2.10.2014

\* created autotest account on rhel1

// add user

system-config-users

// add autotest to labroot group

usermod --gid 16777278 autotest

=====================

Eggplant

\* downloaded eggPlant functional (tgz containing rpm)

provided contact info for free trial license

sudo rpm --verbose --hash --install Downloads/eggPlant\_redhat/eggPlant14.01.rpm

Preparing... ########################################### [100%]

    1:eggPlant ########################################### [100%]

rpm -q --list eggPlant > NOTES/rpm.list.txt

This command gave a 3600+ line of files.

Got evaluation license from Eggplant, good for 2 weeks. This came in an email.

Eggplant email from richard.ward@testplant.com

Ran eggplant from the command line (it resides in /usr/local/bin/eggplant)

Running this for the first time brought up a license manager. Entered license key and left the username empty.

License: 5n5v-bg6f-ezhn-ylyy-hsft-qm

Subsequent runs of eggplant simply brings up eggplant and skips the license step.

A requirement of Eggplant is a vncserver.

Machine rhel1 happens to have tigervnc installed but it doesn't matter.

vncserver :3 -geometry 1900x1020

vnc\* and Xvnc were sftp'ed from rhel1 to rhel10, and placed in ~autotest/vnc.

Ran a vncserver :3 -geometry 1900x1020 on rhel10 and ran eggplant from rhel1.

=================

SIKULI

\* download sikuli jar

https://launchpad.net/sikuli/sikulix/1.0.1/+download/sikuli-setup.jar

\* download OpenCV

http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.8/opencv-2.4.8.zip/download

\* download tesseract (used version tesseract-3.02.02-5.mga4.i586.rpm)

http://rpmfind.net/linux/rpm2html/search.php?query=tesseractNeeded cmake for OpenCV for Sikuli (at a minimum)

Install docs: http://docs.opencv.org/doc/tutorials/introduction/linux\_install/linux\_install.html

Source tarball from: http://sourceforge.net/projects/opencvlibrary/files/latest/download (opencv-2.4.8.zip)

yum search all cmake

That command yielded cmake28.i686

yum install cmake28.i686

This created a cmake28 executable in /usr/bin

In ~autotest, extracted opencv-2.4.8.zip into ~autotest/opencv-2.4.8

cd ~autotest/opencv-2.4.8

mkdir release

cd release

cmake28 -D CMAKE\_BUILD\_TYPE=RELEASE \_D CMAKE\_INSTALL\_PREFIX=/usr/local ..

From release directory (above),

make

This took about 10+ minutes.

sudo make install

This added many files, and executables to /usr/local/bin/

Need tesseract-ocr for Sikuli

tesseract wiki: https://code.google.com/p/tesseract-ocr/

yum search failed - need to build from source

downloaded tarball from https://code.google.com/p/tesseract-ocr/downloads/detail?name=tesseract-ocr-3.02.02.tar.gz

compile docs: https://code.google.com/p/tesseract-ocr/wiki/Compiling

Docs say Leptonica is required to build tesseract

Tesseract 3.02 requires at least v1.69 of Leptonica.

Download leptonica source from: http://www.leptonica.org/source/leptonica-1.70.tar.gz

Leptonica build docs: http://tpgit.github.io/UnOfficialLeptDocs/leptonica/README.html#overview

Untar to ~/leptonica-1.70 and cd there

./configure

make

sudo make install [as root; this puts liblept.a into /usr/local/lib/

                and all the progs into /usr/local/bin/ ]

Build tesseract from source

cd ~/tesseract-ocr

./autogen.sh

./configure

sudo make install

Download tesseract data from: http://tesseract-ocr.googlecode.com/files/tesseract-ocr-3.02.eng.tar.gz

cd ~

tar xvf Downloads/tesseract-ocr-3.02.eng.tar.gz

cd /usr/local/share/tessdata

sudo cp ~/tesseract-ocr/tessdata/\* .

Start sikuli first time: // appears to finalize installation

cd ~/SikuliX

java -jar sikuli-setup.jar

click OK

Select 1 and 5 (Pack1 and tesseract OCR)

click Setup Now

click Yes to download prompt

exits to shell after downloads complete

Start sikuli:

cd ~/SikuliX

./runIDE <= automatically created by previous install step

IDE RUNS BUT CRASHES

SikuliX-1.0.1-SetupLog.txt says:

[debug (2/10/14 1:32:14 PM)] ResourceLoaderBasic: loadLib: Found: VisionProxy

[error (2/10/14 1:32:14 PM)] ResourceLoaderBasic: loadLib: Fatal Error 110: loading: libVisionProxy.so

[error (2/10/14 1:32:14 PM)] ResourceLoaderBasic: loadLib: Since native library was found, it might be a problem with needed dependent libraries

/home/autotest/SikuliX/libs/libVisionProxy.so: libopencv\_core.so.2.4: cannot open shared object file: No such file or directory

[error (2/10/14 1:32:14 PM)] Terminating SikuliX after a fatal error(110)! Sorry, but it makes no sense to continue!

If you do not have any idea about the error cause or solution, run again

with a Debug level of 3. You might paste the output to the Q&A board.

[debug (2/10/14 1:32:14 PM)] SikuliXFinal: cleanUp: 0

DEBUG:

ldd libs/libVisionProxy.so

ldd: warning: you do not have execution permission for `./libVisionProxy.so'

linux-gate.so.1 => (0x003a9000)

libopencv\_core.so.2.4 => not found

libopencv\_highgui.so.2.4 => not found

libopencv\_imgproc.so.2.4 => not found

CORRECTION: re-install sikuli with LD\_LIBRARY\_PATH set

export LD\_LIBRARY\_PATH=/usr/local/lib

mv ~/SikuliX/ ~/SikuliX.1

mkdir ~/SikuliX

cp ~/Downloads/sikuli-setup.jar ~/SikuliX

java -jar sikuli-setup.jar

success

added export of LD\_LIBRARY\_PATH to ~/SikuliX/runIDE script

Attempting to run from the local machine (rhel1):

./runIDE

First thing done was to go to File->Preferences

Set "Where to store images" to /home/autotest/Testing/Sikuli (created that directory manually)

In TextSearch and OCR section of window, clicked "allow searching for text" and "allow OCR"

Clicked "Save" and closed window using "X"

The IDE suggested that the currently running application should be restarted so restarted using

./runIDE

SUCCESS! on both hosts, not sure why previous attempt crashed - maybe because preferences were

not set??

############################

Setting up two nodes to run a common scenario from all tools.

Using rhel1 as host machine from which tests are run

Using rhel10 as system under test (SUT)

Using autotest user on host

Using nagios user on SUT

On host machine, for the autotest user, used ssh-keygen to create ~/.ssh/id\_rsa and ~/.ssh/id\_rsa.pub

On SUT machine, for the nagios user, added the contents of id\_rsa.pub to ~/.ssh/authorized\_keys

To execute a command remotely, from the host to the SUT,

ssh nagios@rhel10 'ls -l'

However we received the error:

reverse mapping checking getaddrinfo for rhel10.uird.local [192.168.195.110] failed - POSSIBLE BREAK-IN ATTEMPT!

Last login: Mon Feb 24 13:56:30 2014 from localhost.localdomain

Fixed this by adding the ip address of rhel10 to the /etc/hosts file in rhel1

Then Bill had the brilliant idea (not being facetious) to add 'sut' to the name in the /etc/hosts file

instead of using rhel10. This way it can change in the future.

### Started a vncserver on the SUT as follows

ssh nagios@rhel10 '/home/nagios/vnc/vncserver -geometry 1800x750 :3'

Needed to set the password for the vnc once.

Ran eggplant and edited the connection to use

user nagios

password nagios

### Running Jenkins

From the ~autotest/Downloads directory:

java -jar jenkins.war --httpPort=8080

8080 is an example, not a requirement. Then to use it, run Firefox and use the port you gave.

Note you can also use https --httpsPort=8080

Created ssh keys

################### Design the common scenario and configure the hosts as needed

February, 25.

The goal of the scenario will be:

\* login to the CIWi web application on rhelserv1

\* use CIWi to install an ADS demo package onto rhel1

\* verify install was successful

\* use CIWi to uninstall the ADS demo package

\* verify uninstall was successful

\* logout of CIWi

Then we will use Jenkins automate running each tool in a passing situation, and

a failure situation and gather metrics for the report from Jenkis data.

To create a situation where the scenario should fail, we will stop

the CIWi Java service. This will cause the CIWi web application to

fail when it attempts to submit an install transaction.

In summary, Jenkins will run:

\* start ciwi service

\* run eggplant scenario expecting success

\* stop ciwi service

\* run eggplant scenario expecting failure

\* start ciwi service

\* run sikuli scenario expecting success

\* stop ciwi service

\* run sikuli scenario expecting failure

\* start ciwi service

\* run ATRT scenario expecting success

\* stop ciwi service

\* run ATRT scenario expecting failure

Created list of generic steps to be performed for the scenario for

each tool (see NOTES/Scenario\_Steps.txt)

From that, we broke down the steps into subroutines by name (see NOTES/Scenario\_Design.txt)

The intent is to implement the scenario with each tool following this design

noting the development time for each tool for comparison in the paper.

################### Host configuration for eggPlant (and other tools)

\* CIWi web application runs on rhelserv1 (url = http://rhelserv1/ciwi/ciwi)

\* defined sut as rhel10 address in /etc/hosts on rhel1 and rhel10

\* using nagois account on sut

\* using autotest account on test host (rhel1)

\* setup ssh keys to allow remote commands from autotest@rhel1 to nagios@rhel10

  to allow starting vncserver on rhel10 from jenkins on rhel1 (and other stuff)

\* setup ssh keys to allow remote commands from wkraemer@rhel1 to wkraemer@rhelserv1

  to allow starting CIWi service from jenkins on rhel1

\* created projects in Jenkins to:

  \* start and stop the CIWi java service on rhelserv1

  \* start and stop vncserver on the sut (rhel10)

  \* run the eggplant scenario for success

  \* run the eggplant scenario for failure

################### Creating Scenarios with eggplant

Eggplant: Started at 1:30 PM February, 25.

Eggplant: Completed at 5:50 PM

Total: 4 hours

However, this included 30 to 45 minutes

of debugging the scenario because CIWi was left in a strange state

when we induced failure. The workaround for this was two fold:

\* detect the error in the eggplant scenario and close

  the error dialog box (click ok)

\* configure firefox to always run in private browing mode so

  that the CIWi session is reset when the browser closes.

Our intent is now to implement these features in the scenarios for

the other tools

NOTE/QUESTION: we made extensive use of OCR in the eggplant scenario.

Eggplant seems to often default to this type of image comparison. Eggplant

also captures a PNG file even though using OCR...??? is it searching

for a similar image and THEN doing OCR???

################### Host configuration for Sikuli

Sikuli does not automatically manage VNC connections like the other tools.

Sikuli is not currently installed on the sut (rhel10)

After much thought about how to manage this to simulate a real world

environment like this (sut and test host are different machines)

we decided on the following:

For test development:

\* manually open a vncviewer on the testhost (rhel1) connected to the sut

\* start sikuli on the test host (rhel1)

\* use Sikuli to manipulate the sut (rhel10) via the vncviewer window

\* NOTE: sikuli IDE displays outside of vnc - we found that this makes

  it easier to capture screen images for menus this way because

  the menus stay open in the vncviewer when we activate the sikuli

  IDE for capture.

For test execution:

\* ensure that xhost + is performed on the sut vncserver

  (added step to Jenkins project that starts vncserver: ssh )

\* run sikuli on test host with DISPLAY exported to sut.

  export DISPLAY=sut:3

  /home/autotest/SikuliX/runIDE -r /home/autotest/AutomatedTest/Scenario/sikuli/Scenario.sikuli

##### later, found this info to help with capuring menu images with sikuli:

http://baydin.com/blog/2010/06/5-sikuli-pitfalls-and-how-to-avoid-them/comment-page-1/

4. Having trouble with context-sensitive/popup menus

If you tried to follow along with the previous example, you likely ran into a problem when you tried to capture the “Control Panel” option in the start menu. After opening the start menu, switching focus back to the Sikuli IDE will cause the start menu to close, thwarting your effort to capture an image. Now, you could use PrintScreen while the start menu is open, paste the image into an image processor, and then use Sikuli to capture the image from the image processor, but thankfully, there’s a better way.

Sikuli installs hotkeys for common tasks like capturing an image (CTRL + SHIFT + 2 by default), and they don’t cause the current program to lose focus. So you can simply open the start menu/context-sensitive menu of your choice and use the hotkey to capture the screen. That way the menu won’t disappear in the process.

BUT - found this also didn't work well when running IDE in vnc (menus close)!!!!!!

################### Creating Scenarios with Sikuli

Sikuli: Started at 3:00 PM February, 26.

Sikuli: Stop at 5:00 PM February, 26

(not complete yet...not really even started!)

Subtotal: 2hrs

Note: about an hour of this was debugging strangeness we were

seeing with sikuli.

Sikuli: Continuing at 9:00 AM February, 27.

10:35 Sikilu IDE has issue - will not allow file save:

[error] SikuliIDE: Problem when trying to invoke menu action doSave

Error: null

Quit and restarted IDE, no apparent issues

Sikuli: Stop at 12:00 PM February, 27

Sikuli: continue at 1:00 PM February, 27

Sikuli: Completed at 1:45 PM February, 27

Subtotal: 3hrs 45 mins

Total time: 5hrs 45mins