coded values.

a) src/example1.java:

Example1 navigates through mainframe screens by searching for unique strings and writing values at specified rows/columns. The code is similar to the following:

```
writePosition(3, 26, companyName);
writeKey(Key.ENTER);

if (!waitForPosition(2, 72, “PAEN01MA”, 10)) {
    logger.fatal("Not data screen");
    return;
}

if (getCrc32() != 0x65b6d963) {
     throw new JagacyException(
        "Screen has changed, contact mainframe support");
}
```

You can run example1 by executing `run_example1.bat` (for Windows) or `run_example1.sh` (for Linux and Mac OS X).

b) src/example2.java:

Example2 navigates through mainframe screens by searching for unique strings and writing values at specified fields. The code is similar to the following:

```
writeField(2, 1, companyName);
writeKey(Key.ENTER);

if (!waitForField(1, 232, “PAEN01MA”, 10)){
    logger.fatal("Not data screen");
    return;
}

if (getCrc32() != 0x65b6d963) {
     throw new JagacyException(
        "Screen has changed, contact mainframe support");
}
```

You can run example2 by executing `run_example2.bat` (for Windows) or `run_example2.sh` (for Linux and Mac OS X).

6. Writing a Custom Emulator

The Java class `src/MyEmulator.java` demonstrates how to write a custom emulator with automated logon and logoff routines. You can run MyEmulator by executing `run_MyEmulator.bat` (for Windows) or `run_MyEmulator.sh` (for Linux and Mac OS X). Useful emulator properties are:

Property	Default Value	Allowed Values
jagacy.emulator.showcert	true	true or false
jagacy.emulator.autoTab	false	true or false
jagacy.emulator.history	true	true or false
jagacy.emulator.history.max	10	A numeric value.
jagacy.emulator.printer	None	A printer name.
jagacy.emulator.blinkCursor	true	true or false
jagacy.emulator.pasteBlock	false	true or false

jagacy.emulator.keymap*	None	"C:\Program Files\Jagacy 3270\extra.keymap" for Windows, "/Applications/Jagacy 3270/extra.keymap" for Linux/Mac OS X, or any valid keymap file, ftp, or http location.
jagacy.emulator.language	en	:locale: -- Use Java locale object Other values (examples) are: de, en, es, fr, it, pt.
jagacy.emulator.fontNumber	10	0 = Lucida Sans Typewriter Bold 1 = Courier New Bold 2 = Courier Bold 3 = Letter Gothic Bold 4 = Consolas 5 = Monaco 6 = Lucida Sans Typewriter Regular 7 = Lucida Console 8 = Courier New 9 = Courier 10 = Monospaced 11 = Andale Mono
jagacy.emulator.laf	nimbus	metal nimbus

*System property only.

Note that these properties also work for swing3270.

7. Customizable Language Translations

Jagacy can display languages other than English. Using the `jagacy.emulator.language` property and the translations in the resources subdirectory of the installation directory, any other language can be displayed. Examples included in the directory are de, en, es, fr, it, and pt. Developers are encouraged to submit their translations to info@jagacy.com to be included in the next release.

8. Getting Started Quickly

The following description will demonstrate how easy it is to create a screen scraping application quickly. It uses pseudo-code to represent the steps necessary to implement code rapidly.

The logon Method

This method should sign in and navigate to the “base” page. The base page is the page where all queries, updates, inserts, and deletes originate from.

The logoff Method

This method should be able to navigate back to the sign off page from anywhere within the session. This is usually accomplished by using shortcut commands and keys that the mainframe programmers have placed in the application.

The processing Method(s)

This method should navigate from the base page to the target page, perform queries, inserts, deletes, and/or updates, and navigate back to the base page.

Each method described above uses the same pseudo-code to navigate through pages:

a) Check for the unique string on a page. If the page is the target page, then finished.
b) Check the screen CRC.
c) Enter text and/or keys to navigate to the next appropriate page.
d) Call a wait method.
e) Goto a).

That’s it! For a Java example of how this works, please see the examples in the src subdirectory.

9. Screen-scraping Hints

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

- Intermediate Screens – Using Jagacy 3270 during logon, a few mainframes have intermediate screens. A developer will see that the session has connected but the keyboard is locked. Jagacy 3270 logs warnings when it detects possible intermediate screens. Using Swing 3270 with a slow update speed will show the developer how many intermediate screens exist. We would suggest adding the following method to wait for intermediate changes during logon:

```
private boolean waitForChange(int timeout,
   int intermediateScreenCount) {

if (intermediateScreenCount < 0) {
     throw IllegalArgumentException(
        "Invalid intermediateScreenCount: " +
        intermediateScreenCount)));
   }

   // Add one for target screen:
   intermediateScreenCount++;

   boolean isSuccess = false;
   while (intermediateScreenCount– > 0) {
      isSuccess = waitForChange(timeout);
      if (isSuccess) {
         break;
      }
   }

   return isSuccess;
}
```

- Run through a session using Swing 3270. Note all text, keys, unique strings (with fields and/or coordinates), screen CRCs, and amount of time it takes to navigate through each screen.

- 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 data retrieval.

- 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 3270's wait functions. In addition, verify that the screen's CRC has not changed.

- 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 Swing 3270 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 PF3 key 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 3270 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 `<SessionName>.window=false`). This may unnecessarily slow down the application. Also, set the log level to error, and do not use an extended terminal if possible (for example, use IBM-3278-2 rather than IBM-3278-2-E).

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