PANDORAFMS

ENTERPRISE

Pandora UX monitoring plugin

INDEX

1. INTRODUCTION

Plugin that allows standard execution of prerecorded PDR and PWR sessions, revision OUM 745 06/03/2020.

2. GENERAL PARAMETERS

Plugin execution configuration

-exe

path to your favorite automation utility binary or testing tool. Also you can select an internal mode::

** Windows only:
PDR - complex desktop testing tools
** windows & linux:
PWR - complex web testing tools (Selenium tests)
** requires curl in PATH:
stats - quick web server performance monitoring
** Linux only:
goliat – quick web checks based on Pandora Webserver

-args *(optional)*

arguments for the automation utility

-script

script(s) to be executed
you can define as many as you need. In example:
-script script1,script2,script3

-folder *(optional)*

folder where the screenshots will be stored

-name *(optional)*

* PWR: define transaction name
* PWD: phase name, define as many as you need.
example: -name phase1,phase2,phase3
they would match the defined scripts

-pwr_port *(optional)*

port where PWR is listening
default 4444, requires -exe PWR

-pwr_host *(optional)*
host where PWR is listening
default localhost, requires -exe PWR

-pwr_browser *(optional)*
browser to use with PWR checks
*default Firefox, requires -exe PWR

-pwr_classic *(optional)*
use old Test::WWW::Selenium libraries, instead PandoraFMS::WebDriver, requires -exe PWR, default 0 (use this parameter to launch your test with Selenium v2)

-retries *(optional)*
number of max retries in case of error

-ss_config *(optional)*
optional screenshot settings: X,Y,Width,Height
in example: -ss_config 0,0,100,100 to get a 100x100 rectangle from 0,0 coords
[Windows only]: set -ss_config active to capture active window

-checkpoint *(optional)*
create new control screenshot

-s *(optional)*
macro substitution, changes every instance of _anytext_ with the value you set. Place now+1 as value to add a day to current date
You can define as -s -f flags as you need. They will be paired in order:
-s macro1=value1 -f format1 -s macro2=value2 -f format2

-f *(optional)*
format for the date macro substitution
use std notation:
example: -f "%y%m%d" for 2016, Sep 13 to obtain 160913

-k *(optional)*
use errorlevel only for check if the automation utility has failed. By default is being check if the utility reports any instance of '[error]'

-max_differences *(optional)*
Specifies the max allowed % of difference between control and retrieved images. By default is set to 0.5. Numeric.

Email configuration

-to *(optional)*
mail account to be notified in case of error

-from *(optional)*
mail remitent

-subject *(optional)*
mail subject

-content *(optional)*
mail content

-smtp_gw *(optional)*
smtp gateway, default 127.0.0.1

-smtp_port *(optional)*
smtp Gateway port, default 25

Pandora FMS module configuration

-t name *(optional)*
(only PDR mode) enables and sets the 'global' transaction name so, every test run, will be associated to the main transaction modules named 'UX XXXX name'

-interval *(optional)*
configures a module interval to generated modules

-g *(optional)*
configures a module group to generated modules

-post *(optional)*
post command after executing the test. Example, kill firefox:
-post "taskkill /F /IM firefox.exe"
-post "killall firefox"

-tag_list x *(optional)*
add x (comma separated) tags to the generated modules, example: tag1,tag2,...

Extra configuration tokens

-exit_on_fail *(optional)*
exits if cannot connect to PWR server. Default 0

-phase_snapshot *(optional)*
generates screenshots per fase. Default 0

-classic_mode *(optional)*
sends latency report always(0), sends latency report only when successful test (1). Default 0

-v *(optional)*
be verbose

-agent *(optional)*
defines the execution as server plugin, sets the name of the target agent where the data will be stored

-agent_group *(optional)*
agent group, if token “agent” is defined

-interval *(optional)*
interval, if token “agent” is defined. Default 300

-tentacle_ip *(optional)*
Pandora Server's IP, default 127.0.0.1

-tentacle_port *(optional)*
tentacle port, default 41121

-tentacle_opts *(optional)*
tentacle extra options

-local_folder *(optional)*
default /var/spool/pandora/data_in

-mode *(optional)*
[local|tentacle] transfer mode, default local

3. SESSIONS AUTOMATION

3.1 Web transaction execution

3.1.1 Standard execution

To launch pre-recorded PWR sessions, indicate that the working mode is PWR, and the file that contains the session guidelines. Its execution in Windows is:

```
pandora_ux_x64.exe -exe PWR -script tests\std.html
```

Optionally, it is possible to indicate the folder where we want no store the screenshots in case of failure in our tests and the ip and port in which the selenium hub that will perform the check is hosted.

```
pandora_ux_x64.exe -exe PWR -script tests\std.html -folder C:\sondas\ -
pwr_host 192.168.80.40 -pwr_port 4445
```

The following modules will be returned:

- UX_Status_project_name: if the sequence succeeded or failed.
- UX_Time_project_name: time taken to complete the sequence.
- UX_Snapshot_project_name: screenshot right before the error, if any.

Example of successful execution:

```
 <module>
        <name><![CDATA[UX_Status_std.side]]></name>
        <type>generic_proc</type>
        <data><![CDATA[1]]></data>
        <description><![CDATA[Test OK]]></description>
</module>
 <module>
        <name><![CDATA[UX_Time_std.side]]></name>
        <type>generic_data</type>
        <data><![CDATA[16.317]]></data>
        <description><![CDATA[Test OK]]></description>
</module>
```

Output with erroneous execution example:

```
<module>
        <name><![CDATA[UX_Status_std.side]]></name>
        <type>generic_proc</type>
        <data><![CDATA[0]]></data>
        <description><![CDATA[Failed to execute verifyText]]></description>
</module>
```

```
<module>
        <name><![CDATA[UX_Time_std.side]]></name>
        <type>generic_data</type>
        <data><![CDATA[15.463]]></data>
        <description><![CDATA[Failed to execute verifyText]]></description>
</module>


<module>
        <name><![CDATA[UX_Snapshot_std.side]]></name>
        <type>async_string</type>
        <data><![CDATA[
iVBORw0KGgoAAAANSUhEUgAA…JRU5ErkJggg==]]></data>
        <description><![CDATA[Image (last error)]]></description>
</module>
```

3.1.2 Phase-based execution

If you have a transactional recording with Pandora FMS UX PWR, the system itself will generate the required modules to identify each of the indicated stages, so the execution will not change regarding the previous case. Just indicate the corresponding side file, which in this case will contain the different stages. Windows execution example

```
pandora_ux_x64.exe -exe PWR -script tests\std.side -folder C:\sondas
```

The following modules will be returned by stage:

UX_Time_project_name.phase_order
UX_Status_project_name.phase_order

If there is any phase with an error, the following module will also be created:

UX_Snapshot_project_name.phase_order

It will display an image of the web at the time of the error.

The global summary modules identified with the following names will also be returned:

UX_Global_Time_project_name
UX_Global_Status_project_name
UX_Global_Snapshot_project_name

A web image at the time of the error will be displayed.

3.2 PDR session execution

3.2.1 Standard execution

To launch pre-recorded PDR sessions, the working mode indicated is the path to the pdr.cmd file displayed at the corresponding point, the argument of said "-r" file, the file that contains the session directives (-script), the directory where to store the screenshots (-folder) ending in "\", which is optional, where to save the screenshots in the folder where the pdr is located. You may also enter the number of consecutive retries in case of failure, optional parameter.

In the next run, the screen capture is also customized to collect only the active window:

```
pandora_ux_x64      -exe      C:\PDR\pdr.cmd      -args      -r      -script
C:\pandora_ux\calculadora.sikuli -folder C:\pandora_ux\ -ss_config active
```

The following modules will be returned:

UX_Time_project_name.
UX_Status_project_name.
UX_Control_Snapshot_project_name (only on the first run).

If there is an error at any stage, the following module will also be created:

UX_Snapshot_project_name.

And it will show an image of the active window (with -ss_config active) from when the failure took place.

Example of successful execution:

```
 <module>
        <name><![CDATA[UX_Status_calculadora.sikuli]]></name>
        <type>generic_proc</type>
        <data><![CDATA[1]]></data>
        <description><![CDATA[C:\pandora_ux\calculadora.sikuli       execution
completed
Control snapshot rebuilt
]]></description>
        <tags>UX</tags>
        <module_group>UX</module_group>
</module>
<module>
        <name><![CDATA[UX_Time_calculadora.sikuli]]></name>
        <type>generic_data</type>
        <data><![CDATA[20.204]]></data>
        <description><![CDATA[C:\pandora_ux\calculadora.sikuli       execution
completed
Control snapshot rebuilt
]]></description>
        <tags>UX</tags>
```

```
        <module_group>UX</module_group>
        <module_parent>UX_Status_calculadora.sikuli</module_parent>
</module>
<module>
        <name><![CDATA[UX_Control_Snapshot_calculadora.sikuli]]></name>
        <type>async_string</type>
        <data><![CDATA[
IBCAIAAAAOCnfhAAAAAXNSR…/4x79e/7757f8H2C00s1C73yMAAAAASUVORK5CYII=]]></data>
        <description><![CDATA[Control image rebuilt]]></description>
        <tags>UX</tags>
        <module_group>UX</module_group>
        <module_parent>UX_Status_calculadora.sikuli</module_parent>
</module>
```

Example of output with failed execution:

```
<module>
        <name><![CDATA[UX_Status_std.html]]></name>
        <type>generic_proc</type>
        <data><![CDATA[0]]></data>
        <description><![CDATA[Failed to execute verifyText]]></description>
        <tags>UX</tags>
        <module_group>UX</module_group>
</module>
<module>
        <name><![CDATA[UX_Time_std.html]]></name>
        <type>generic_data</type>
        <data><![CDATA[15.463]]></data>
        <description><![CDATA[Failed to execute verifyText]]></description>
        <tags>UX</tags>
        <module_group>UX</module_group>
        <module_parent>UX_Status_std.html</module_parent>
</module>
<module>
        <name><![CDATA[UX_Snapshot_std.html]]></name>
        <type>async_string</type>
        <data><![CDATA[
iVBORw0KGgoAAAANSUhEUgAA…JRU5ErkJggg==]]></data>
        <description><![CDATA[Image (last error)]]></description>
        <tags>UX</tags>
        <module_group>UX</module_group>
        <module_parent>UX_Status_std.html</module_parent>
</module>
```

3.2.2 Transaction-based execution

If you have recorded different processes with PDR and you have checked they work by playing them continuously, run:

```
pandora_ux_x64.exe -exe C:\PDR\pdr.cmd -args -r -t calculadora_trans -script
C:\PDR\calc.sikuli,C:\PDR\savecalc.sikuli,C:\PDR\savefile.sikuli       -folder
C:\PDR\ -ss_config active
```

As it can be seen, just indicate the path of the new script in the -script parameter separated by a comma from the previous script and use the -t parameter with the name of the transaction that will include the different phases . If the process had more phases, the same logic would be followed, for example:

```
pandora_ux_x64.exe -exe C:\PDR\pdr.cmd -args -r -t proceso_transaccional -
script   C:\PDR\script1,C:\PDR\script2,C:\PDR\script3,C:\PDR\script4   -folder
C:\PDR\ -ss_config active
```

The line to add to the agent configuration file, in this case, is the following::

```
module_plugin   pandora_ux_x64.exe   -exe   C:\PDR\pdr.cmd   -args   -r   -t
calculadora_trans                                                    -script
C:\PDR\calc.sikuli,C:\PDR\savecalc.sikuli,C:\PDR\savefile.sikuli       -folder
C:\PDR\ -ss_config active -checkpoint -post "taskkill /F /IM calc.exe"
```

Thanks to the -checkpoint parameter, you may see screenshots of the result of each phase in the Pandora FMS console.

The following modules will be returned by stage:

UX_Time_project_name.phase_order
UX_Status_project_name.phase_order

If there is any phase with an error, the following module will also be created:

UX_Snapshot_project_name.phase_order

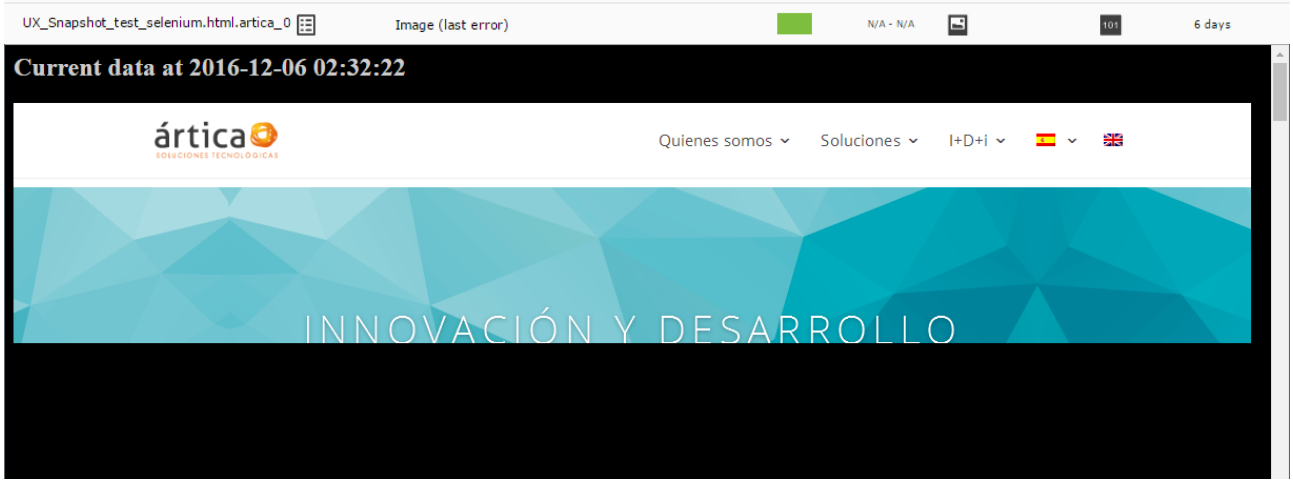
It will also display an image of the web at the time of the failure, should it take place.

The global summary modules identified with the following names will also be returned:

UX_Global_Time_project_name
UX_Global_Status_project_name
UX_Global_Snapshot_project_name

It will also display an image of the web at the time of the failure.

Example of screenshot with incomplete transaction.



Note: The error image will only be shown when the UX client and the PWR server are running on the same machine. Otherwise, the directory for the delivery of images by the PWR server must be accessible by the client in order to display the image in Pandora.