

ICAFun

ICA Plugin for JMeter

IT Beratung Jörg Kalsbach

MAR 2011

Version:

2011-03-31: Jörg Kalsbach, several clarifications

2010-12-09: Jörg Kalsbach, Demo Version, Typos

2010-11-18: Jörg Kalsbach, created

About this document

This document describes the ICA Plugin for JMeter. The plugin adds a Citrix ICA Recorder and a Citrix ICA Player to the standard jmeter elements. Recorder and player allow for the recording and replaying of citrix sessions within a jmeter testplan. The plugin is tightly integrated with the usual jmeter logic and allows for the use of bitmap comparisons as assertions within different steps of a citrix testplan.

Target audience

Target audience for this document is anyone who is involved in the business of planning, building or executing load tests against citrix applications.

ICA Plugin for JMeter: Feature list

The ICA Plugin for JMeter offers the following features:

- Tight integration into the jmeter framework
- Capture Citrix sessions
- Group captured interactions into steps
- Display steps as samplers in citrix testplan
- Allow for variable content in captured steps (tags)
- Use csv data sets to substitute variable content during replay
- Allow for the bitmap comparison of regions of interes (ROI) during capture and replay
- Allow for bitmap comparisons as assertions during replay

- Allow for “waitfor” functionality: Wait for the appearance of a previously recorded bitmap in a ROI. If the bitmap does not show up during a configurable interval the step will be marked as failed

Note: bitmap comparisons on ROIs are less complicated than it seems at first glance. A ROI is nothing more than a two dimensional matrix of pixels. A pixel is nothing more than a 3-tuple of Red Green Blue values. So we end up with a two dimensional matrix of 3-tuples of integers. Just a bunch of numbers that can be used to compute a hash value that is (almost) unique to this particular matrix of RGB tuples. During replay the same ROI at the replay screen is used as input for the same hash function and the two hashes are then compared.

ICA Plugin for JMeter: Demo Version

The demo version is restricted in two ways. It allows only one Citrix user. This user can open several sessions in parallel but all sessions belong to the same user. There is no waitfor functionality. The times shown during replay will be solely dependent on sleep times during capture. The bitmap comparison feature works as described.

Requirements

- Understanding of JMeter
- User level understanding of citrix
- Oracle Java SE JRE 1.6 or higher
- Citrix Online plug-in-Web must be installed on the machine running JMeter: search for “Citrix Online plug-in-Web” and install a suitable version

Installation

Let \$jmeter_root be c:\software\jakarta-jmeter-2.4

- Unpack the provided archive to \$jmeter_root\ica-plugin
- Add or modify the search_paths entry in \$jmeter_root\bin\jmeter.properties. The search_paths entry must include: ../ica-plugin/ica-plugin.jar;../ica-plugin/com4j.jar
- Add the following registry key so that the ICA API will be allowed to connect to the Citrix server and the plug-in can obtain a valid session to work with:
HKLM\Software\Citrix\ICA Client\CCM Name: AllowSimulationAPI Type: REG_DWORD Data: 1

Verification

Verify the layout of \$jmeter_root. There must be a directory \$jmeter_root\ica-plugin containing the files com4j.dll, com4j.jar and ica-plugin.jar. Otherwise the plugin will not be available.

To verify the proper working start jmeter as usual. Jmeter.log must contain lines like

```
2011/03/31 10:17:59 INFO - jmeter.JMeter: search_paths=../ica-plugin/ica-plugin.jar;../ica-plugin/com4j.jar
2011/03/31 10:17:59 INFO - jmeter.JMeter: Adding to classpath: ../ica-plugin/ica-plugin.jar
2011/03/31 10:17:59 INFO - jmeter.JMeter: Adding to classpath: ../ica-plugin/com4j.jar
```

Otherwise the plugin will not be available.

Cookbook

This cookbook explains the planing of a capture session, the actual capture session and the construction of a testplan that uses the capture file. Furthermore the section shows how to debug the testplan and how to verify it's proper working.

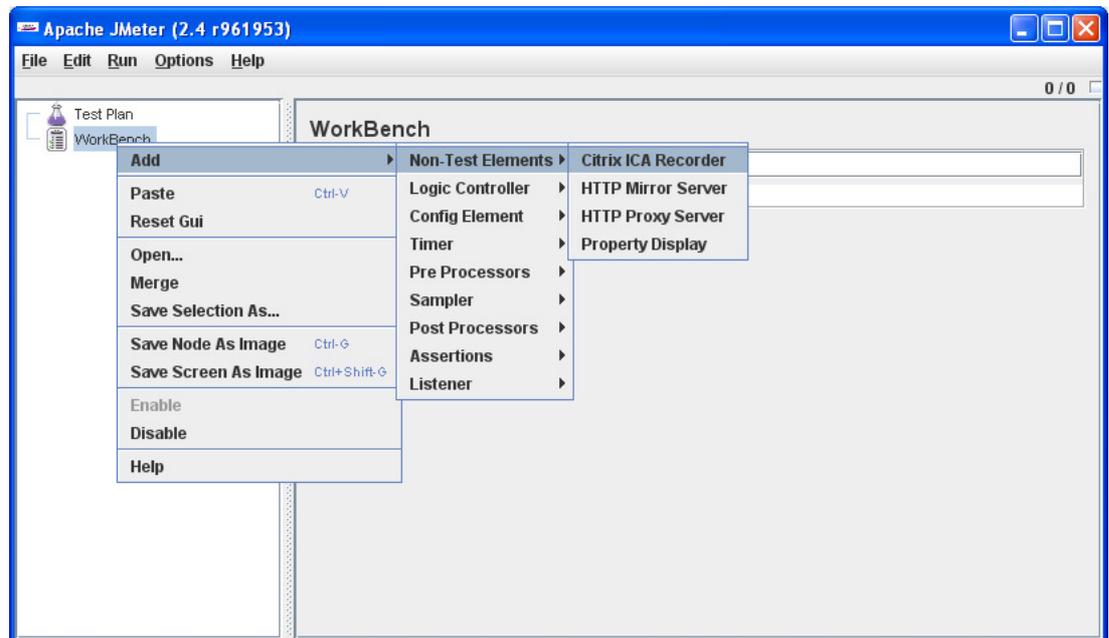
Planning

We should have a clear idea of our testplan before we start our capture session. To demonstrate the capabilities of the ica-plugin for jmeter we stick to a simple plan:

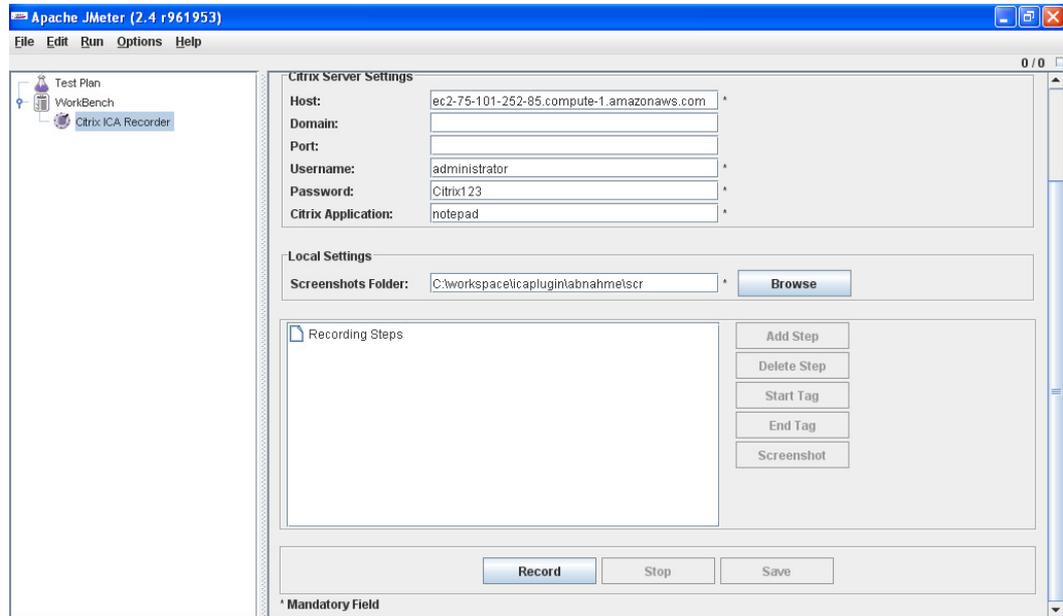
- Open the notepad application.
- Maximise the window with ALT-SPACE-x (Note: this is highly recommended to make the capture more robust).
- Enter a text into the notepad application. At runtime this text will be read from a csv dataset.
- Save the file with a filename. At runtime this filename will be read from a csv dataset.
- Take a screenshot from the notepad application after saving the file. This screenshot proves that the system works as expected.
- Exit from the notepad application. This closes the remote session.

Capture

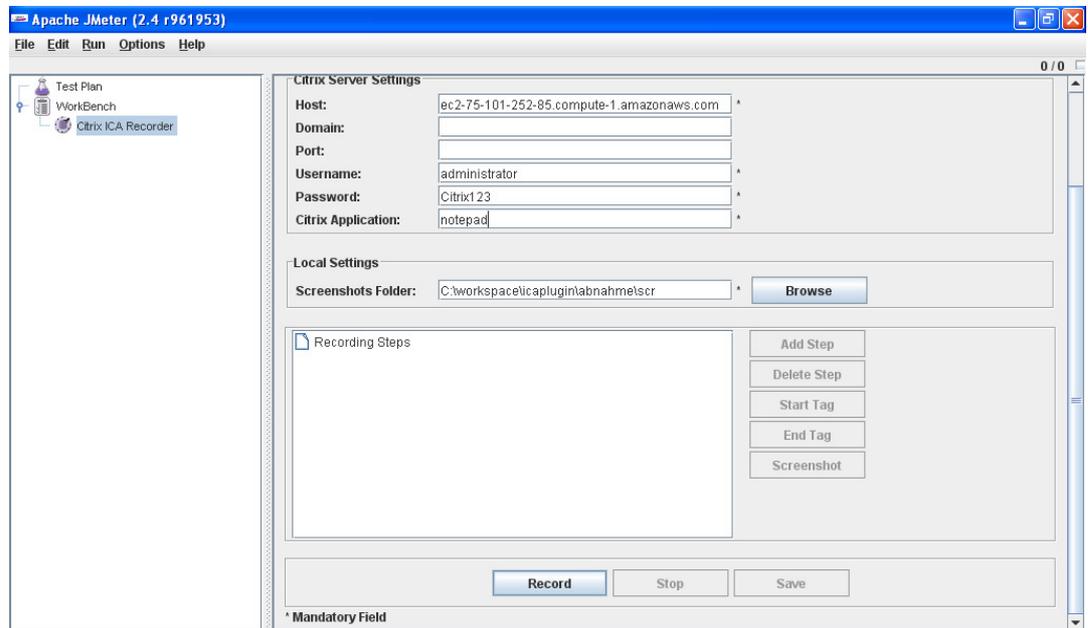
We open jmeter as usual. In order to prepare the capture we add a Citrix ICA Recorder as a non test element to the workbench:



We configure the recorder. We need at least the hostname of the citrix server, username for the login, password for the login and the application we want to test. This application has to be published for the user. Note: for convenience the recorder element should be saved for future use. This way we can reuse it in future recording sessions.

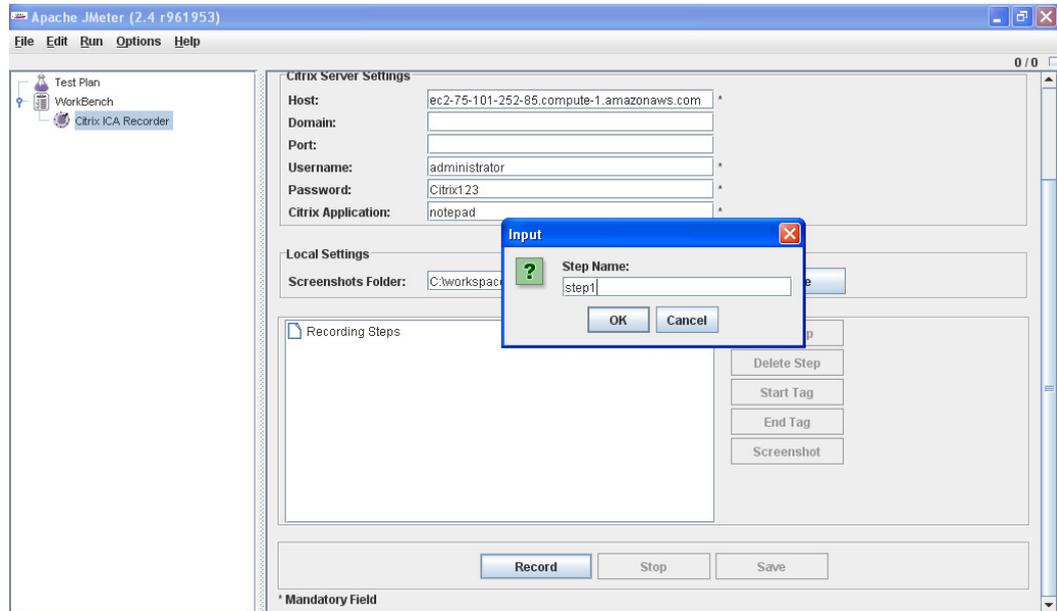


We start the recording by hitting the record button.

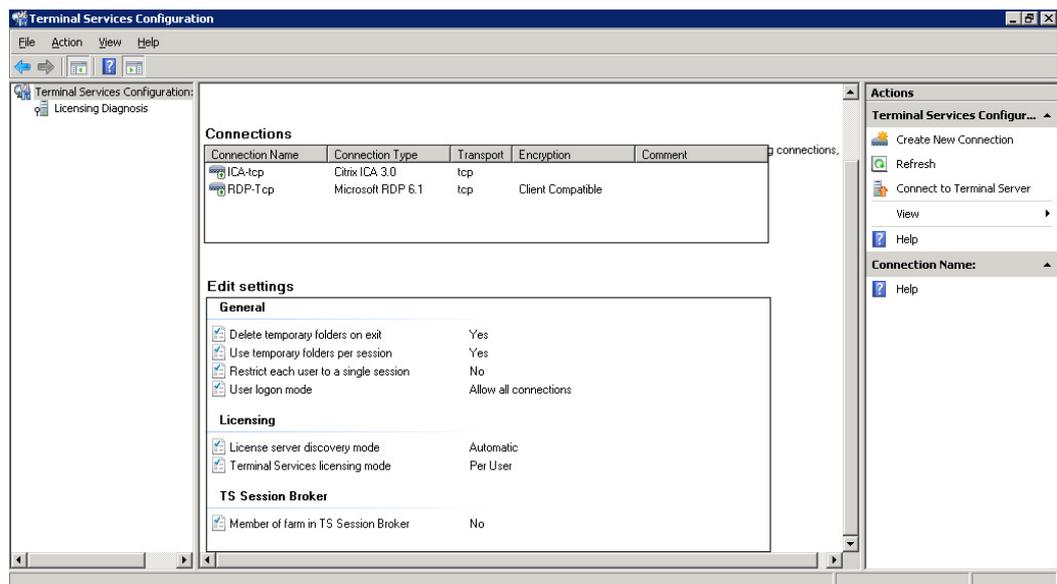


We choose "step1" as name for the first step of our recording.

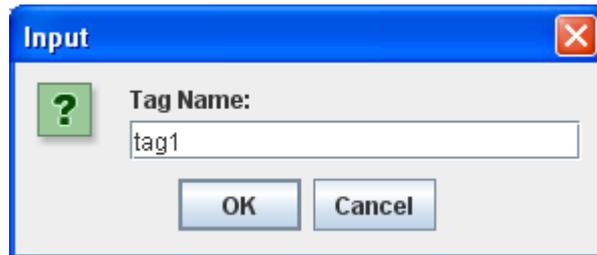
ICAfun



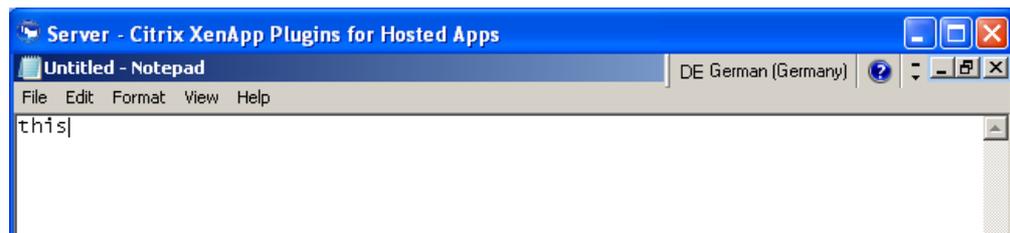
Now a normal citrix session builds up. Note: during replay we will use multiple threads to simulate multiple connections. To allow multiple sessions the citrix server has to be properly configured. On the server the "Restrict each user to a single session" setting has to be set to "No".



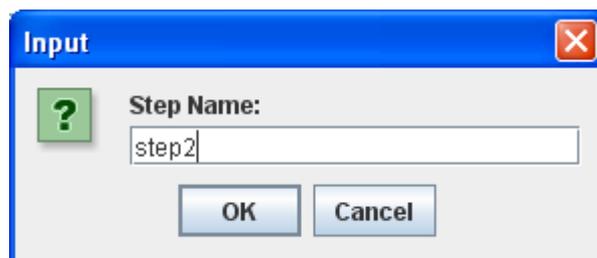
We maximise the notepad application with ALT-SPACE-x. The text we enter into notepad shall be variable. Therefore we start a tag with the name "tag1".



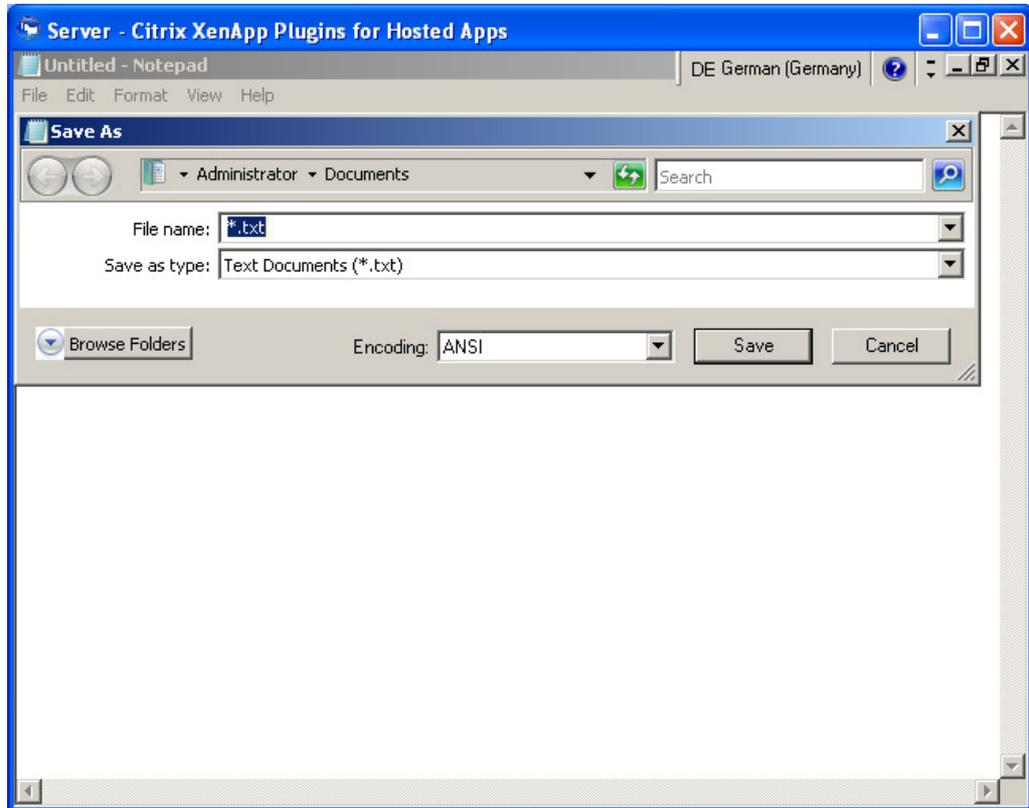
Now we enter the text of our choice into notepad.



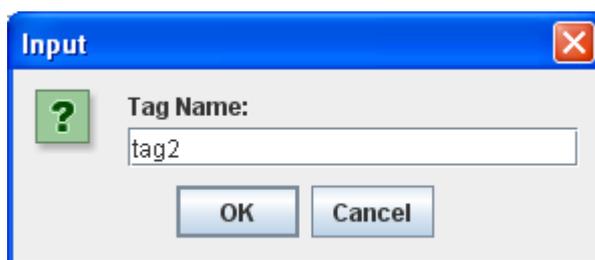
We end the tag to indicate where the variable text ends. We start a new step named "step2".



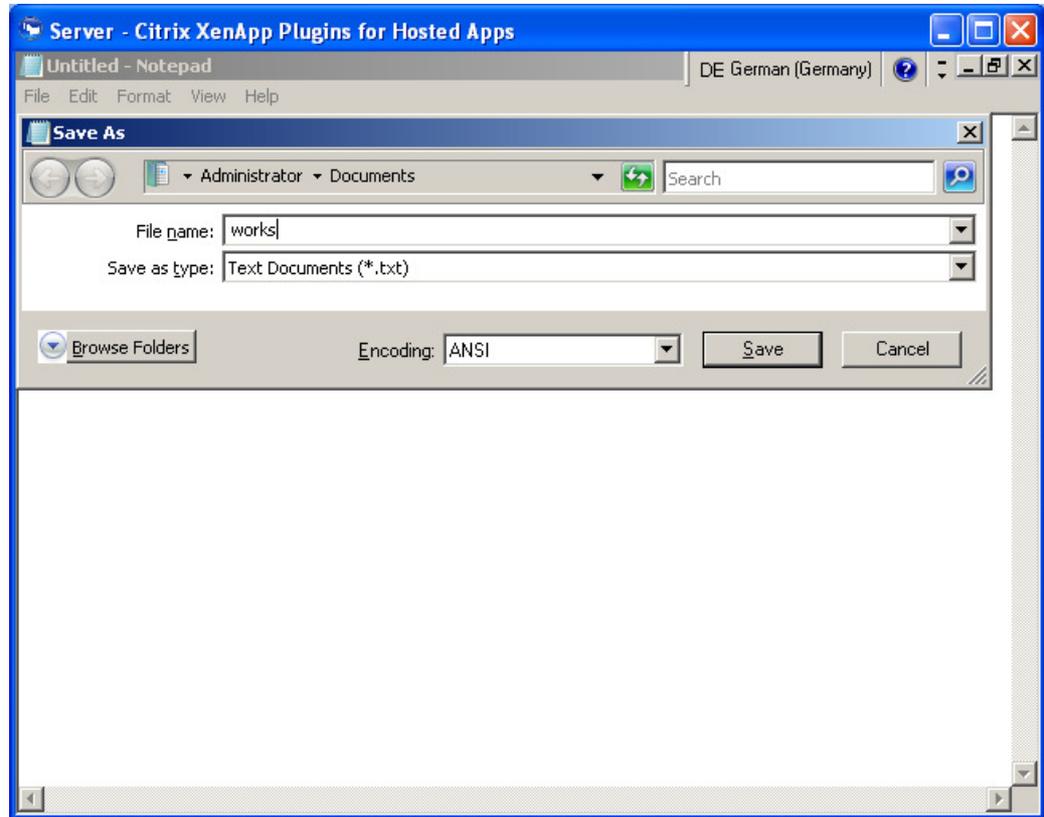
We choose "file-save" in the menu. The filename shall be variable.



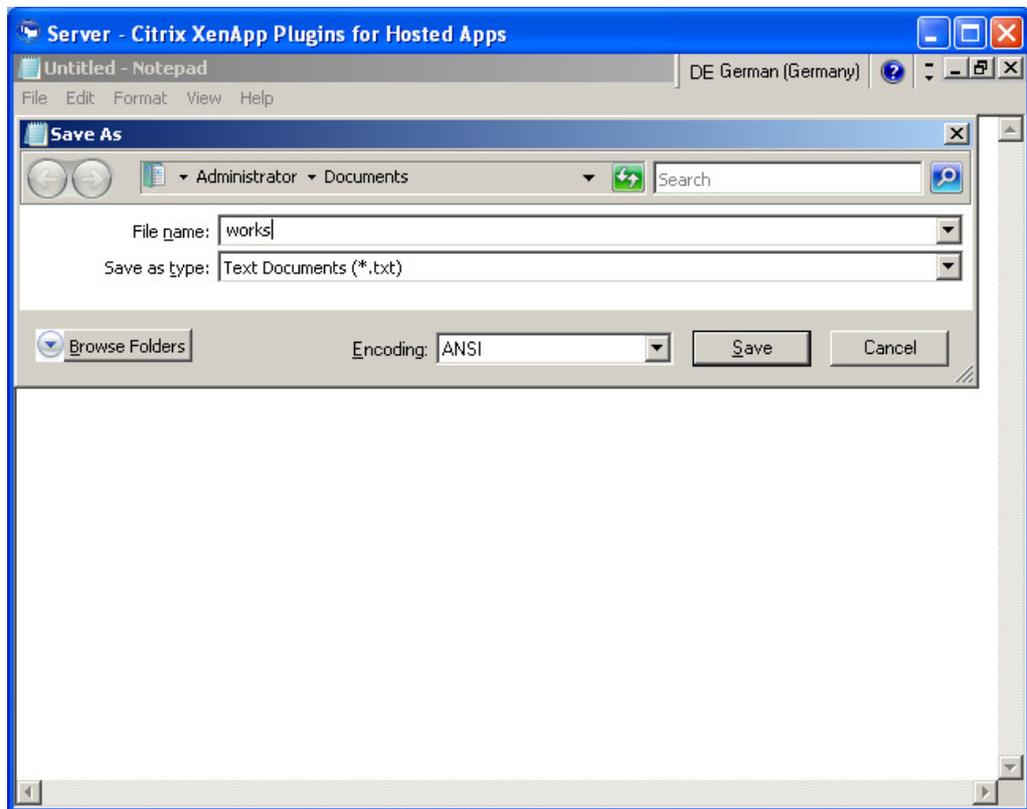
So we start a new tag named „tag2“ ...



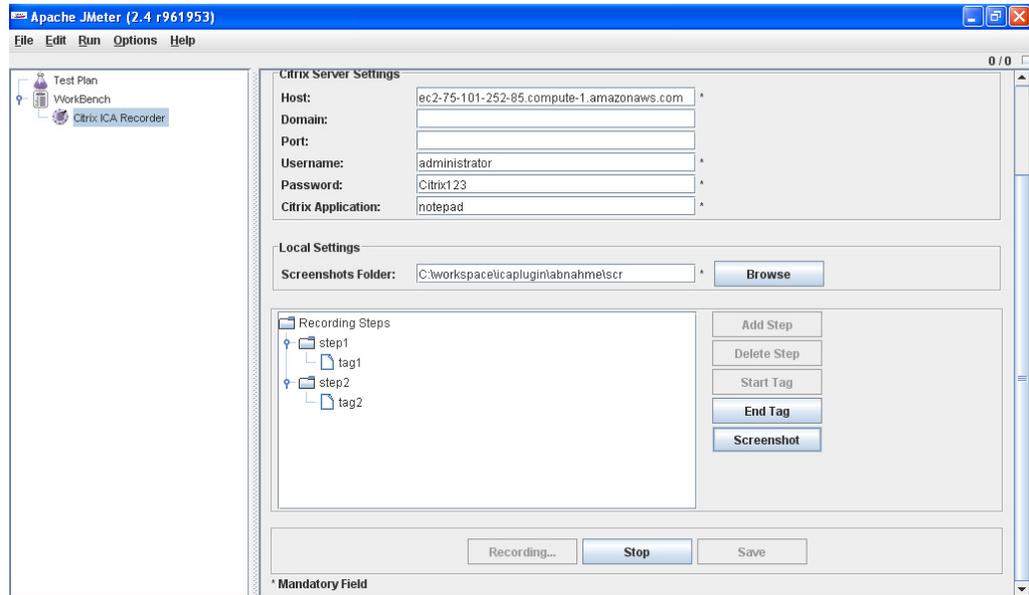
... and enter the filename of our choice.



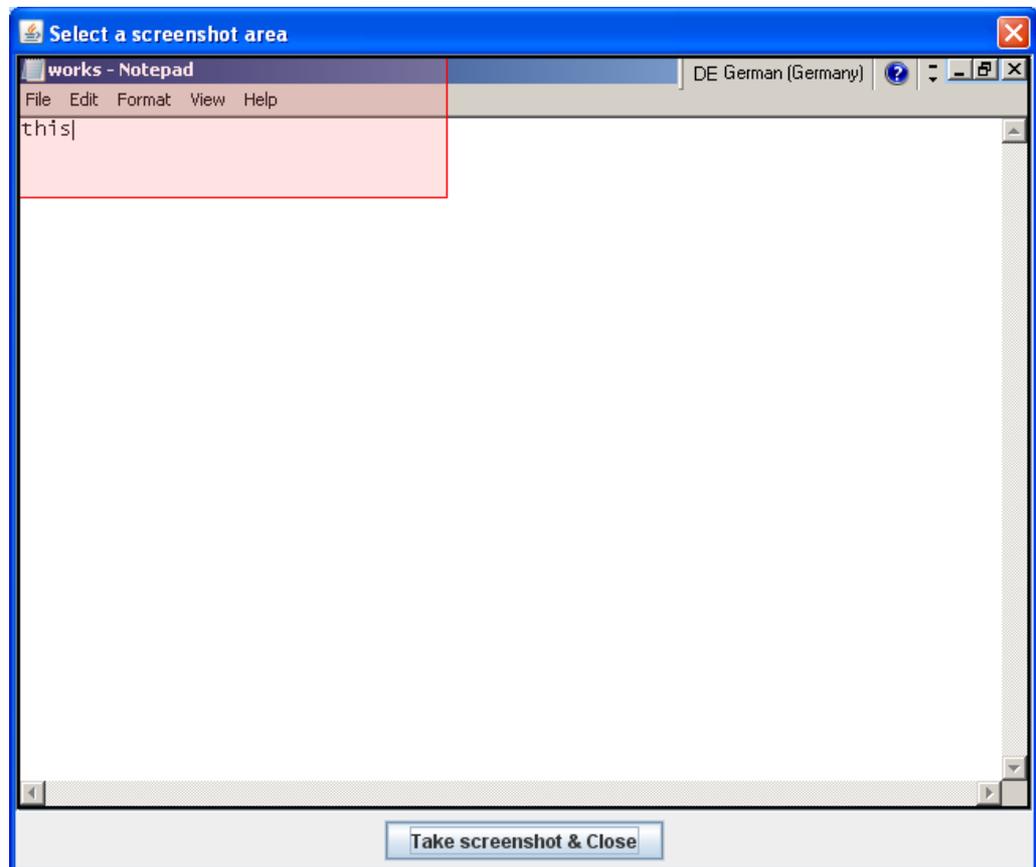
We end the tag and save the file.



We use a hashmap computed on a region of the screen to verify the proper working of the replay. We press the screenshot button ...



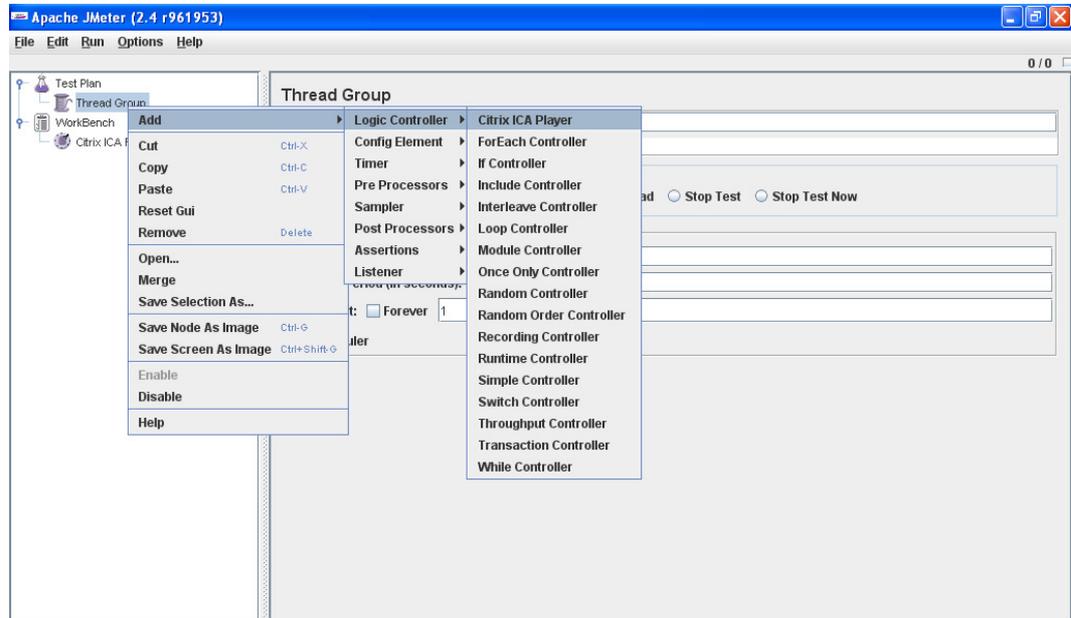
... choose a region of interest (aka „ROI“) and press the “Take screenshot and close” button. Note that our notepad session now displays the filename.



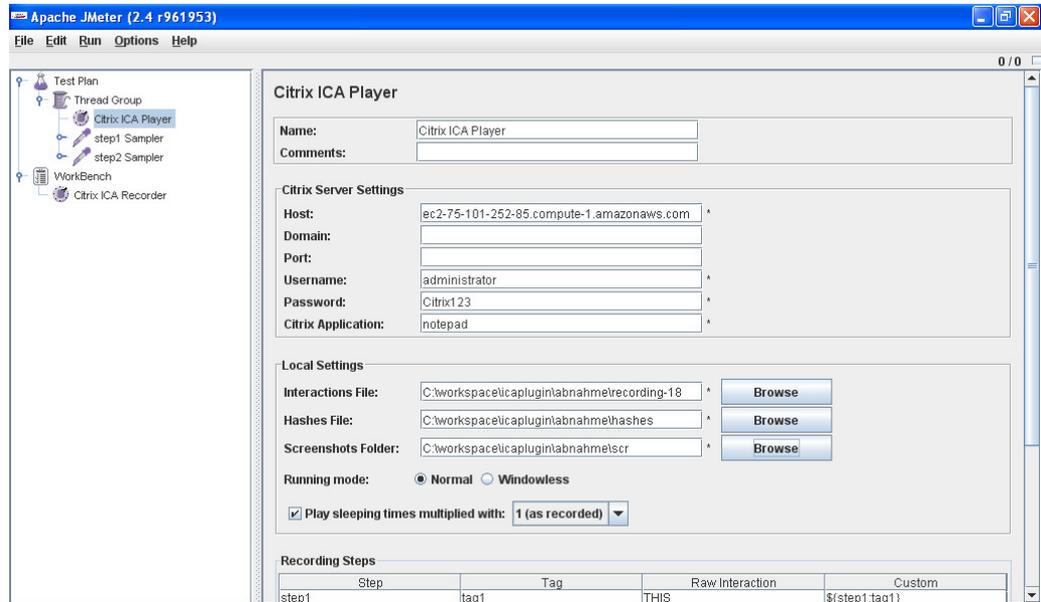
Now we exit the notepad application („File – exit“) wait until the session closes, stop the recording and save the capture into a file. We are through with the capture, sigh and start right into the replay.

Replay

We construct a testplan for the replay. We add a thread group with one thread and two runs. We add a Citrix ICA Player as a logic controller to the thread group.



We configure the player: Citrix server settings should be obvious. The interactions file points to the saved recording. The hashes file can be an arbitrary file. During bitmap comparisons the player dumps the bitmap hashes into this file. The screenshot folder can be any folder. During replay screenshots are saved into this directory. Note how the steps are displayed as samplers.

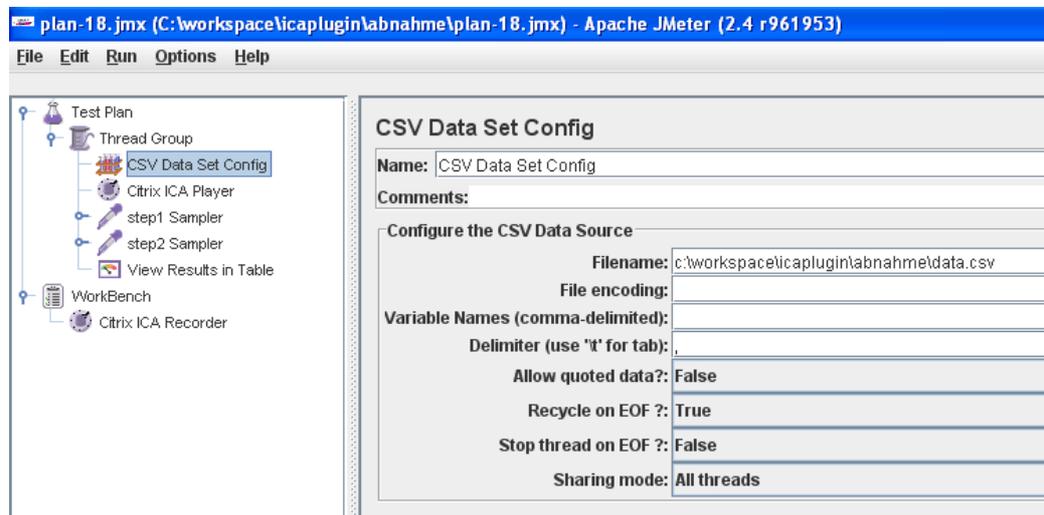


Now we prepare a data file for the csv dataset config element. Note the relationship between column headers, step names and tag names. Finally we add a “View results in a table” listener.

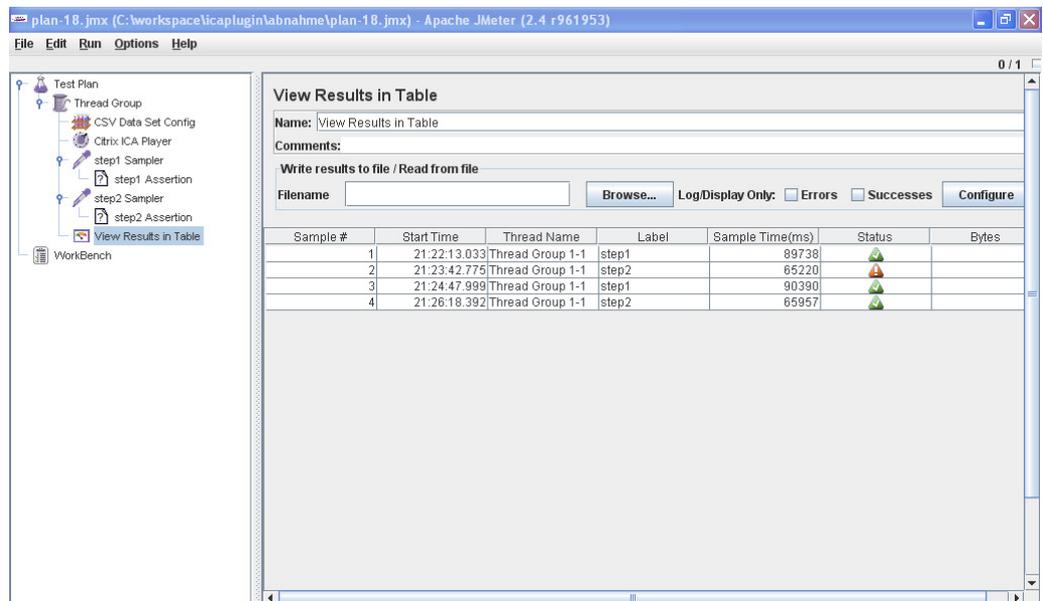
Listing of data.csv:

```
step1:tag1,step2:tag2  
this,fails  
this,works
```

We add a csv data set config `_before_` the player. We do not need to specify the variable names as the data files contains properly named headers.



Now we are ready to start the test. We enjoy the replay and see the expected result. We made a screenshot of a notepad session with the text "this" and the filename "works". Step two fails in the first run (filename: "fails") and succeeds in the second run (filename: "works")



Now we can play with the thread group settings: make it two threads and one run. Our confidence in the replay grows. Once we are satisfied with our testplan we

configure the player to run in windowless mode to save resources. With this particular testplan it is of course necessary to delete the generated files after each run. The plan serves as proof of concept.

Now we are ready to start building real test plans according to our needs.

Hints

- Build repeatable test plans! The cookbook testplan is an example for a non repeatable testplan. The second iteration will fail due to an “overwrite existing file” dialog. If you enter a filename “a” in the first run and try to enter a filename “aa” in the second run the run can fail as windows will offer you the filename “a” in an autocompletion dropdown.
- Use ALT-SPACE-x to maximize your application window.
- Avoid using the ALT-TAB combination when switching between citrix window and jmeter window. The keys will be recorded and may wreak havoc during replay.
- Use the mouse to switch between citrix window and jmeter window. Switching windows is necessary to start and stop steps and tags.
- Take care to enter nothing but text between start tag and end tag. All key events within a tag will be substituted during replay. This means that control sequences will be lost!
- Use bitmap assertions wisely in order to capture only the relevant ROIs for screenshots.
- Try to capture the test on the same hardware you use for replay. The bitmap assertion mechanism is by its very nature vulnerable to display intricacies (colourdepth, display size and friends)
- Verify your testplan by running it in single thread and normal mode. It will save you time in the long run.
- Manually turn off NUMLOCK after a replay session.



Contact:

Jörg Kalsbach / j.kalsbach@jk-itberatung.de / +491781898024

Have fun!