



Supervision d'expérience utilisateur UX et WUX



om:

<https://pandorafms.com/manual/!776/>

permanent link:

https://pandorafms.com/manual/!776/fr/documentation/pandorafms/monitoring/13_user_monitorization

2024/06/10 14:34



Supervision d'expérience utilisateur UX et WUX

Introduction

E Le contrôle de l'expérience utilisateur consiste à enregistrer des tâches automatisées de navigation sur le web (PWR/WUX) et d'interaction avec le bureau et les applications du système Windows (PDR). Ces enregistrements peuvent aller d'un clic de souris sur un navigateur web à l'ouverture d'une application sur le bureau, en passant par la saisie d'un texte ou la réalisation d'une recherche sur le web. Cela permet de configurer des fonctions et de les enregistrer automatiquement en vue d'une exécution ultérieure à la recherche de résultats.

Différences entre supervision UX et WUX

Ces deux systèmes de surveillance servent à l'exécution de tâches automatisées de navigation web grâce au système Pandora Web Robot Daemon (PWRD).

Le système UX réalise des surveillances grâce à l'agent software installé sur une machine, tandis que la supervision WUX se base sur un serveur intégré dans Pandora FMS.

Pandora Web Robot Daemon (PWRD) est un service qui prévoit les outils nécessaires pour automatiser les sessions de navigations web. Pour cela, il utilise une archive qui contient une liste des actions nécessaires pour naviguer sur le portail web de l'utilisateur.

Par ailleurs, la surveillance UX permet l'exécution de tâches automatisées qui interagissent avec le bureau et les applications du système Windows®. Ce type de tâches ne peuvent être réalisées avec WUX.

Pandora FMS UX est un système qui exécute des tâches automatisées, remettant à Pandora FMS un rapport avec les résultats de l'exécution, le temps requis et des captures avec les éventuelles erreurs rencontrées.

Si vous disposez d'un système d'automatisation de tâches, Pandora FMS UX permet aussi d'exécuter les scripts que vous avez déjà et d'en surveiller leur exécution.

Il est important de préciser l'usage de ce type de surveillance. Ces deux méthodes se basent sur l'exécution de la supervision grâce aux plugins dans les agents software installés dans les machines correspondantes.

Pour l'exécution des transactions web Pandora FMS utilise Selenium Engine.

- Selenium version 2.
- Selenium version 3.

Surveillance PWR UX

Compte tenu des **différences entre la surveillance UX et la surveillance WUX**, de manière générale, la surveillance UX doit respecter une série de conditions préalables liées à l'environnement à surveiller, avec les étapes préliminaires suivantes :

- Installer Java®.
- Créez un profil dans Mozilla Firefox®.
- Installation d'un service Selenium®.
- Répartir les PWR dans le système.
- Installez l'IDE de Selenium® pour Mozilla Firefox®.
- Enregistrement d'une session PWR.

Pour l'exécution standard de sessions préenregistrées, utilisez la fonction **Plugin Pandora UX** dans sa dernière version 28-04-2022.

Expérience utilisateur Web (WUX) avec Selenium 3

Pour le déploiement de Selenium dans les serveurs WUX on utilisera un stack de conteneurs pour un déploiement rapide et facile à étendre.

Configurations préalables

Docker et Docker Compose doivent être installés ; on recommande l'utilisation de CentOS en tant que système d'exploitation de base.

Pour installer Docker, suivez la documentation selon votre système d'exploitation dans

<https://docs.docker.com/engine/install/>

Pour Selenium, utilisez les images officielles pour l'installation et déploiement du stack dans:

<https://hub.docker.com/u/selenium>

Le répertoire de Selenium met à votre disposition de différentes images avec navigateurs. Pour Pandora FMS, nous vous recommandons les conteneurs Mozilla Firefox® et Google Chrome®.

Déploiement du stack Selenium

Afin de déployer le stack Selenium, créez un fichier en format YAML avec les configurations nécessaires :

```
# To execute this docker-compose yml file use `docker-compose -f up`
# Add the `-d` flag at the end for detached execution
version: "3"
services:
  selenium-hub:
    image: pandorafms/pandorafms-selenium-hub
    mem_limit: 2G
    container_name: selenium-hub-v3
    logging:
      driver: "json-file"
      options:
        max-file: "5"
        max-size: "4m"
        mode: "non-blocking"
    environment:
      - TZ=Europe/Amsterdam
    ports:
      - "4444:4444"

  chrome:
    image: pandorafms/pandorafms-selenium-node-chrome
    mem_limit: 2G
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - selenium-hub
    logging:
      driver: "json-file"
      options:
        max-file: "5"
        max-size: "4m"
        mode: "non-blocking"
    environment:
      - TZ=Europe/Amsterdam
      - HUB_HOST=selenium-hub
      - HUB_PORT=4444

  firefox:
    image: pandorafms/pandorafms-selenium-node-firefox
    mem_limit: 2G
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - selenium-hub
    logging:
      driver: "json-file"
```

```
options:
  max-file: "5"
  max-size: "4m"
  mode: "non-blocking"
environment:
  - TZ=Europe/Amsterdam
  - HUB_HOST=selenium-hub
  - HUB_PORT=4444
```

Dans l'exemple ci-dessus, vous devez apporter les modifications nécessaires pour chaque cas, comme la limite de mémoire, le fuseau horaire, le nombre de nœuds, etc.

Vous pouvez garder le fichier avec le nom `docker-compose.yml` pour rendre son identification plus facile.

Pour activer le conteneur de la configuration que vous avez défini, il faut juste d'exécuter (remplacer `<compose-file>` par le nom que vous avez choisi pour le fichier) :

```
docker-compose -f <compose-file> up -d
```

Pour vérifier les services en exécution dans le conteneur, utilisez :

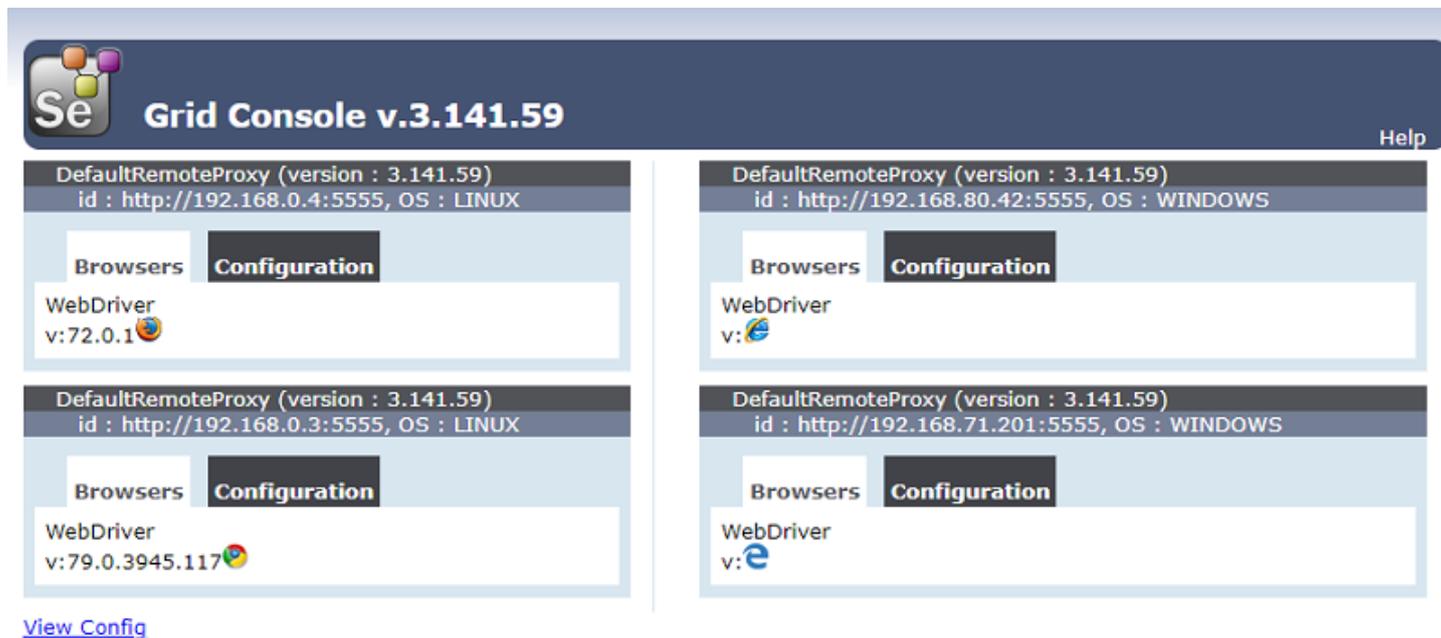
```
docker-compose -f <compose-file> ps
```

Pour afficher l'état et les journaux des services Selenium, utilisez la commande suivante :

```
docker-compose -f <compose-file> logs
```

Une fois les vérifications pertinentes sont faites, pour vérifier que le *grid* fonctionne correctement et que les *workers* ont été enregistrés comme nous avons défini dans le fichier de configuration, accédez à l'URL suivante :

```
http://<ip_selenium_server>:4444/grid/console
```



The screenshot displays the Selenium Grid Console v.3.141.59 interface. At the top left is the Selenium logo (Se) and the version number. A 'Help' link is visible at the top right. The main area shows four worker nodes arranged in a 2x2 grid. Each node has a header bar with 'DefaultRemoteProxy (version : 3.141.59)' and its ID and OS. Below each header are two tabs: 'Browsers' and 'Configuration'. The 'Browsers' tab is active for all nodes, showing the WebDriver version and a browser icon. The nodes are:

- Top-left: ID: http://192.168.0.4:5555, OS: LINUX, WebDriver v:72.0.1, Chrome icon.
- Top-right: ID: http://192.168.80.42:5555, OS: WINDOWS, WebDriver v:72.0.1, Edge icon.
- Bottom-left: ID: http://192.168.0.3:5555, OS: LINUX, WebDriver v:79.0.3945.117, Chrome icon.
- Bottom-right: ID: http://192.168.71.201:5555, OS: WINDOWS, WebDriver v:79.0.3945.117, Edge icon.

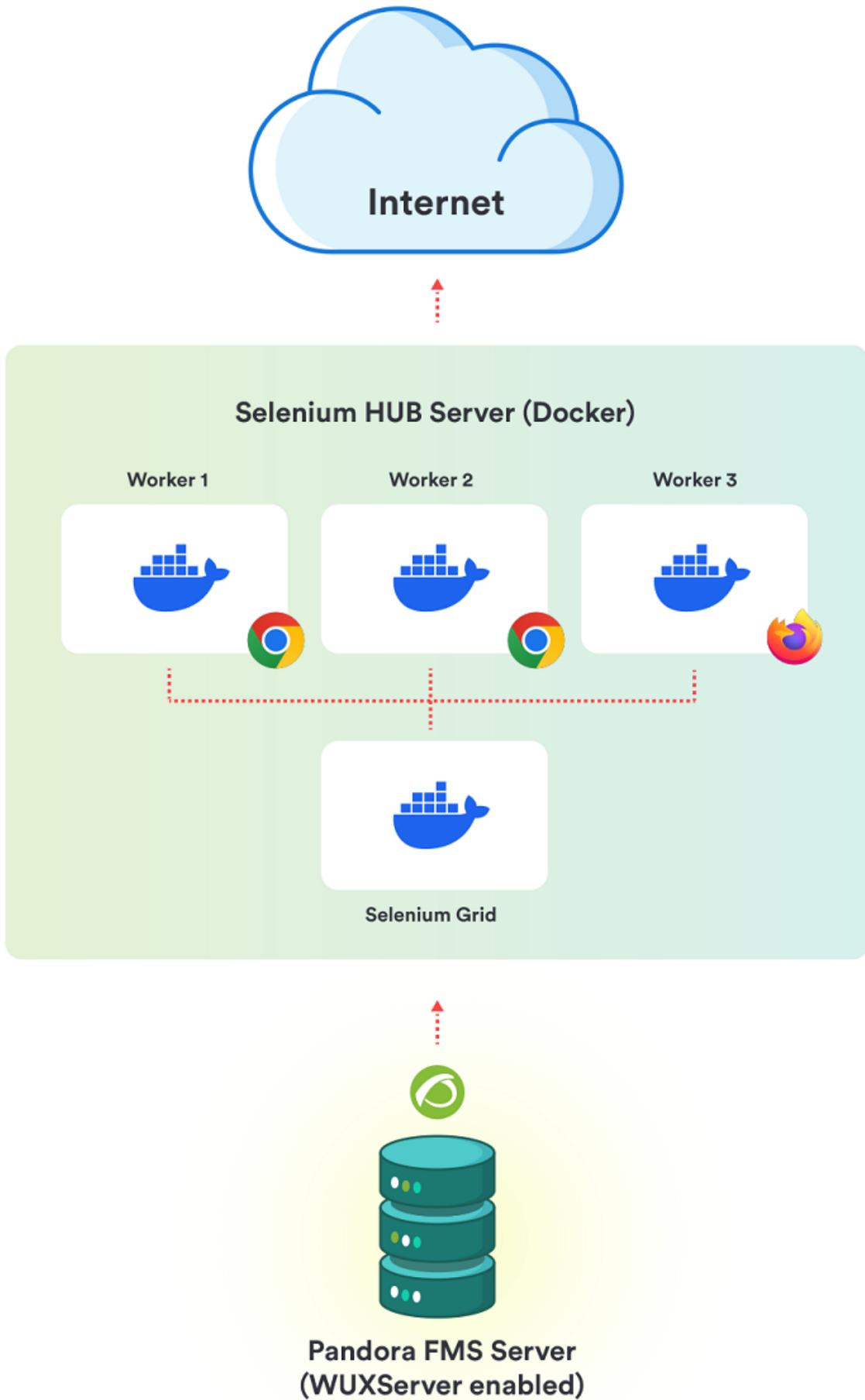
A [View Config](#) link is located at the bottom left of the grid.

Si vous avez besoin d'augmenter le nombre de *workers* :

```
docker-compose -f <compose-file> scale chrome=X firefox=Y
```

Infrastructure du service Selenium

Selenium fonctionne en tant que HUB dans lequel un conteneur est activé, qui fonctionne en tant que grille où déployer les conteneurs worker dont vous avez besoin.



Configuration de Pandora FMS

Pour l'utilisation du mode centralisé (WUX) il est nécessaire d'appliquer la configuration suivante au serveur Pandora FMS.

Quel que soit le mode choisi, une fois initié vous pouvez attribuer des exécutions aux sessions de navigation, en ajoutant les paramètres de configuration WUX server au fichier de configuration du serveur Pandora FMS.

Ajoutez la configuration suivante à la fin du fichier `/etc/pandora/pandora_server.conf` (remplacez `ip_wux_host` par l'adresse IP du serveur, si c'est le même serveur qui exécute le serveur Pandora FMS, utilisez `127.0.0.1`) :

```
wuxserver 1
wux_host <ip_wux_host>
wux_port 4444
```

Le fichier de configuration du serveur a un nouvel jeton pour éliminer les sessions qui peuvent être en file d'attente chaque fois que le serveur Pandora FMS est initié.

```
clean_wux_sessions 1 #(default)
```

Gestion des fils du Pandora FMS

- La gestion des threads de `wuxserver` se fait automatiquement lors du démarrage du service `pandora_server`.
- Elle est effectuée en tenant compte du nombre de nœuds d'un navigateur MINOR dans le hub Selenium. Par exemple :
 - Si 2 nœuds Firefox et 2 nœuds Chrome sont configurés dans le *hub*, le nombre de threads `wuxserver` sera 2.
 - Si 1 nœud Firefox et 4 nœuds Chrome sont configurés dans le *hub*, le nombre de threads sera 1.
 - Si 6 nœuds Firefox sont configurés dans le *hub*, le nombre de threads sera 6.

Notez que chaque thread indique les sessions qui peuvent être envoyées simultanément du `wuxserver` au *hub* Selenium.

Annexe : Ajouter des workers pour Internet Explorer et Microsoft Edge

S'il est nécessaire de lancer des transactions web contre des navigateurs Microsoft®, il est nécessaire de configurer une machine (qu'elle soit physique ou virtuelle) avec la version de Windows® souhaitée et configurer le contrôleur ou *driver* en suivant la documentation officielle.

Documentation pour l'installation du *driver* Internet Explorer® :

```
github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver
```

Nous recommandons l'utilisation du driver 3.141 version 32bit pour éviter les problèmes de performance de la version 64bit.

Documentation pour l'installation du *driver* de Microsoft Edge® :

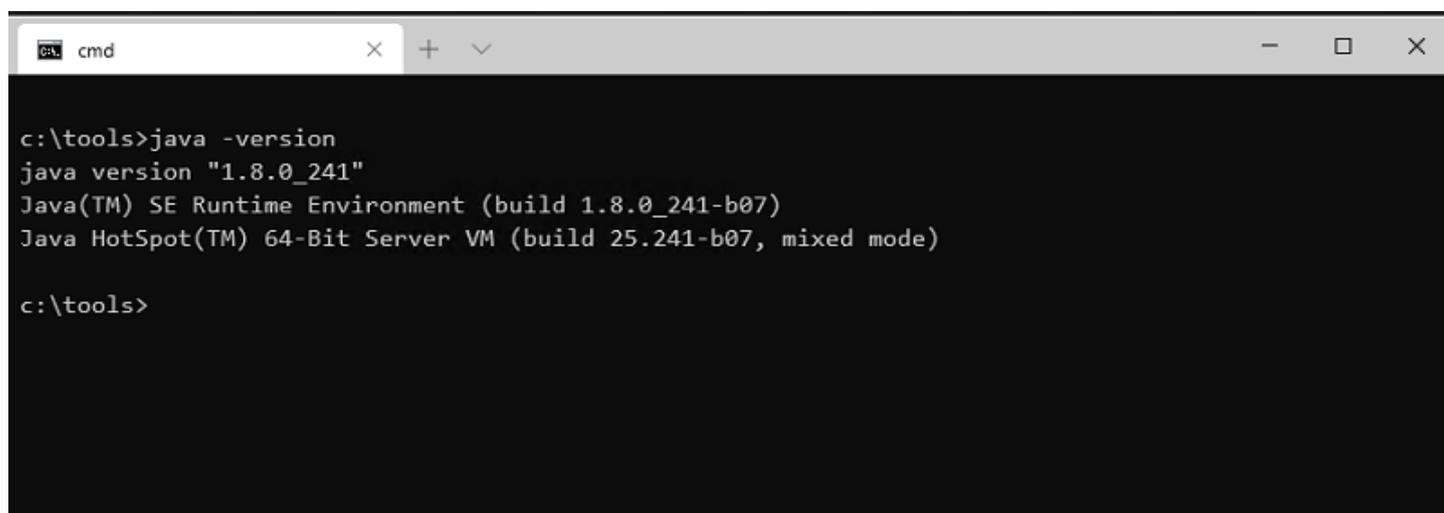
```
https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/
```

Pour l'exécution Selenium, il sera nécessaire d'avoir installé Java® dans l'appareil Windows®.

Pour vérifier si vous avez Java®, exécutez la commande suivante :

```
java -version
```

Obtenez une sortie pareille à la suivante :



```
cmd
c:\tools>java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)

c:\tools>
```

Le fichier JAR Selenium est aussi nécessaire pour exécuter le serveur localement et l'enregistrer dans votre grille. Vous pouvez l'obtenir dans :

```
https://www.selenium.dev/downloads/
```

Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users.

[Previous Releases](#)

Selenium Server (Grid)

The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).

Latest stable version [3.141.69](#)

To use the Selenium Server in a Grid configuration see the [documentation](#)

Latest Selenium 4 Alpha version [4.0.0-alpha-5](#)

The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver.

Please make sure that this is available on your \$PATH (or %PATH% on Windows) in order for the IE Driver

Download version 3.150.1 for:

[32 bit Windows IE](#) (recommended) [64 bit Windows IE](#) [CHANGELOG](#)

Pour activer le serveur Microsoft Edge®, exécutez dans un terminal dans le répertoire où vous avez le fichier JAR le suivant :

```
java -jar selenium-server-standalone-<VER>.jar -port 5555 -role node -hub
http://<ip_selenium_server>:4444/grid/register -browser
"browserName=MicrosoftEdge,
platform=WINDOWS, maxInstances=1"
```

Pour activer le serveur Internet Explorer® la commande est similaire, mais vous devez spécifier le chemin <PATH> du *driver* que vous avez téléchargé :

```
java -Dwebdriver.ie.driver=<PATH>IEDriverServer.exe -jar selenium-server-
standalone<VER>.jar -port 5555 -role node -hub
http://ip_selenium_server:4444/grid/register -browser "browserName=internet
explorer,platform=WINDOWS,maxInstances=1"
```

```
c:\Selenium
λ java -Dwebdriver.ie.driver=C:/tools/IEDriverServer.exe -jar selenium-server-standalone-3.141.59.jar
-port 5555 -role node -hub http://192.168.80.44:4444/grid/register -browser "browserName=internet explorer,platform=WINDOWS,maxInstances=1"
14:29:14.742 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
14:29:14.977 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5555
2020-04-13 14:29:16.258:INFO::main: Logging initialized @2433ms to org.seleniumhq.jetty9.util.log.StdErrLog
14:29:16.742 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
14:29:16.898 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5555
14:29:16.898 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
14:29:17.227 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
14:29:18.336 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://192.168.80.44:4444/grid/register
14:29:18.367 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
|
```

Configurez le pare-feu de Windows® pour permettre le trafic dans les ports spécifiés dans la commande d'exécution. Dans le cas des exemples antérieures : 5555 et 4444.

Enregistrement

Vous devez tenir sur compte que les enregistrements faites dans Selenium 2 peuvent ne pas fonctionner correctement.

Pour la version 3 de Selenium, ils seront compatibles les anciens enregistrements ainsi que les nouveaux, bien que ça ne garantit pas la compatibilité totale avec les versions anciennes.

Pour pouvoir enregistrer une nouvelle session, installez l'extension Selenium IDE dans le navigateur que vous souhaitez.

Firefox

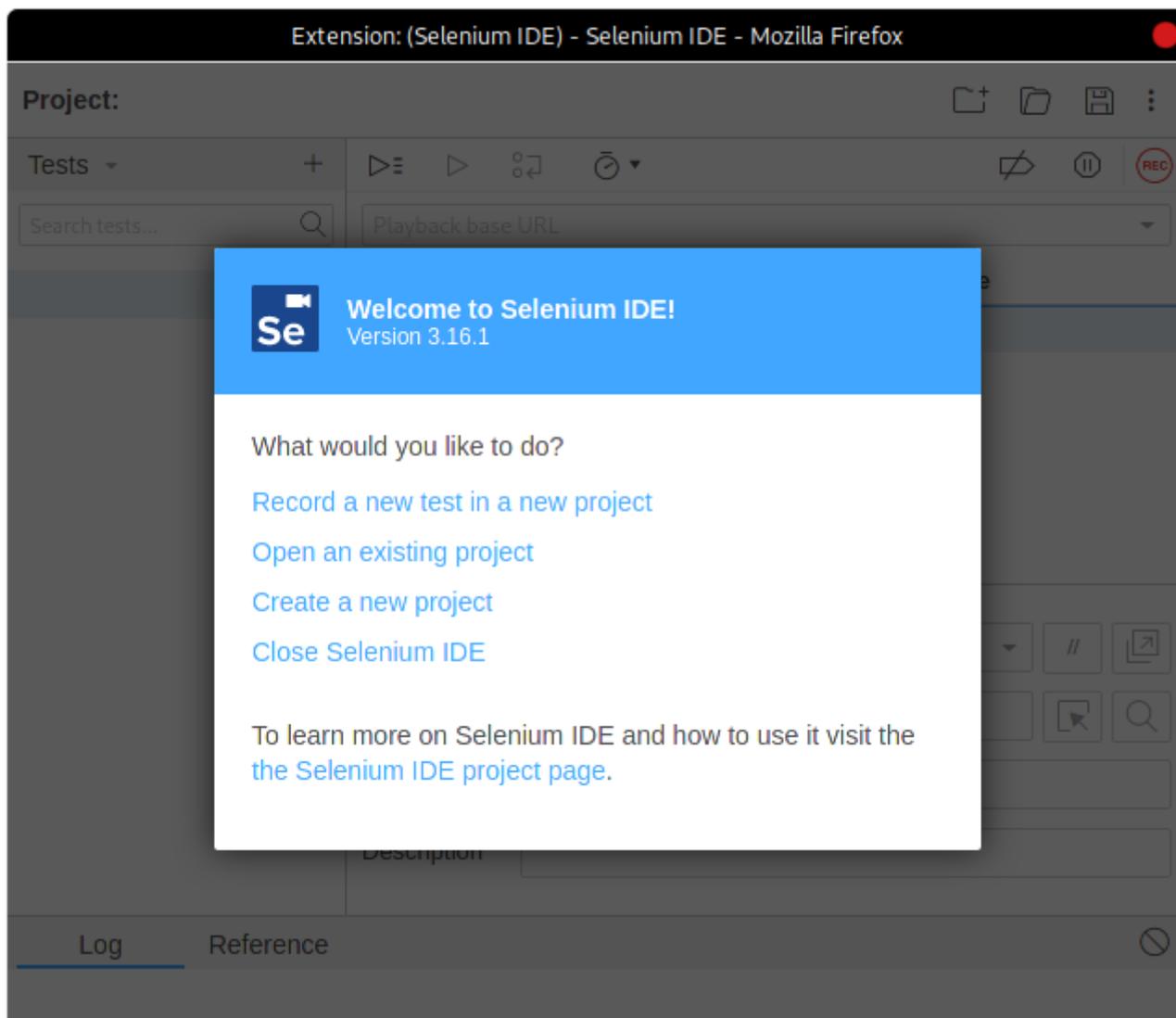
<https://addons.mozilla.org/es/firefox/addon/selenium-ide/>

Chrome

<https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd>

L'icône  sera celui qui permet d'initier l'environnement d'enregistrement une fois l'extension est installée.

Si vous cliquez, ça vous montrera un menu qui vous permet d'initier de nouveaux projets d'enregistrement.



Une fois l'enregistrement fait, vous aurez un résultat pareil à celui-ci :

Extension: (Selenium IDE) - Selenium IDE - pruebaswux - Mozilla Firefox

Project: pruebaswux

Tests +

Search tests...

http://varian.artica.lan

	Command	Target	Value
1	// phase_start:Inicio		
2	open	/web	
3	unchecked	name=check1	
4	check	name=check1	
5	// phase_end:Inicio		
6	// phase_start:Captura		
7	store attribute	xpath=//input[@name='check1']@checked	estado
8	store xpath count	xpath=//*[name='check1']	aa
9	store attribute	xpath=//input[@name='check1']@name	name
10	// storeExtraction	input name="(*?)"	inputs
11	// phase_end:Captura		
12	// phase_start:Muestra		
13	echo	\${aa}	
14	echo	\${estado}	
15	if	\${name} != "check1" && \${aa} == 1	

Command #

Target

Value

Description

Log Reference

Apart de la fonctionnalité fournie par Selenium, Pandora FMS dispose des commandes personnalisées pour extraire de l'information du cible à superviser.

Commandes

Vous pouvez trouver la liste de commandes compatibles pour les utiliser dans Pandora FMS dans [l'annexe des commandes](#).

Recommandations et suggestions lors des enregistrements

Ici nous détaillons une série de recommandations et conseils lors des enregistrements des transactions Selenium IDE ainsi que pour intégrer avec Pandora FMS les commandes les plus complexes :

- Diviser la transaction en étapes aussi que possible. Donc les modules d'état et de temps créés ainsi que les captures d'écran seront segmentés et il sera plus facile de localiser où est-ce que la transaction a échoué.
- Utiliser la commande Selenium `set speed` et `wait for` pour éviter de faux négatifs. Lors de l'exécution d'une transaction, les commandes Selenium n'ont pas de délais par défaut depuis qu'un d'entre eux finit jusqu'à l'exécution du suivant, et quelques-uns n'ont pas de temps de finalisation ou `timeout`. Cela fait que l'exécution soit exécutée très rapidement dans aussi peu de temps que possible, mais en raison d'un processus si rapide, si le web est un peu lente ou il prend un seconde de plus, il ne trouvera pas l'élément dans la nouvelle page et la vérification échouera. Par exemple, après exécuter la commande `click` et changer de page, si après vous avez une commande qui interagit avec un élément de la nouvelle page et la page prend une seconde de plus pour charger, il ne trouvera pas l'élément dans la nouvelle page et la vérification échouera. Pour éviter ces situations, il existe la commande `set speed`, qui ajoute un délai avec la quantité de millisecondes indiqués dans le cible entre chaque commande. Nous vous recommandons de l'établir au début de la transaction. Il y a aussi pour les cas où vous savez que la charge d'une page ou d'un élément peuvent se retarder, les commandes `wait for element present`, `wait for visible` et `wait for text` dans lesquelles vous pouvez établir le temps en millisecondes qu'il attendra pour que l'élément apparaisse sur la page avant de considérer la transaction échouée. Il est important de remarquer que l'utilisation de ces commandes augmente la fiabilité de la vérification il aussi augmente le temps d'exécution de la transaction.
- Faire des vérifications des éléments. Pour ça des commandes telles que `assert` et `verify`, dans ses différents versants. Finir une transaction par exemple avec un clic ne vous garantit pas que la nouvelle page que l'élément doit ouvrir soit en fait ouverte, seulement qu'il est possible de cliquer sur l'élément. Si après le clic, vous entrez `verify text` dans un texte que vous savez que vous allez seulement charger après de cliquer, ça serait en fait une manière de vérifier que la page où vous allez après cliquer est disponible.
- Utiliser `store window handle` dans les transactions dans lesquelles vous allez changer de fenêtres. Le changement de fenêtre (avec `select window`) peut échouer si vous n'avez pas stocké précédemment un identifiant dans la fenêtre initiale.
- Utiliser `xpath` lorsque le *Target* par identificateur de CSS ou lorsque vous voulez chercher du contenu dans la page. Par défaut l'enregistreur Selenium IDE introduit dans le cible (*target*) de l'élément le localisateur de CSS, mais aussi enregistre le localisateur par `xpath`, pourtant il est possible de voir tous les localisateurs qu'il garde si vous cliquez dans la boîte du cible dans l'enregistreur:

The screenshot shows the Selenium IDE interface with the following configuration:

- Command:** assert text
- Target:** |css=#\36 fn79666fiykxpysdro91x1ccr_message sp
- Value:** css=#\36
fn79666fiykxpysdro91x1c
cr_message span
- Description:** xpath=//div[@id='6fn79666fiykxpysdro91x1ccr_message']/div/h2/span
- Provided value:** xpath=//h2/span
- Test value:** xpath=//span[contains(., 'Innovadores de la monitorización')]

Aussi, en utilisant des chemins par xpath, il est possible de chercher des textes dans les étiquettes des pages pour réaliser des enregistrements beaucoup plus dynamiques. Dans la capture d'écran antérieure regardez qu'il est possible d'utiliser le texte "Innovateurs de la supervision" dans toutes les étiquettes span de la page, pas dans un localisateur particulier.

- Utilisation correcte de la commande `execute script`. Cette commande exécute une partie du code de JavaScript dans la fenêtre dans laquelle la transaction est localisée. Il est recommandée de lire le guide pour connaître son utilisation et les différents options qu'elle offre : <https://ui.vision/rpa/docs/selenium-ide/executescript>. Cependant, l'utilisation des variables stockées précédemment (par une commande `store text` par exemple) doit être entre doubles guillemets pour que le `webdriver` de Pandora FMS les interprète correctement. Ici il y a un exemple d'une variable stockée avec `store text` et son utilisation avec `execute script` de telle manière que le serveur Pandora FMS l'interprète correctement. Notez que cette utilisation de la variable entre des guillemets échouera lors de l'exécution du script dans l'enregistreur Selenium IDE :

```
store text      css=.folder:nth-child(1) abbr      test
execute script  if ("${test}".match(/(_thismonth_)(\d)+ago/)) { return 1; } else if ("${test}".match(/(_lastmonth_)(\d)+/)) { return 2; } else { return 0; } x
```

Expérience utilisateur Web (WUX) avec Selenium 2

Selenium 2 est maintenant obsolète, la version actuelle est Selenium 3.

Configurations préalables

Selenium

Déploiement avec Pandora Web Robot Daemon (PWRD)

Pandora FMS Web Robot Daemon est un service qui fournit les outils nécessaires afin d'automatiser les sessions de navigation web, en utilisant un fichier qui contient une liste d'actions nécessaires pour naviguer dans le portail web de l'utilisateur.

Il est intégré dans le serveur Pandora FMS et vous pouvez le trouver dans le dossier `/usr/share/pandora_server/util/pwr` lors de l'installation du serveur Enterprise (Linux) ou dans la librairie de modules (Windows).

Pour le télécharger allez vers le lien suivant :

<https://pandorafms.com/library/pandora-ux-and-wux-pwr-recorder/>

Il contient :

- Binaire du navigateur Firefox version 47.
- Profil préconstruit pour l'exécution de sessions de navigation web.
- Serveur d'automatisation de sessions (serveur Selenium).

Déploiement du serveur Selenium dans systèmes Windows

Exigences préalables :

- Installer Java® 1.8 dans la machine qui fournira le service.
- Installer Firefox® 47.0.1 (disponible à télécharger depuis <https://ftp.mozilla.org/pub/firefox/releases/47.0.1/>).
- Préparer un profil de Firefox® qui sera utilisé pour tester les sessions automatisés (optionel) : <http://toolsqa.com/selenium-webdriver/custom-firefox-profile/>
- Créer le répertoire suivant : `C:\PWR`.

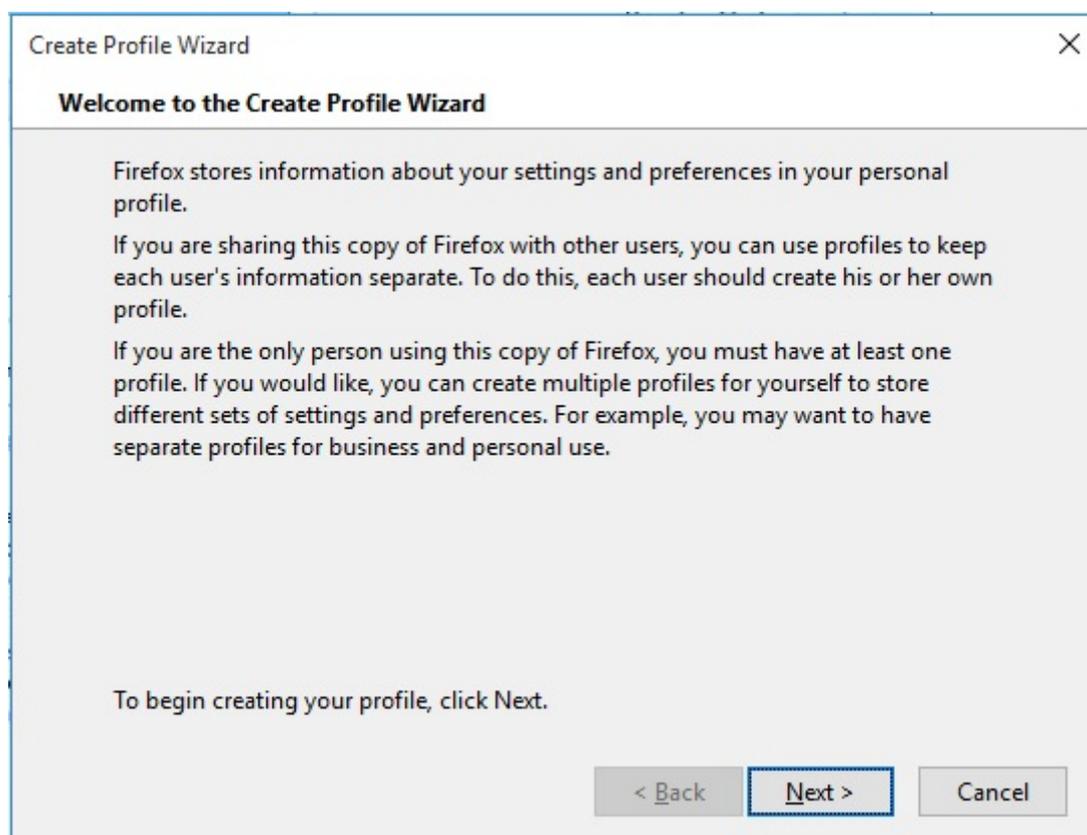
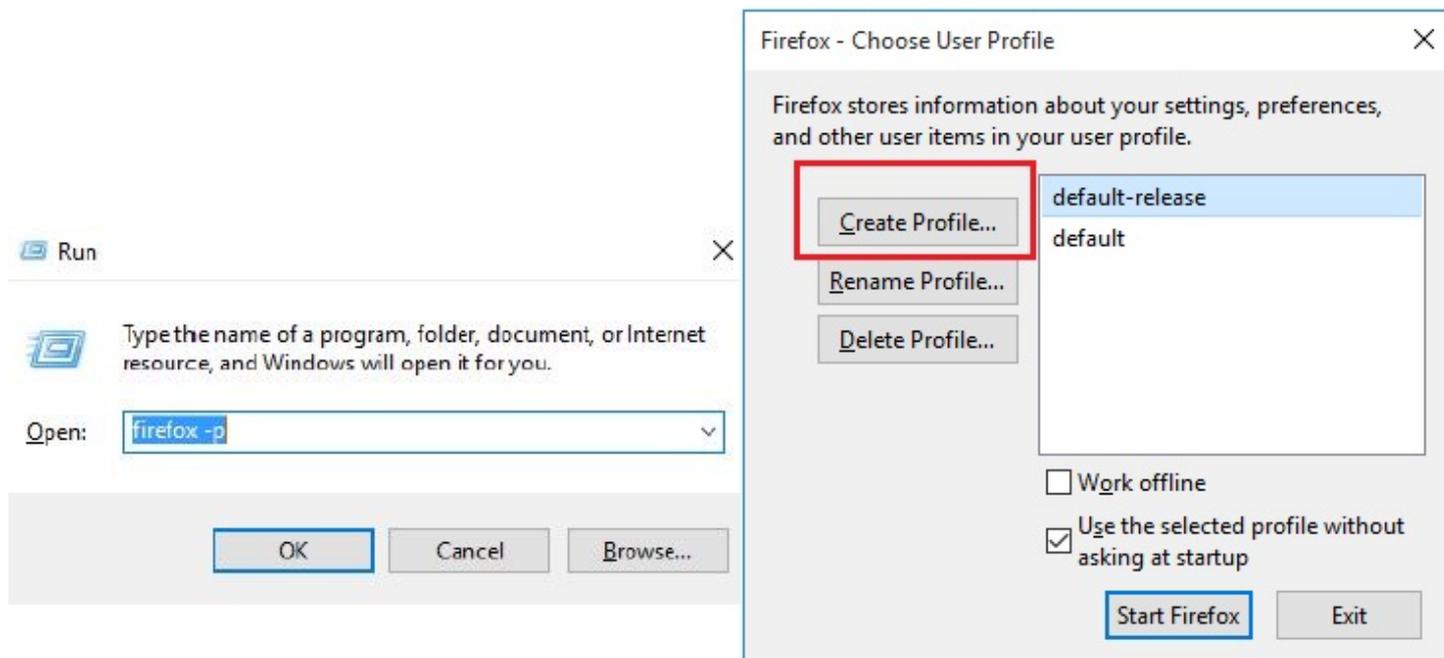
Pour télécharger le `PWR_Server.zip`, allez vers le lien suivant :

<https://pandorafms.com/library/pwr-server-for-ux-monitoring/>

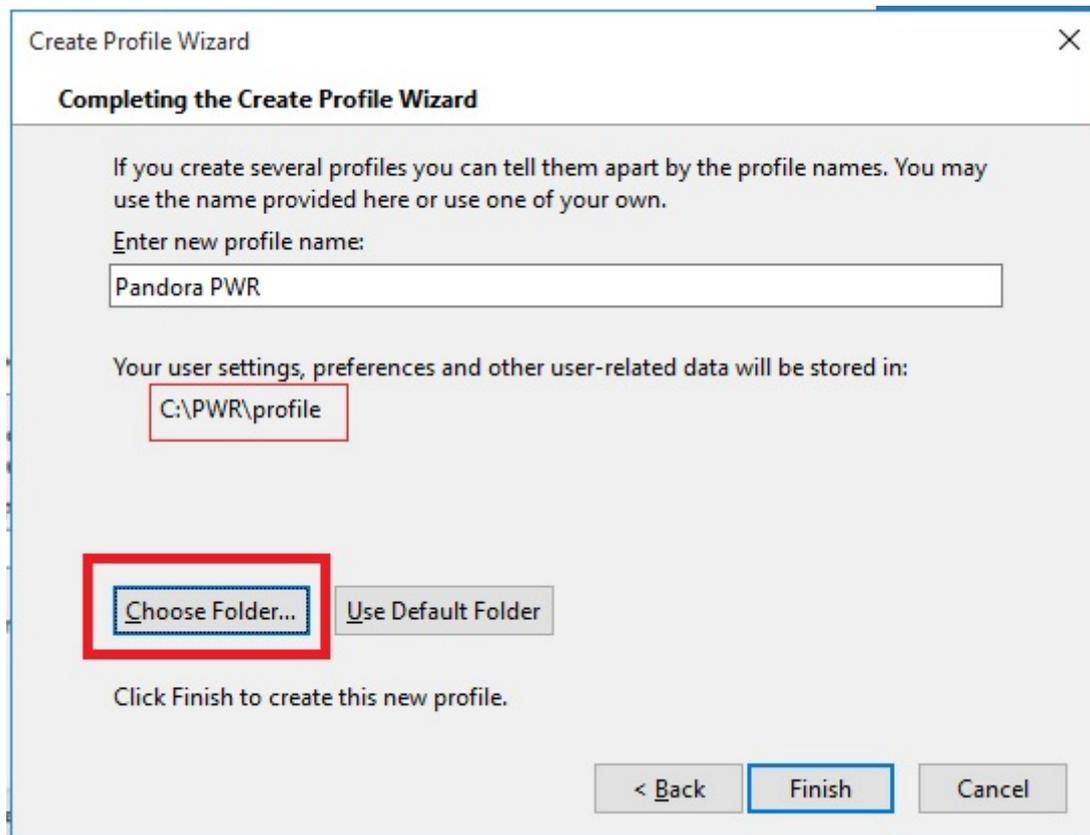
Faites la distribution de fichiers suivante :

- Décompression du fichier `PWR_Server.zip` à `C:\PWR\`.
- Exporter le profil de Firefox® vers `C:\PWR\profile`

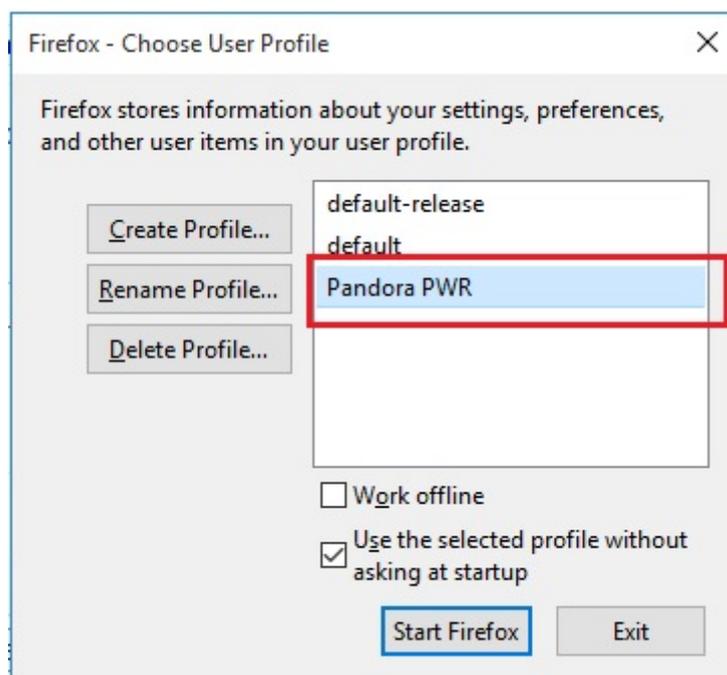
L'utilisation d'un profil pour faire des vérifications n'est pas obligatoire ; cependant il est recommandé de l'utiliser surtout si vous utilisez proxy ou lorsqu'il faut utiliser le complètement automatique des mots de passe. Pour créer un profil de Firefox® suivez les étapes des suivantes :



Choisir le répertoire de destination :



Initiez Firefox® avec le nouvel profil pour personnaliser les options telles que le proxy, l'affichage des menus contextuels ou *popups*, etc. :



Après installez le service exécutant le fichier BAT `service_installer.bat` qui a été fourni. Pour un fonctionnement correcte du `service_installer.bat` il faut modifier le contenu, en ajoutant les chemins existants dans le fichier telle que vous les avez installés dans votre machine. Par exemple, dans le cas de Java®, il pourra seulement fonctionner en tant que Java® si vous avez configuré correctement son PATH ; autrefois, sinon, il faut entrer le PATH complet dans le

fichier. Finalement, initiez le service :

```
net start PWRSRV
```

Depuis ce moment là, vous aurez le serveur Selenium en exécution dans votre machine. Cependant, vous pouvez l'exécuter manuellement (en arrêtant le service précédemment) au cas où vous voulez faire certaines tâches de dépuración où *debugging*, par le biais de la commande suivante :

```
java -jar C:\PWR\server\selenium-server-standalone-2.53.0.jar -  
firefoxProfileTemplate C:\PWR\profile -port 4444 -v
```

Déploiement du serveur Selenium dans systèmes Linux

Exigences préalables :

- Installer Java® 1.8 dans la machine qui fournira le service.
- Installer Firefox® 47.0.1 (disponible à télécharger depuis <https://ftp.mozilla.org/pub/firefox/releases/47.0.1/>).
- Préparer un profil de Firefox® qui sera utilisé pour tester les sessions automatisés (optionel) : <http://toolsqa.com/selenium-webdriver/custom-Firefox-profile/>
- Installer xorg-x11-server-xvfb.
- Installer Java®.

Le composant Selenium du démon PWRD précise Java® pour fonctionner, pour l'installer exécutez :

```
yum install Java
```

Pour que PWRD soit capable de lancer Firefox® dans le serveur GNU/Linux, il est nécessaire d'installer, en plus, gtk2 et gtk3 pour pouvoir virtualiser un environnement graphique :

```
yum install xorg-x11-server-Xvfb gtk2 gtk3
```

S'il n'est pas disponible dans les répertoires, vous pouvez trouver les fichiers .rpm dans les liens suivants :

ftp://rpmfind.net/linux/centos/6.6/os/x86_64/Packages/xorg-x11-server-Xvfb-1.15.0-22.el6.centos.x86_64.rpm

ftp://rpmfind.net/linux/centos/7.4.1708/os/x86_64/Packages/gtk2-2.24.31-1.el7.x86_64.rpm

Pour faire une installation dans les packages .rpm manuellement :

```
yum install xorg-x11-server-Xvfb-1.15.0-22.el6.centos.x86_64.rpm
yum install gtk2-2.24.31-1.el7.x86_64.rpm
```

Une fois installés les exigences préalables, continuez avec l'installation de `install_pwr.sh`. Ledit installateur se trouve par défaut dans le dossier `/usr/share/pandora_server/util/pwr/install_pwr.sh` et il s'exécute comme suit :

```
cd /usr/share/pandora_server/util/pwr/
./install_pwr.sh --install
```

Une fois installé, démarrez le service :

```
/etc/init.d/pwr start
```

Vous pouvez utiliser le *script* pour démarrez le serveur Selenium :

```
#!/bin/sh
# Monitoring selenium process
if [[ "`ps aux |grep selenium` ]]; then
    exit
else
    if [[ "`ps aux |grep Xvfb` ]]; then
        Xvfb :99 -ac &
        export DISPLAY=:99
    fi
    export DISPLAY=:99
    java -jar /usr/share/pandora_server/util/pwr/selenium-server-standalone-2.53.1.jar &
fi
```

Ou bien manuellement avec les commandes suivantes :

```
$ Xvfb :99 -ac &
-> Pulse Enter to continue
$ export DISPLAY=:99
$ java -jar /usr/share/pandora_server/util/pwr/selenium-server-standalone-2.53.1.jar -port 4444 &
```

Vous avez aussi la possibilité de faire l'installation personnalisée avec un utilisateur et un répertoire différent de l'installation par défaut.

Modes de fonctionnement de PWRD

PWRD fournit quelques modes de fonctionnement :

Standalone

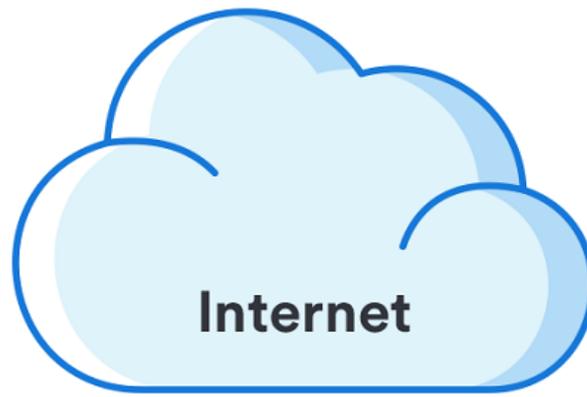
Mode standard, il démarre une seule instance de PWRD. Chaque des instances sera associée à un serveur Pandora FMS.

HUB

Mode concentrateur. Dans ce mode, le service PWRD n'évaluera pas les sessions de navigation directement, mais des "noeuds" devront être enregistrés afin d'exécuter les tâches. C'est le mode grappe ou *cluster* du service PWRD. Chaque HUB sera attribué à un serveur Pandora FMS.

PWRD en mode standalone

PWRD en mode *standalone* démarrera le démon et lui rendra prêt pour exécuter les actions indiqués par l'utilisateur à partir du serveur WUX.



WUXServer PWRD Standalone



PWRD



Pandora FMS Server
(WUXServer enabled)

Pour démarrer :

```
/etc/init.d/pwrdd start
```

Pour voir l'état :

```
/etc/init.d/pwrdd status
```

Pour arrêter :

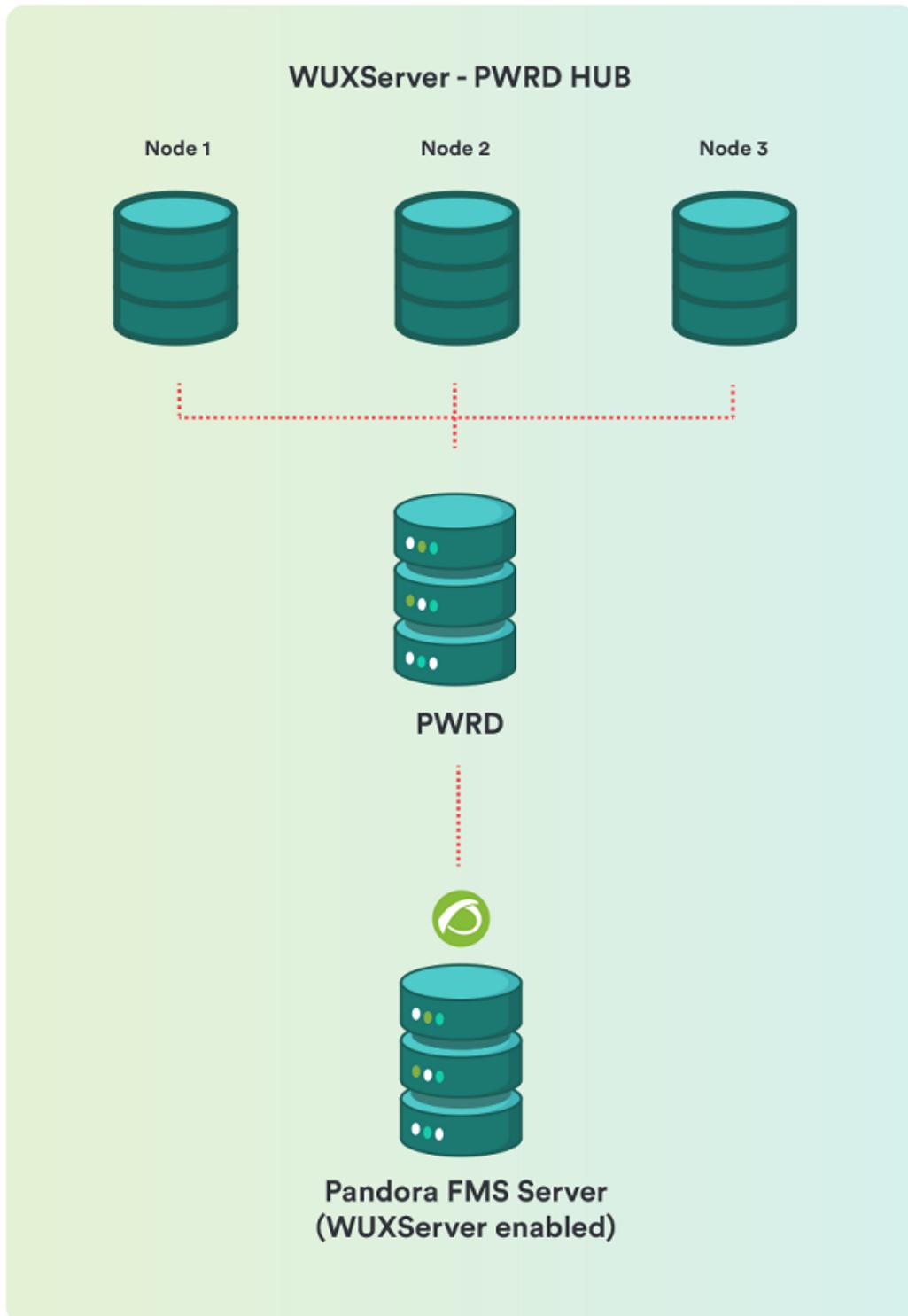
```
/etc/init.d/pwrdd stop
```

PWRD en mode HUB

Le mode concentrateur (HUB) démarrera le démon en tant qu'équilibreur de charge. Dans ce mode de travail, le système équilibrera la charge entre tous les noeuds enregistrés dedans, en attribuant l'exécution des sessions de navigation aux noeuds en fonction de leur charge de travail.

Vous pouvez voir à tout moment l'état du HUB en accédant à la console de son administration (remplacez <dir_ip_HUB> par l'adresse IP du HUB) :

```
http://<dir_ip_HUB>:4444/grid/console
```



Pour démarrez :

```
/etc/init.d/pwrld start-hub
```

Pour voir l'état :

```
/etc/init.d/pwrld status-hub
```

Pour arrêter :

```
/etc/init.d/pwrld stop-hub
```

Ajouter les noeuds PWRD à HUB

Pour ajouter un nouveau noeud PWRD vous avez besoin de :

- Un concentrateur (PWRD en mode HUB).
- Les fichiers PWRD, dans la même machine ou dans une différente.
- Connectivité TCP/4444 depuis l'appareil qui stocke le neud vers l'equipement qui stocke le HUB.

Avec ce mode de travail, le service traitera toutes les requêtes en file d'attente depuis le concentrateur (HUB), en lui retournant les résultats des exécutions. Il sera le HUB celui qui " parle " exclusivement avec le serveur WUX, étant visible pour celui qui exécute les actions de l'utilisateur.

Pour démarrer et enregistrer dans le HUB (remplacer hub par l'IP du serveur PWRD HUB) :

```
/etc/init.d/pwrld start-node http://hub:4444/grid/register
```

Pour voir l'état :

```
/etc/init.d/pwrld status-node
```

Pour arrêter :

```
/etc/init.d/pwrld stop-node
```

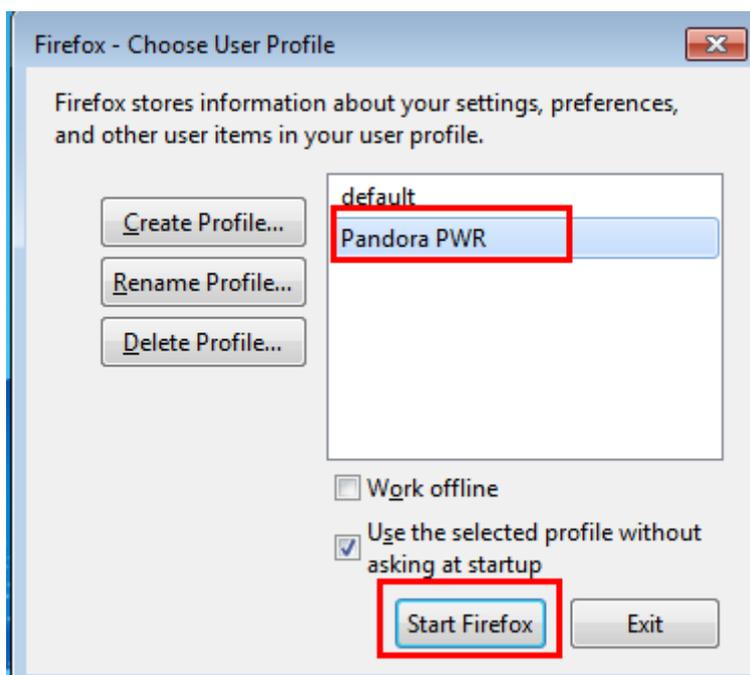
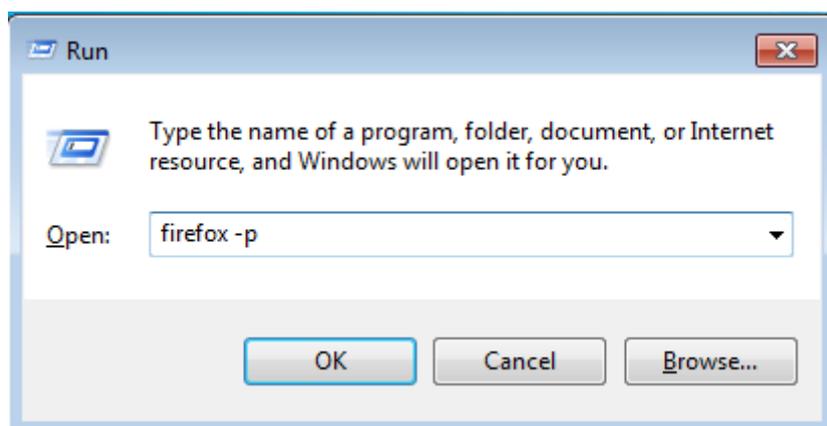
Télécharger des certificats pour Firefox

Il est possible que quelques-unes des vérifications configurées s'exécutent contre des pages web qui utilisent des certificats auto-signés ou signés par une CA non incluse entre celles qui sont acceptées par défaut dans Firefox®, donc dans ces cas il sera nécessaire de charger le certifie lui-même dans le profil Firefox® qui est utilisé.

Pour faire ça, il est plus facile de démarrer le navigateur dans un environnement graphique, accéder à l'URL et ajouter le certificat SSL. Après nous vous expliquons comment faire ça sous MS Windows® et sous GNU/Linux® :

Avec PWRD déployé dans des systèmes Windows

Dans ce cas, puisque vous comptez avec un environnement graphique, vous avez seulement à démarrer le navigateur Firefox® avec le profil utilisé pour les vérifications :



Une fois initié, accédez à l'URL avec le certificat que vous voulez charger et ajoutez-le en tant qu'exception pour le navigateur :



Your connection is not secure

The owner of artica.es has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

[Learn more...](#)

Report errors like this to help Mozilla identify and block malicious sites

Go Back

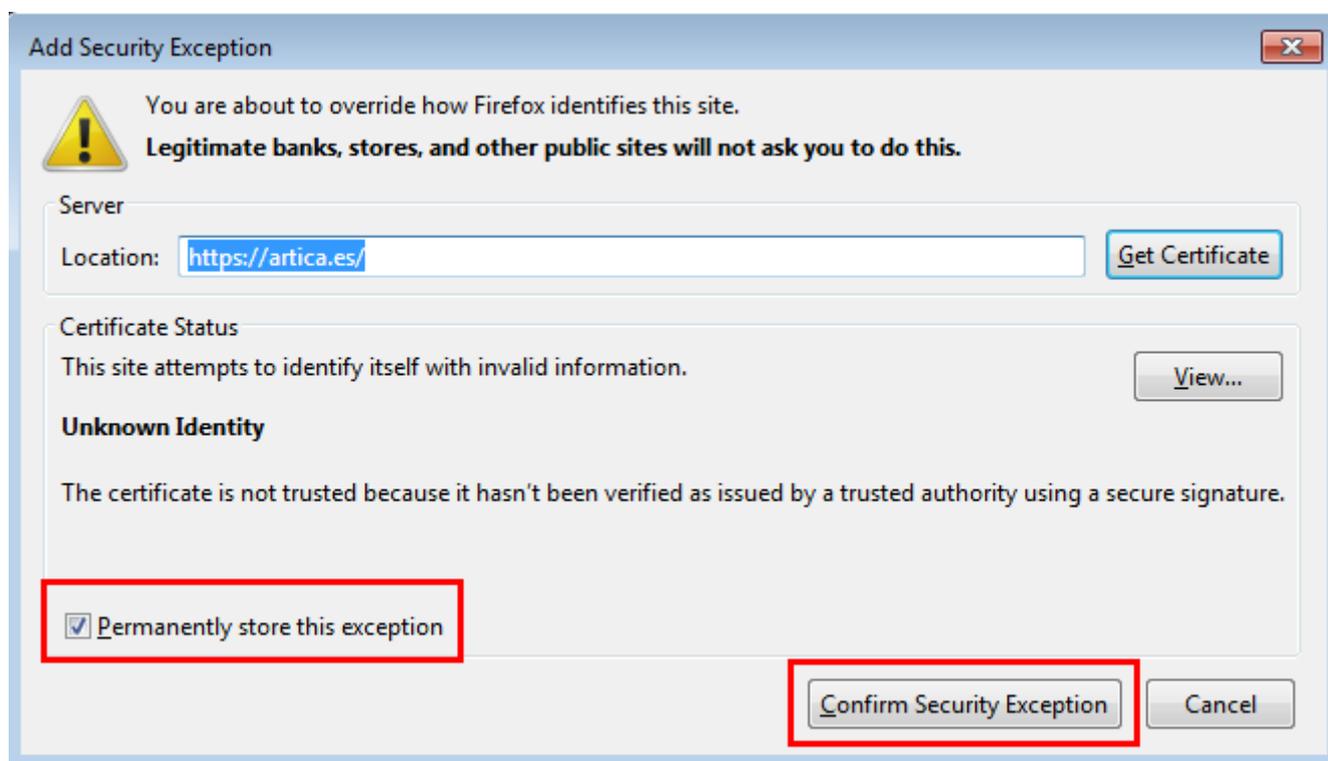
Advanced

artica.es uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is unknown.
The server might not be sending the appropriate intermediate certificates.
An additional root certificate may need to be imported.

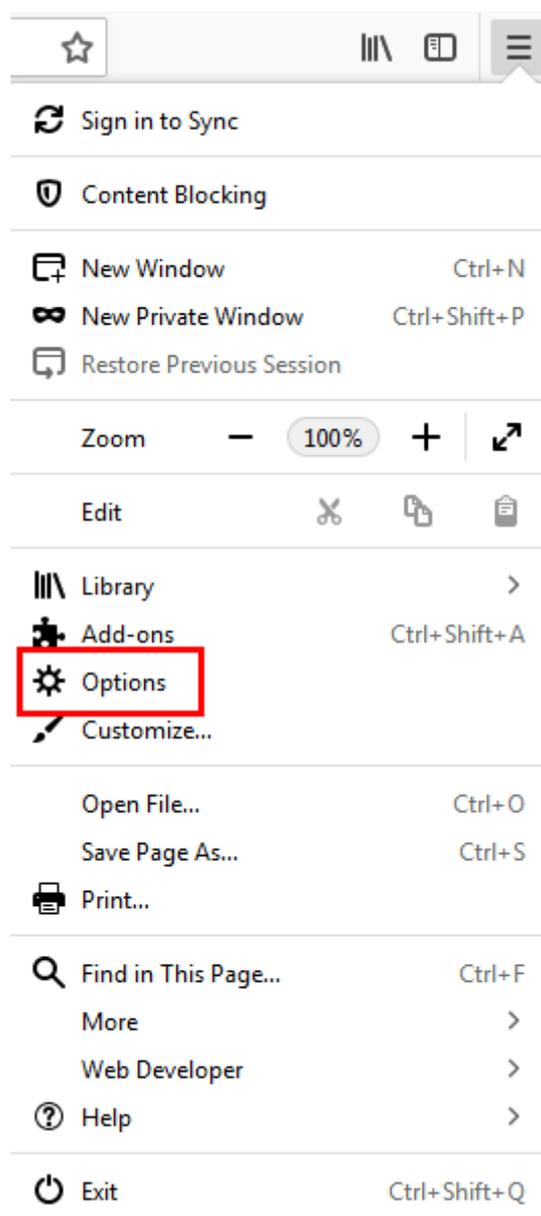
Error code: [SEC_ERROR_UNKNOWN_ISSUER](#)

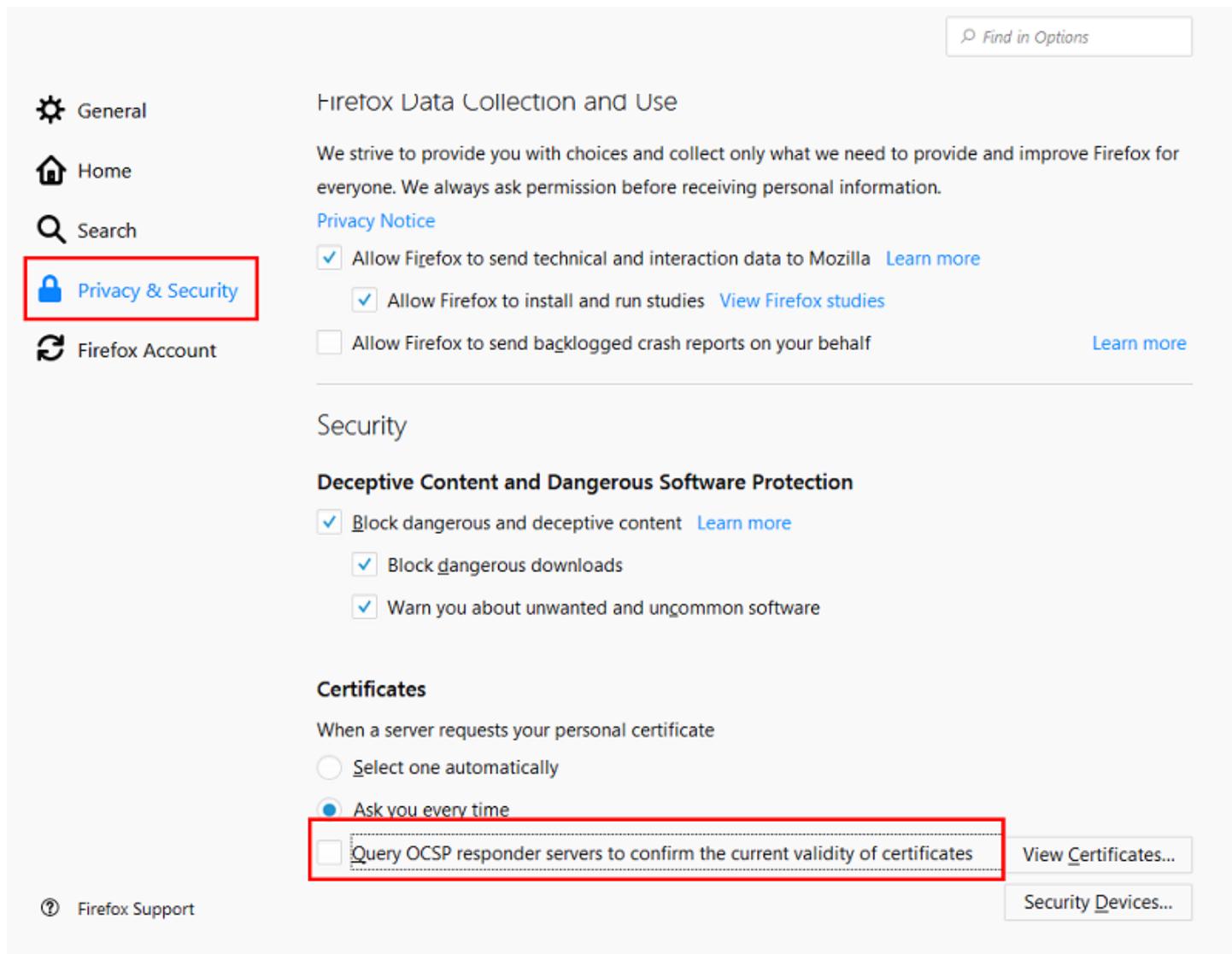
Add Exception...



Une autre possibilité, si vous voulez accepter n'importe quel type de certificat SSL et tenant sur compte qu'il est un environnement de test et/ou supervision, serait d'accéder aux options de Firefox®, et l'onglet de " Privac   et s  curit   ", et d  s  lectionner le ch  mp " Consulter aux

serveurs qui répondent OCSP pour confirmer la validité actuelle des certificats ” :





Firefox Data Collection and Use

We strive to provide you with choices and collect only what we need to provide and improve Firefox for everyone. We always ask permission before receiving personal information.

[Privacy Notice](#)

- Allow Firefox to send technical and interaction data to Mozilla [Learn more](#)
- Allow Firefox to install and run studies [View Firefox studies](#)
- Allow Firefox to send backlogged crash reports on your behalf [Learn more](#)

Security

Deceptive Content and Dangerous Software Protection

- Block dangerous and deceptive content [Learn more](#)
 - Block dangerous downloads
 - Warn you about unwanted and uncommon software

Certificates

When a server requests your personal certificate

- Select one automatically
- Ask you every time
- Query OCSP responder servers to confirm the current validity of certificates [View Certificates...](#)

[Security Devices...](#)

Firefox Support

Avec PWRD déployé dans des systèmes Linux

En général, lorsque vous installez un serveur GNU/Linux®, un environnement de bureau graphique n'est pas inclu, donc pour pouvoir suivre les m^mes étapes que précédement il faut faire un redirection des graphiques X vers un équipement qui aie un environnement de bureau graphique. Cela se fait de différents manières si vous redirigez les X vers un bureau graphique dans GNU/Linux ou MS Windows®.

Rédirigeand X11 vers un bureau Linux

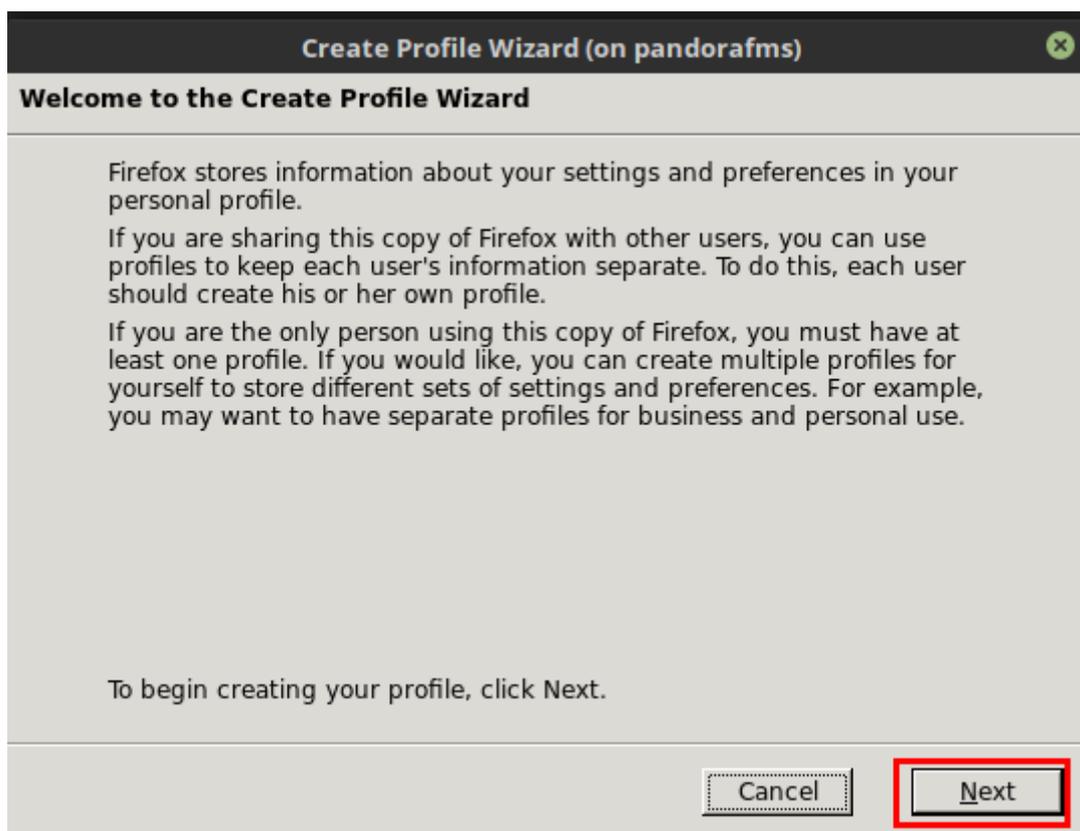
Établissez une connexion SSH au serveur PWRD avec le paramètre “ -X ” :

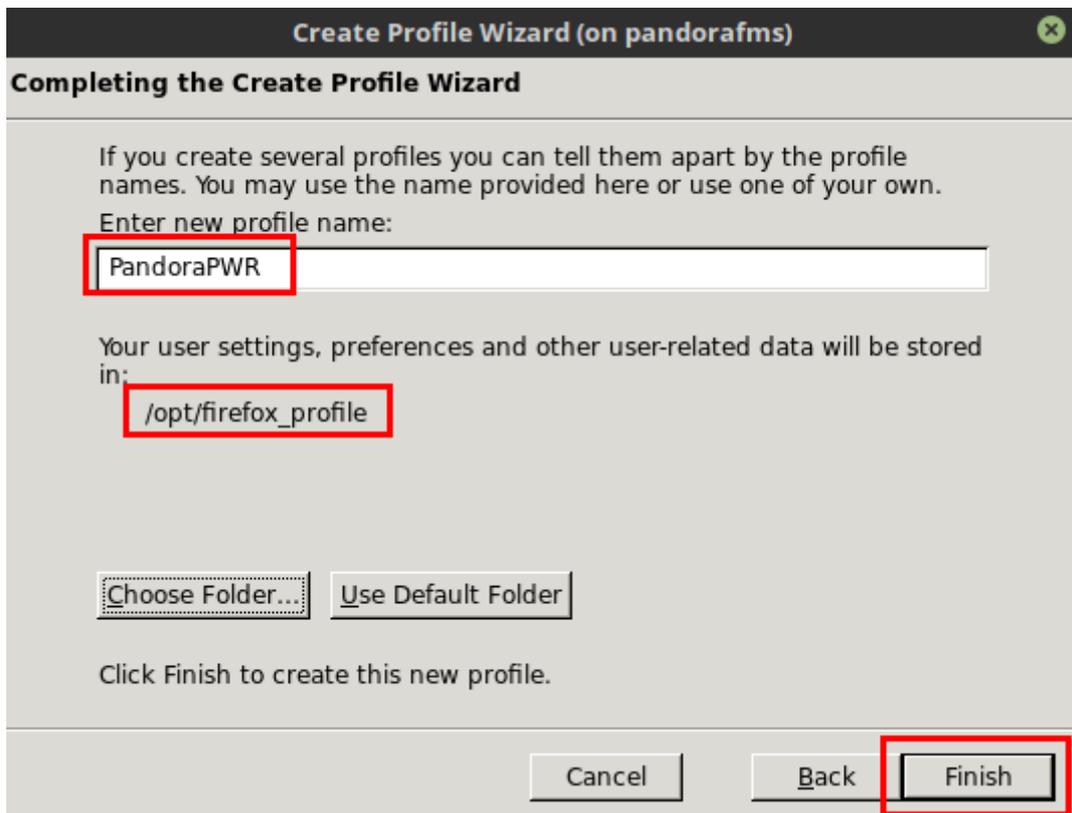
```
ssh -X user@pwr_d_ip_address
```

Cela fait, lorsque vous lancez le navigateur Firefox, vous pourrez voir comment ça est affiché sur le bureau :

```
firefox -p
```

Dans une installation par défaut vous verrez seulement le profil " default ", donc il est recommandable de faire pareille à ce que vous avez fait pour l'installation dans MS Windows® et créer un nouveau profil à utiliser :





Si vous sauvegardez les profil dans un chemin différent de /opt/firefox_profile il est nécessaire d'éditer le *script* de démarrage du PWRD /etc/init.d/pwrdd pour indiquer un nouvel chemin au nouveau profil :

```
PWROPTS=" -firefoxProfileTemplate /path/to/profile/folder"
```

Le profil créé, démarrez le navigateur :



Une fois initié, accédez à l'URL avec le certificat que vous voulez télécharger et ajouter en tant qu'exception pour le navigateur :



This Connection is Untrusted

You have asked Firefox to connect securely to **artica.es**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

▶ Technical Details

▼ I Understand the Risks

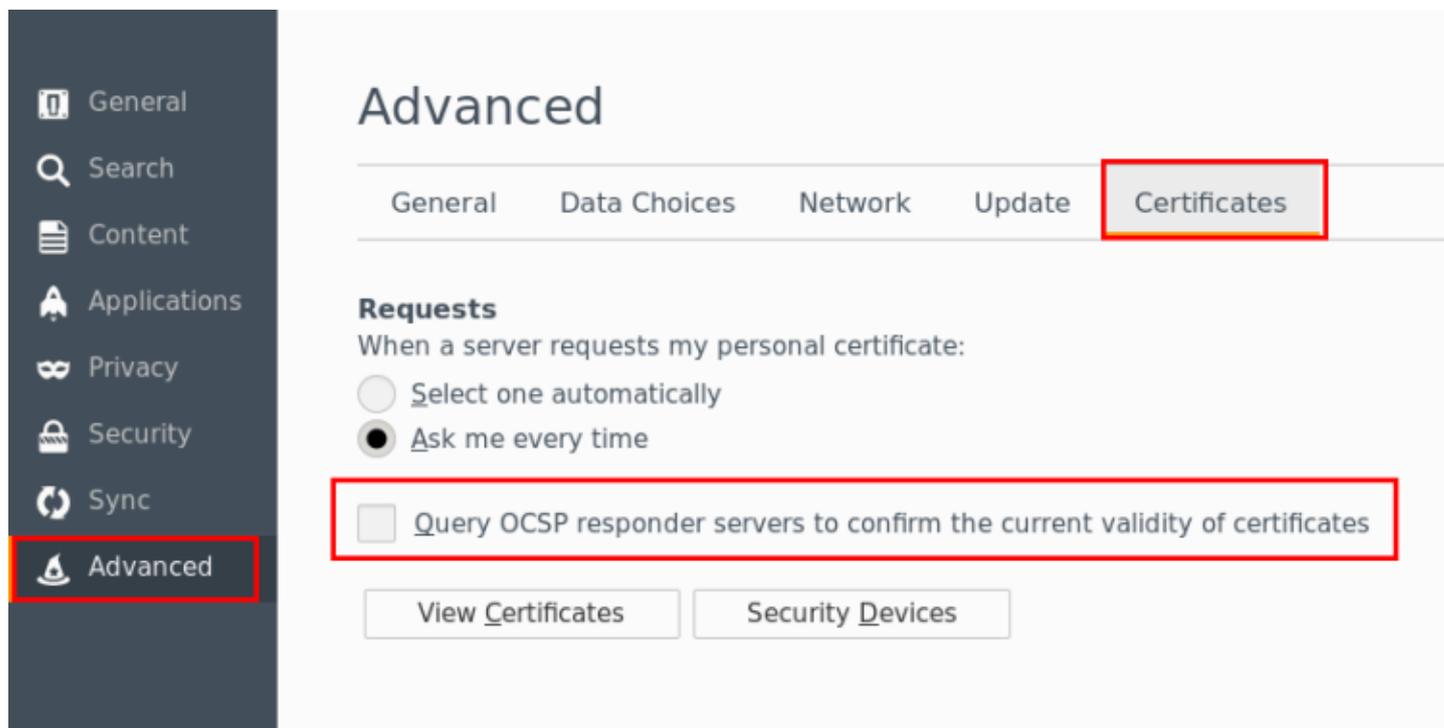
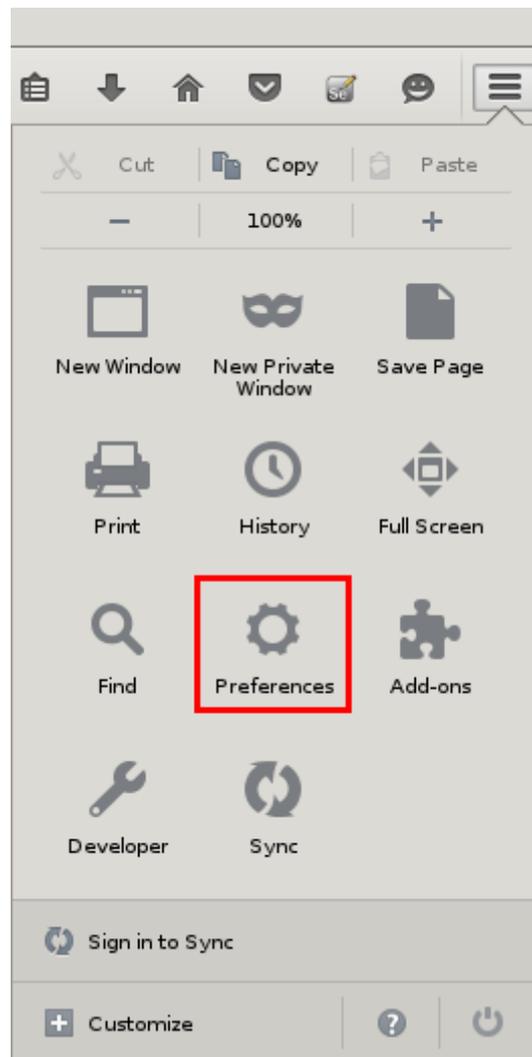
If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...



Une autre possibilité si vous voulez accepter n'importe quel certificat SSL et gardant à l'esprit qu'il est un environnement de test et/ou supervision, il serait d'accéder aux options de Firefox, à l'onglet de " Avancé > Certificats ", et désélectionner le champ " Consulter aux serveurs qui répondent OCSP pour vérifier la validité actuelle des certificats " :

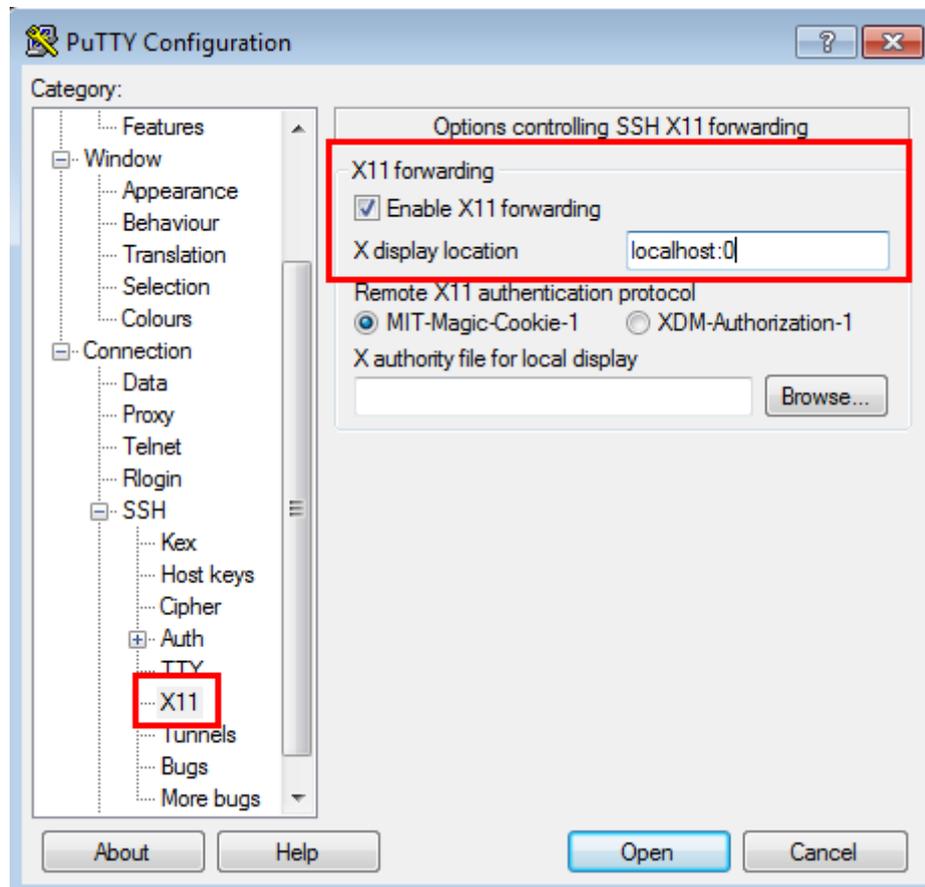


Rédirigeant X11 vers un bureau Windows

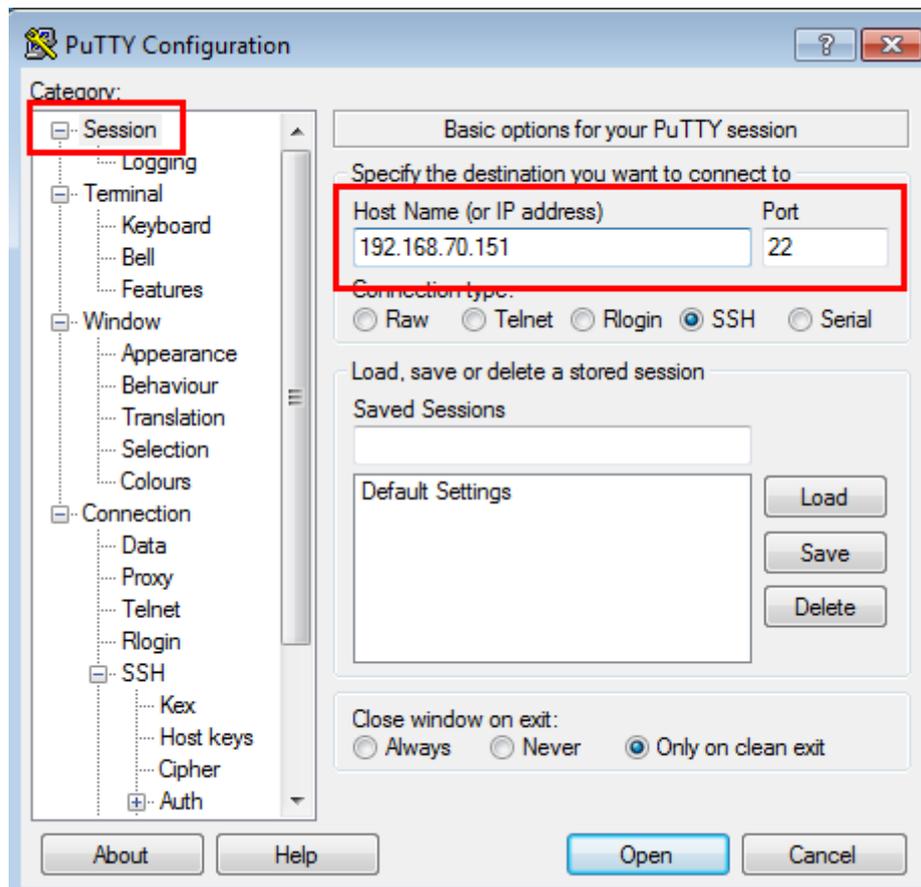
Dans le cas de Windows, au début installez un serveur de X pour pouvoir faire la redirection,

comme par exemple Xming. Une fois installé et initié, vous pouvez faire la redirection des X.

En utilisant le client SSH **Putty**, allez vers la section " Connexion > SSH > X11 " avant de faire la connexion, et assurez-vous de sélectionner l'option " Enable X11 forwarding " ainsi que de remplir le champ " X display location " en tant que " localhost:0 " :



Après retournez vers la section " Session " et établissez la connexion :

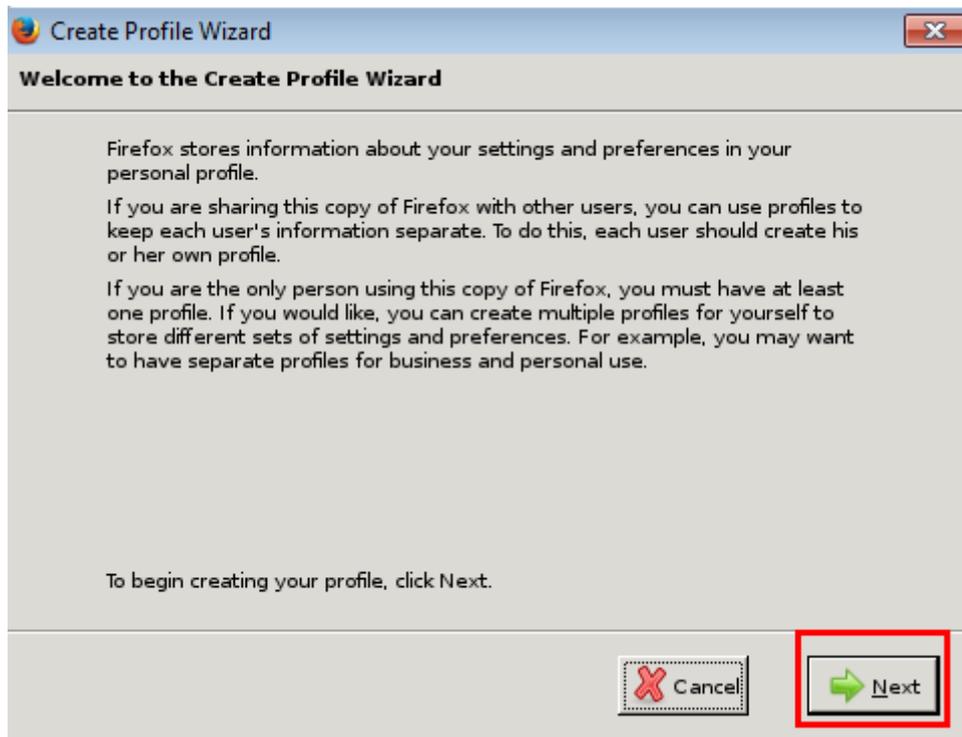


Une fois connecté, si vous lancez le navigateur Firefox® dans le serveur PWRD, vous pourrez l'afficher sur le bureau MS Windows®.

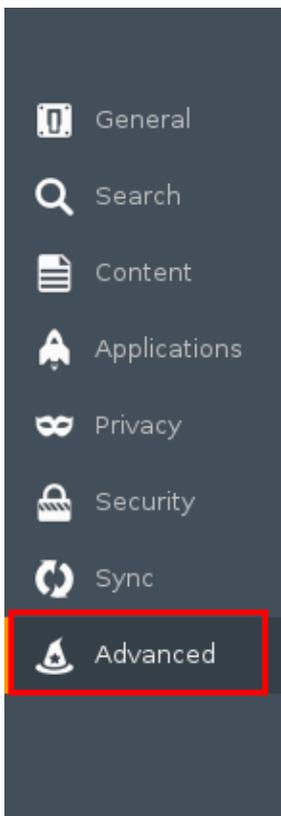
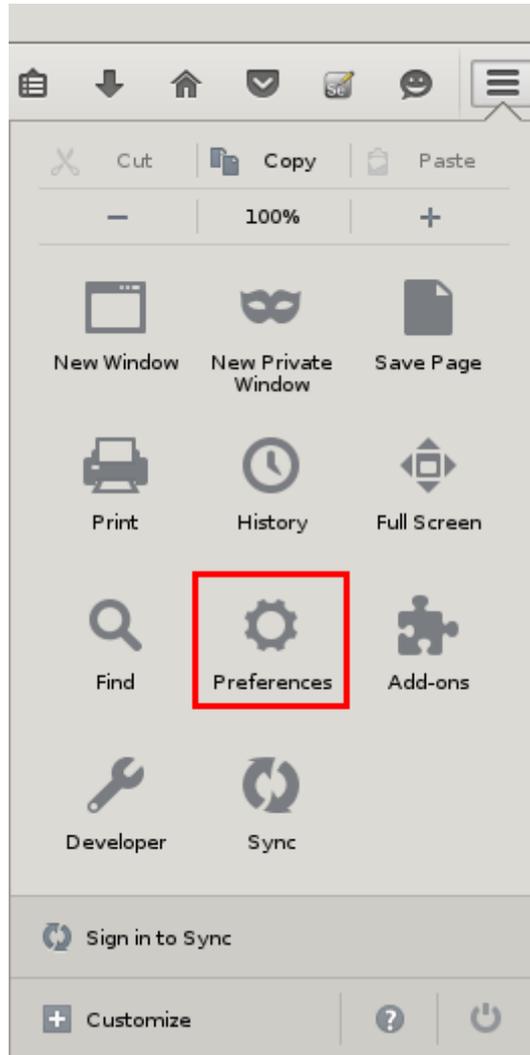
```
firefox -p
```

Dans une installation par défaut vous verrez seulement le profil " default ", donc il serait recommandable de faire de même que dans l'installation dans Windows® et créer un nouveau profil dans lequel utiliser :

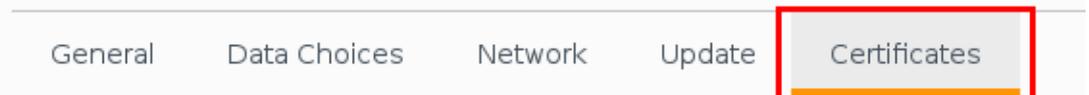




si vous sauvegardez le profil dans un chemin différent de `/opt/firefox_profile`, il sera nécessaire d'éditer le *script* de démarrage du PWRD `/etc/init.d/pwrdd Certificats`, et désélectionner le champ " Consulter les serveurs qui répondent OCSP pour confirmer la validité actuelle des certificats " :



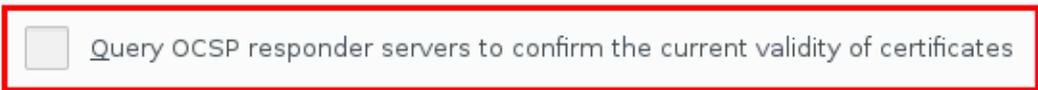
Advanced



Requests

When a server requests my personal certificate:

- Select one automatically
- Ask me every time



Configuration du serveur PFMS

Pour l'utilisation du mode centralisé (WUX) il est nécessaire d'appliquer la configuration suivante au serveur Pandora FMS.

Quel que soit le mode choisi, une fois démarré vous pouvez commencer à attribuer des exécutions de vos sessions de navigation, en ajoutant les paramètres de configuration du serveur WUX au fichier de configuration du serveur Pandora FMS.

Supposant que vous avez déployé le serveur PWRD dans le même serveur dans lequel le serveur Pandora FMS s'exécute, il ajouterait la configuration suivante dans `/etc/pandora/pandora_server.conf` :

```
wuxserver 1
wux_host 127.0.0.1
wux_port 4444
wux_timeout 30
```

Le paramètre `wux_timeout` établit le temps maximale des transactions jusqu'à 30 secondes : s'il est nécessaire, modifiez cette valeur selon votre environnement particulier.

Gestion des fils du PFMS

En cas d'utilisation du PWRD en mode *hub* :

- La gestion des threads de `wuxserver` se fait automatiquement lors du démarrage du service `pandora_server`.
- Elle est effectuée en tenant compte du nombre de nœuds d'un navigateur MINOR dans le hub Selenium. Par exemple :
 - Si 2 nœuds Firefox et 2 nœuds Chrome sont configurés dans le *hub*, le nombre de threads `wuxserver` sera 2.
 - Si 1 nœud Firefox et 4 nœuds Chrome sont configurés dans le *hub*, le nombre de threads sera 1.
 - Si 6 nœuds Firefox sont configurés dans le *hub*, le nombre de threads sera 6.

Notez que chaque thread indique les sessions qui peuvent être envoyées simultanément du `wuxserver` au *hub* Selenium.

Enregistreur de sessions (PWR)

La nouvelle version de Selenium IDE est compatible à partir de la version 745 de Pandora FMS (Selenium 3). Les transactions enregistrées dans Selenium 2 ne seront pas

affectées.

Avant de superviser un expérience utilisateur, faites l'enregistrement. Selon le type de technologie que vous avez choisi il utilisera un système d'enregistrement ou un autre.

Les sessions enregistrées avec Selenium 2 seulement pourront se faire avec Firefox®.

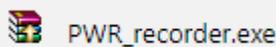
Pour faire l'enregistrement d'une navigation avec PWR l'enregistreur PWRD disponible dans la librairie de modules est nécessaire :

<https://pandorafms.com/library/pandora-ux-and-wux-pwr-recorder/>

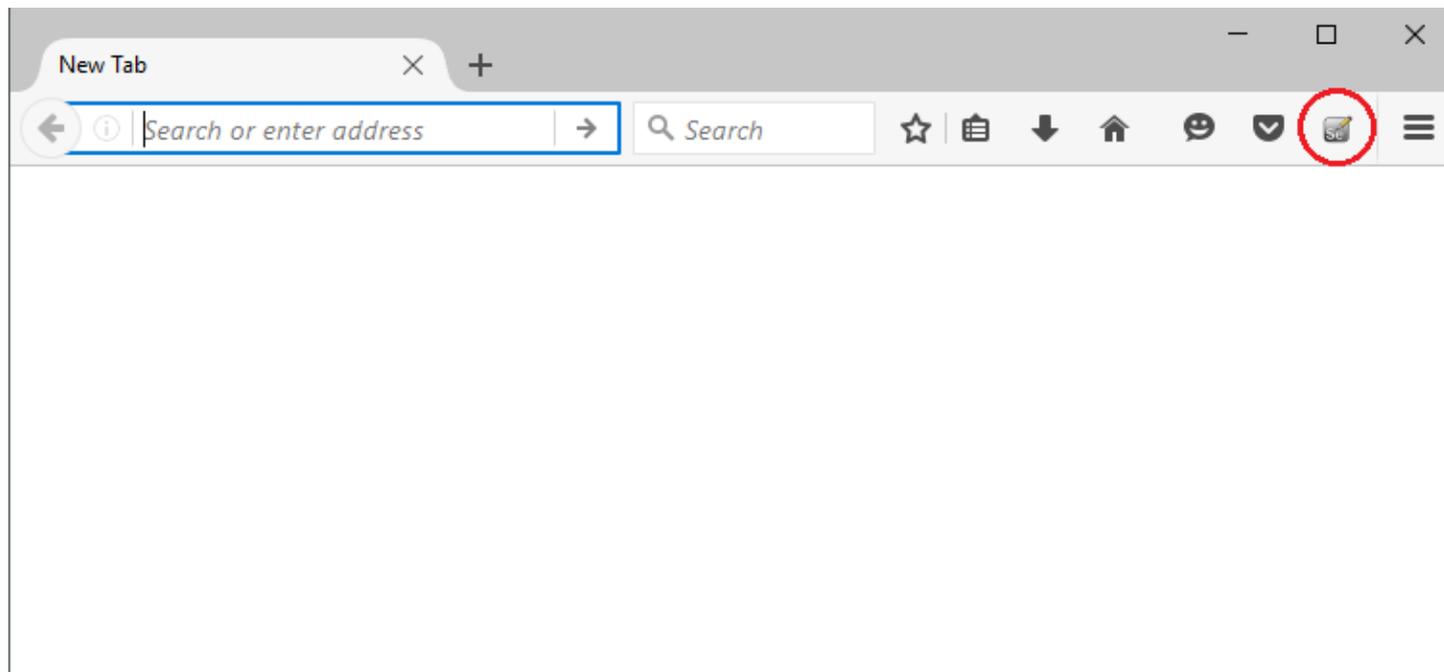
Il contient :

- Navigateur web [Firefox versión 47.0.1](#).
- Extension [Selenium IDE](#).

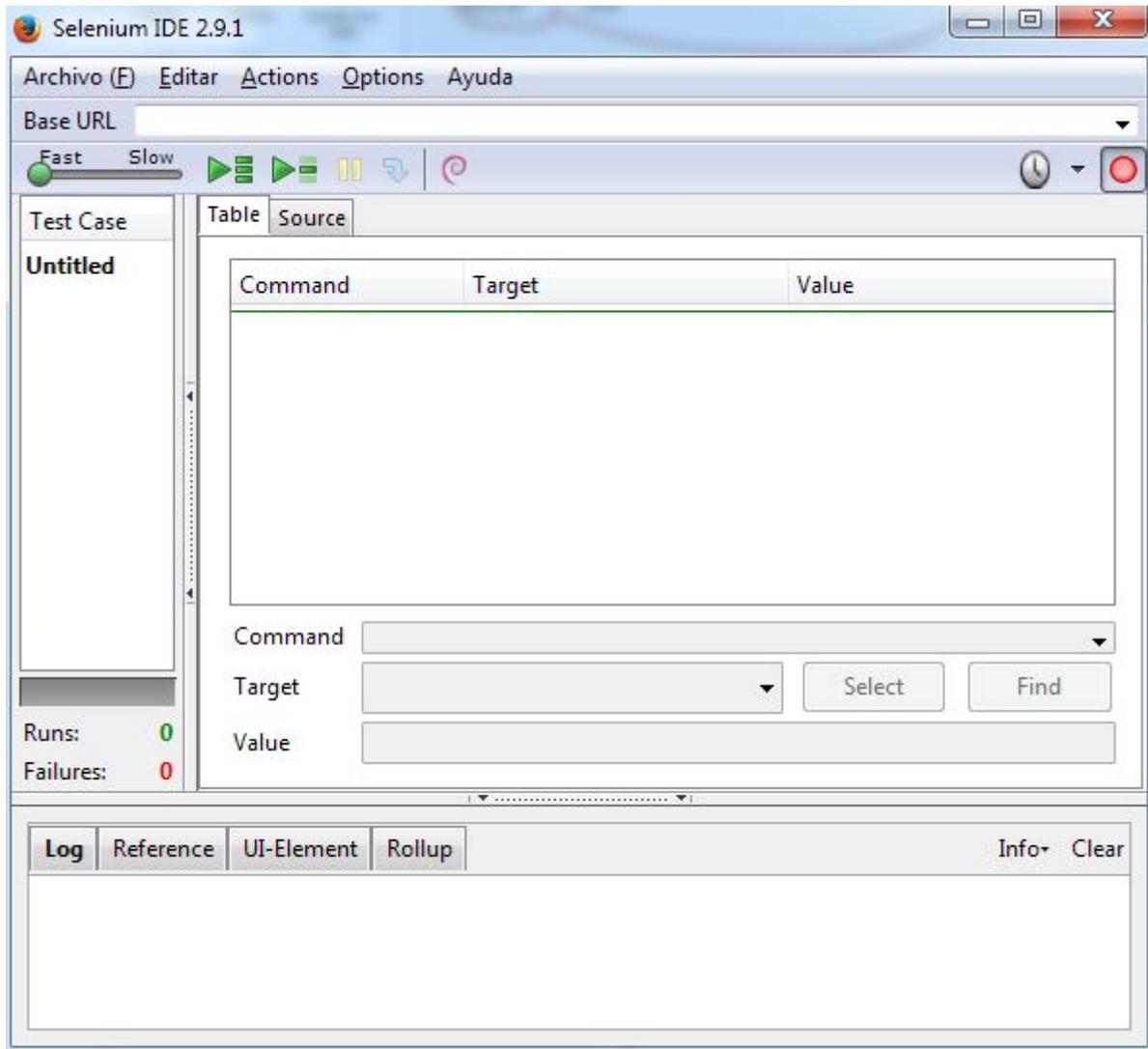
Démarrez l'enregistreur PWR_recorder.exe



Firefox démarrera automatiquement avec l'environnement prêt à enregistrer des sessions PWR :



Après accéder à l'IDE de Selenium, vous pouvez initier l'enregistrement de l'expérience utilisateur comme suit :

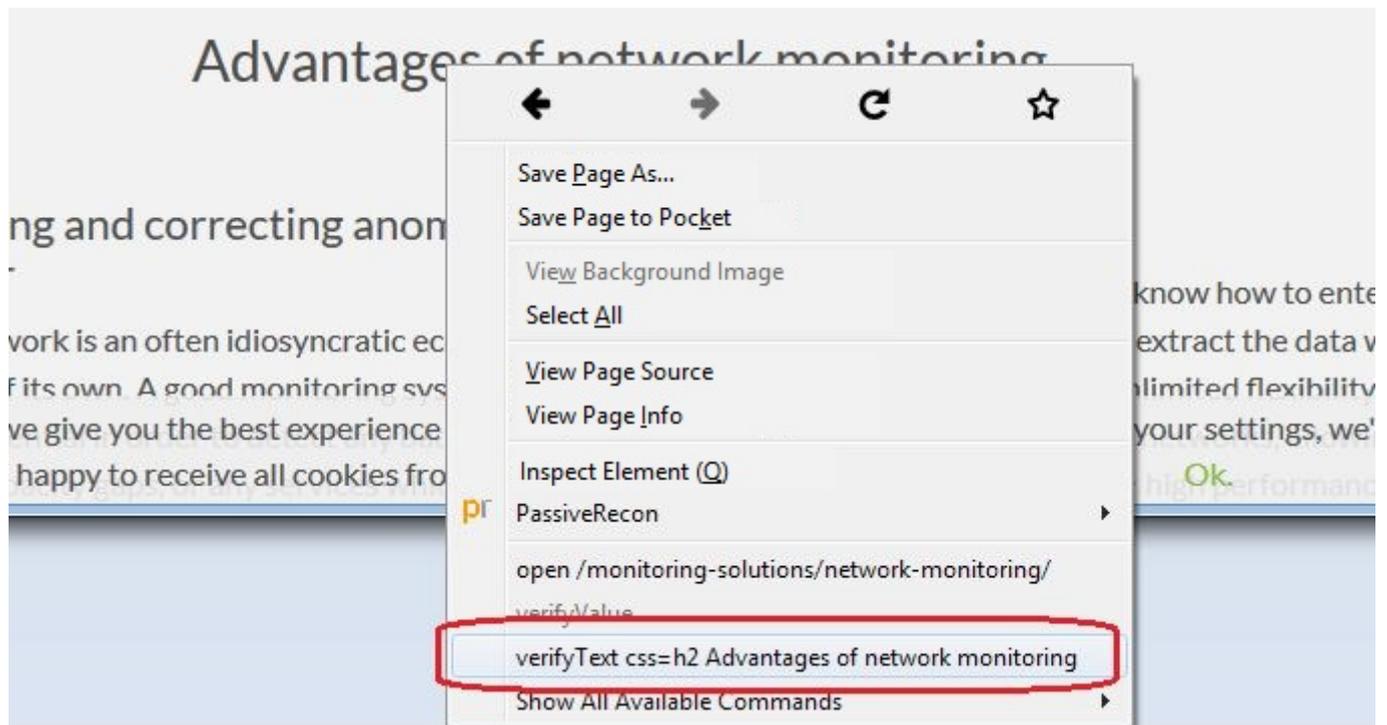


Naviguez par le site web à superviser ; les différents actinos de chaque étape finie seront affichés sur l'enregistreur.

Afin d'arrêter l'enregistrement, utilisez le bouton suivant, situé sur la coin superieur droit de l'enregistreur :



Une fois complétées les actions, faites les vérifications sur la page ; par exemple, vérifier l'existence d'un texte déterminé pour assurer que la page chargée est la correcte. Pour ça, cliquez sur le bouton droit de la souris sur une section de texte dans la fenêtre du navigateur pendant l'enregistrement, et sélectionnez l'option *verifyText* :



Une nouvelle étape dans l'enregistreur en indiquant l'action de vérification de texte indiquée :

Command	Target	Value
open	/monitoring-software/	
clickAndWait	xpath= (//img[@alt='Pandora FMS'])[2]	
clickAndWait	xpath= (//a[contains(text(),'Network Monitoring')])[2]	
verifyText	css=h2	Advantages of network monitoring

Command: verifyText

Target: css=h2

Value: Advantages of network monitoring

Réproduisez la séquence complète par le biais du bouton Play entire test suite et vérifiez qu'il fini correctement :

Archivo (F) Editar Actions Options Ayuda

Base URL https://pandorafms.com/

Fast Slow

Test Case

Untitled 5 *

Table Source

Command	Target	Value
open	/monitoring-software/	
clickAndWait	xpath=//img[@alt='Pandora FMS']][2]	
clickAndWait	xpath=//a[contains(text(),'Network Monitoring')][2]	
verifyText	css=h2	Advantages of network monitoring

Command

Target

Value

Select Find

Runs: 1

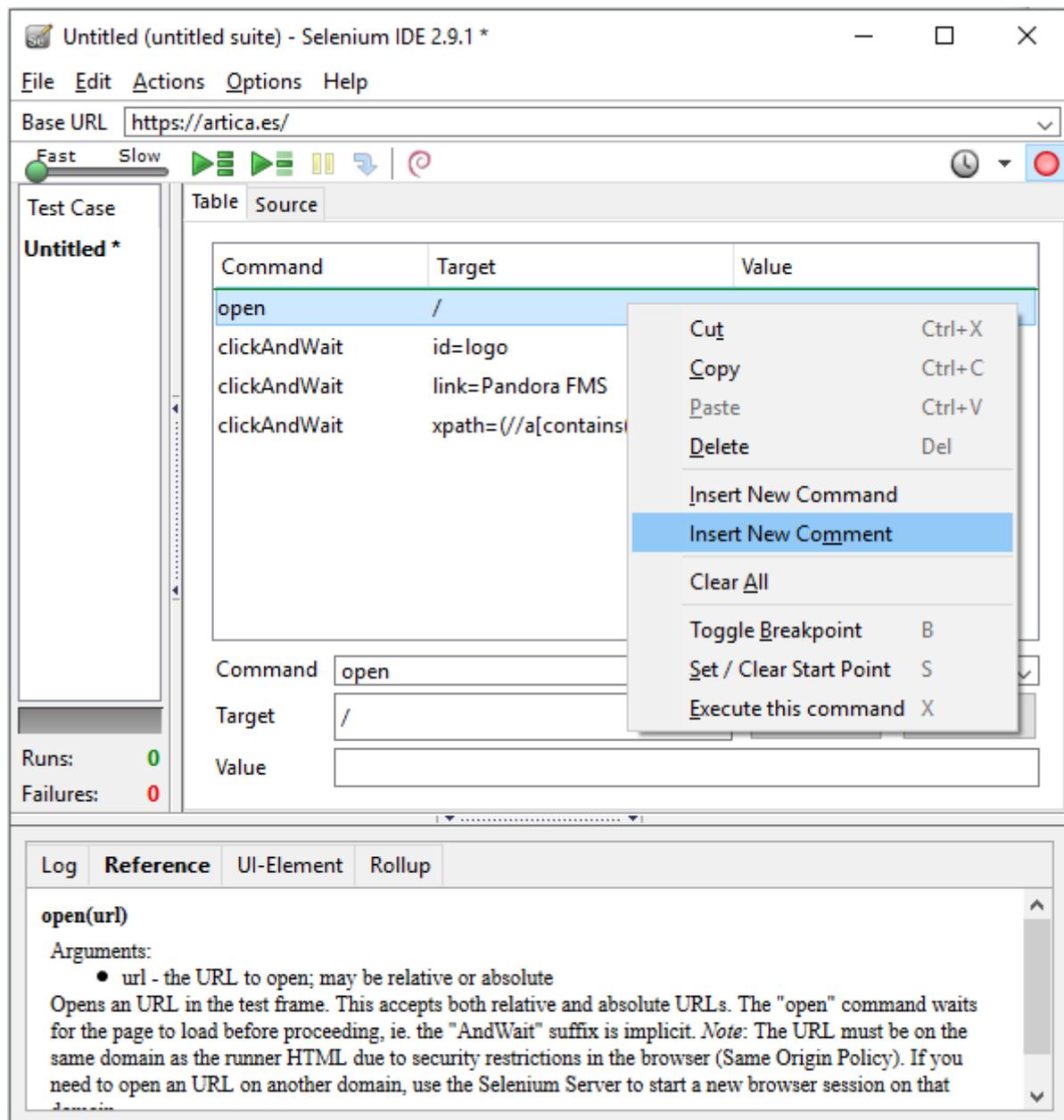
Failures: 0

Une fois vérifié la validité de la séquence de navigation, gardez (Archive → Save Test Case) pour l'exécuter après avec Pandora FMS UX. Le fichier résultant sera un document HTML que Pandora FMS UX interprétera.

Enregistrer une session transactionnelle pour Pandora WUX

Pandora FMS WUX permet de diviser la supervision de la navigation d'un site web en quelques modules, qui représenteront chacune des étapes réalisées.

Pour insérer un nouveau point de contrôle et générer des modules de phase (jusqu'à ce point là) cliquez sur le bouton droit de la souris sur le point où vous souhaitez identifier le départ de la phase.



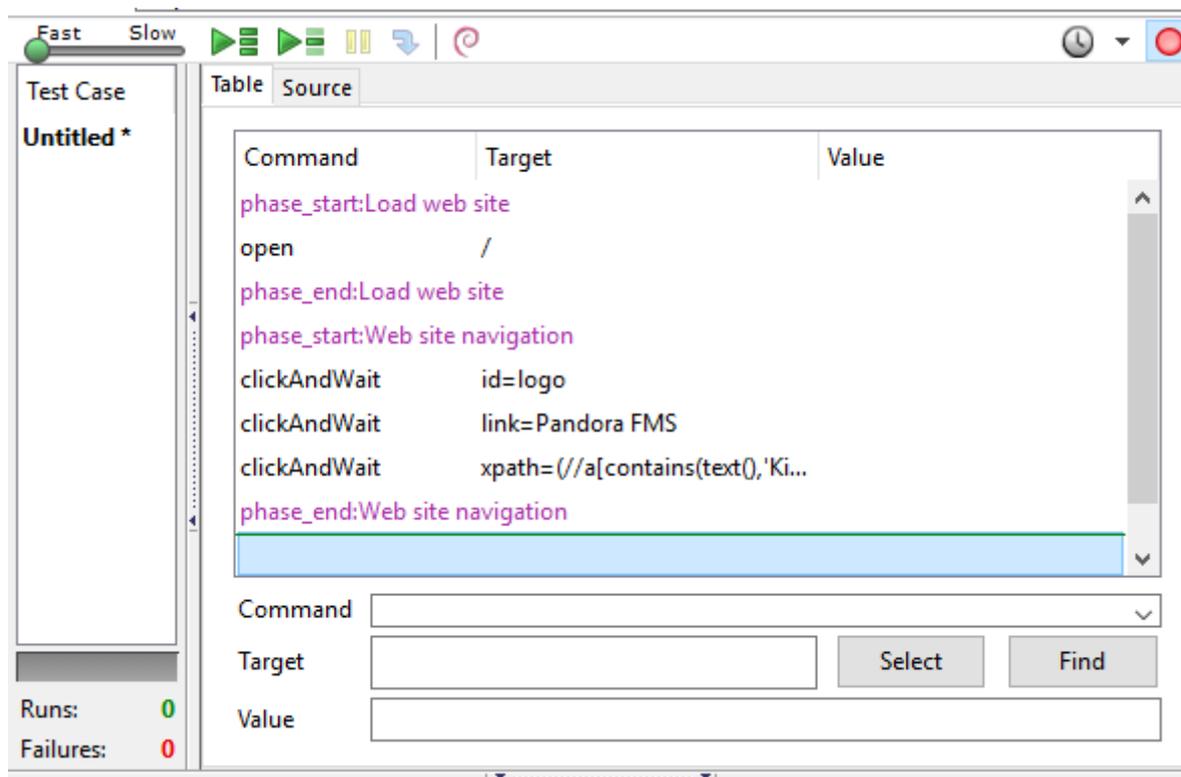
En tant que commentaire, ajoutez le texte suivant :

```
phase_start:<phase_name>
```

L'étape comprise le temps et le résultat de toutes les commandes que vous trouvez usqu'au commentaire suivant :

```
phase_end:<phase_name>
```

Toutes les commandes qui s'exécutent entre une étiquette `phase_start` et `phase_end` seront inclus dans cette étape.



Gardez sur compte que les enregistrements réalisés dans Selenium 3 ne peuvent pas fonctionner correctement.

Exécutions des transactions web

Exécution standard

Pour lancer des sessions pré-enregistrés de PWR appelez à l'exécutable `pandora_ux_x64.exe`, que vous pouvez télécharger depuis le lien suivant :

<https://pandorafms.com/library/user-experience-pandora-ux>

Indiquez que le mode de travail est PWR, et le fichier qui contient les instructions de la session. Dans le chemin indiqué dans le paramètre `-folder` les captures d'écran à montrer en cas d'erreur seront sauvegardées (paramètre optionel). Vous pouvez également introduire le nombre de réessais consécutifs au cas d'échec (paramètre optionel). Son exécution dans Windows est :

```
pandora_ux_x64.exe -exe PWR -script C:\tests\std.html -retries 3
```

Les modules suivants seront retournés :

- `UX_Status_<nom_projet>` si la séquence a été complétée ou a échouée.
- `UX_Time_<nom_projet>` temps employée pour compléter la séquence.
- `UX_Snapshot_<nom_projet>` capture d'écran avec le dernier point avant de l'erreur, s'il y en a.

Exemple de la sortie avec l'exécution correcte :

```

<module>
  <name><![CDATA[UX_Status_std.html]]></name>
  <type>generic_proc</type>
  <data><![CDATA[1]]></data>
  <description><![CDATA[Test OK]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
</module>
<module>
  <name><![CDATA[UX_Time_std.html]]></name>
  <type>generic_data</type>
  <data><![CDATA[16.317]]></data>
  <description><![CDATA[Test OK]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>

```

Exemple de sortie avec exécution incorrecte :

```

<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[data:image/png;base64,
iVBORw0KGgoAAAANSUgAA...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>

```

Si tout a été correcte, vous pouvez ajouter la ligne d'exécution à l'agent de Pandora FMS installé dans la machine que vous ayez préparé pour exécuter les vérifications. La ligne à ajouter au

fichier de configuration de l'agent aura un aspect similaire à celui-ci (dans une seule ligne):

Sous Windows® :

```
module_plugin C:\Users\artica\Documents\Producto\UX-Trans\UX\pandora_ux_x64.exe
-exe PWR -script C:\Users\artica\Documents\Producto\UX-Trans\PWR\sesion.html -
folder <ruta screenshots>
```

Sous Linux :

```
module_plugin /usr/share/pandora_server/util/pwr/firefox/pandora_ux.64 -exe PWR
-script /usr/share/pandora_server/util/pwr/firefox/PruebaUX.html -pwr_port 4444
```

Une fois qu'il rapporte les informations au serveur Pandora FMS, vous verrez que les modules correspondants apparaissent. Vous pouvez habiller la viste en mode jérarquique pour voir la relation entre eux :

	UX_Status_sesion.html	Test OK	 N/A - N/A	1			45 seconds	
	UX_Snapshot_sesion.html	Image (last error)	 N/A - N/A	No image ava				20 hours
	UX_Time_sesion.html	Test OK	 N/A - N/A	4.7			45 seconds	

Comme indiqué précédement, vous verrez le résultat de l'exécution (si elle a été complétée) dans le module `UX_Status_sesion.html`, le temps employé (en secondes) dans le module `UX_Time_sesion.html`, et un autre avec la capture d'écran du dernier erreur dans `UX_Snapshot_sesion.html`, que dans ce cas est vide. Vous pouvez créer des alertes sur ces modules et montrer les graphiques individuelles de chaque module.

Exécution basée sur des étapes

Si vous avez un enregistrement transactionnel avec Pandora FMS UX PWR, comme indiqué dans les sections précédentes, le système lui-même générera les modules nécessaires pour identifier chacune des étapes indiquées, donc l'exécution ne sera pas différent du cas précédent ; uniquement indiquez le fichier html correspondant, qui dans ce cas contiendra les différentes étapes. Exemple d'exécution sous Windows :

```
pandora_ux_x64 -exe PWR -script C:\tests\std.html --folder <ruta screenshots>
```

Les modules suivantes seront retournés par étape :

- UX_Time_nom_projet.étape_ordre
- UX_Status_nom_proyjet.étape_ordre

S'il y a une étape avec des erreurs, le module suivant sera aussi créé :

- UX_Snapshot_nom_projet.étape_ordre

Il montrera une image du web dans le moment de l'erreur.

Les modules de résumé globales seront aussi retrouvés indentifiés avec les noms suivants :

- UX_Global_Time_nom_projet
- UX_Global_Status_nom_projet
- UX_Global_Snapshot_nom_projet

Et une image du web sera montrée dans le moment de l'erreur.

La ligne d'exécution de l'agent sera la même que le cas précédent, mais avec le fichier html qui contient les étapes accordées.

Lorsque l'information arrive correctement au serveur Pandora FMS, il sera affiché sous forme de modules de cette manière. Habilitier la vue sous mode hiérarchique dans la liste de modules vous aide à afficher l'information de manière beaucoup plus claire :

 	UX_Global_Status_sesion_fases.html	Phase fase1 OK		N/A - N/A	1			4 seconds	
 	L UX_Global_Snapshot_sesion_fases.html	Image (last error)		N/A - N/A				6 hours	
 	L UX_Global_Time_sesion_fases.html	Phase fase1 OK		N/A - N/A	9.1			4 seconds	
 	L UX_Status_sesion_fases.html.fase1_0	Phase fase1 OK		N/A - N/A	1			4 seconds	
 	L UX_Time_sesion_fases.html.fase1_0	Phase fase1 OK		N/A - N/A	2.1			4 seconds	
 	L UX_Snapshot_sesion_fases.html.fase1_0	Image (last error)		N/A - N/A	No image ava				6 hours

En accédant à la section *WUX* de l'agent, vous pourrez voir les détails supplémentaires de la transaction et ses étapes.

L'exemple ci-dessous représente un processus de navigation à travers notre site web et a quelques articles, divisé en 5 étapes afin de mesurer exactement les temps employés et savoir où est ce qu'il faut des améliorations ou où est-ce qu'il se produit un goulot d'étranglement :

  UX_Global_Status_pandora_fases.html	Phase open_web OK Phase open_shop OK Phase add_article OK P...		N/A - N/A	1	 	10 seconds
  L UX_Global_Time_pandora_fases.html	Phase open_web OK Phase open_shop OK Phase add_article OK P...		N/A - N/A	21	 	10 seconds
  L UX_Status_pandora_fases.html.add_article_2	Phase add_article OK		N/A - N/A	1	 	11 seconds
  L UX_Time_pandora_fases.html.add_article_2	Phase add_article OK		N/A - N/A	4,7	 	11 seconds
  L UX_Status_pandora_fases.html.finish_shopping_4	Phase finish_shopping OK		N/A - N/A	1	 	11 seconds
  L UX_Time_pandora_fases.html.finish_shopping_4	Phase finish_shopping OK		N/A - N/A	2,5	 	10 seconds
  L UX_Status_pandora_fases.html.keep_buying_3	Phase keep_buying OK		N/A - N/A	1	 	11 seconds
  L UX_Time_pandora_fases.html.keep_buying_3	Phase keep_buying OK		N/A - N/A	5,8	 	11 seconds
  L UX_Status_pandora_fases.html.open_shop_1	Phase open_shop OK		N/A - N/A	1	 	11 seconds
  L UX_Time_pandora_fases.html.open_shop_1	Phase open_shop OK		N/A - N/A	1,9	 	11 seconds
  L UX_Status_pandora_fases.html.open_web_0	Phase open_web OK		N/A - N/A	1	 	11 seconds
  L UX_Time_pandora_fases.html.open_web_0	Phase open_web OK		N/A - N/A	1,6	 	11 seconds

Récupération de valeurs

Pandora FMS UX est capable de récupérer des données directement depuis le web pendant l'exécution des vérifications d'expérience utilisateur.

Pour utiliser cette nouvelle caractéristique, il faut seulement ajouter l'ordre de capturer sous forme de commentaire la vérification de Selenium :

ejemplito (untitled suite) - Selenium IDE 2.9.1

File Edit Actions Options Help

Base URL `http://fringe.lab.artica.lan/`

Fast Slow

Test Case: **ejemplito**

Command	Target	Value
phase_start:Entrada a página		
open	/prueba.php	
type	id=campo	ejemplito
clickAndWait	name=enviar	
phase_end:Entrada a página		
getValue;numeroito;generic_data;	(\d+\.\d*)</spa	
getValue;resultado en texto;generic_data_string;	<h1>(.*?)</h1>	

Command: `getValue;resultado en texto;generic_data_string;<h1>(.*?)</h1>`

Target:

Value:

Runs: 1

Failures: 0

Log Reference UI-Element Rollup

open(url)

Arguments:

- url - the URL to open; may be relative or absolute

Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load before proceeding, ie. the "AndWait" suffix is implicit. *Note:* The URL must be on the same domain as the runner HTML due to security restrictions in the browser (Same Origin Policy). If you need to open an URL on another domain, use the Selenium Server to start a new browser session on that domain.

Dans cet exemple, deux valeurs sont récupérées après de naviguer, il seront représentés en tant que modules dans Pandora FMS (numéro et résultat en tant que texte).

Les étapes pour ordonner le plugin de faire la collecte de données sont les suivants :

- Ajouter un nouveau commentaire au fichier des vérifications avec l'enregistreur Selenium.
- Ajouter la directive au commentaire.

Directive de capture de données

```
getValue;<module name>;<module data type>;<Perl regular expression for data capture>
```

Qu'est-ce que représente chaque champ séparé par des points-virgules?

getValue

Instruction pour le système UX.

Nom du module

Le nom du module tel qu'il apparaît sur Pandora FMS.

Type de donnée du module

Quel type sera utilisé ; il peut être n'importe quel des [types de données](#) supportés par Pandora FMS.

Expression régulière Perl à utiliser pour la capture de données

Elle doit être une expression régulière Perl, avec l'ordre de capture d'information entre parenthèses.

Par exemple, dans le texte :

```
<p> The temperature of the solar panel is: <span class="temperature">54°  
C</span></p>
```

Si vous voulez récupérer la température pour suivre la valeur, spécifiez l'expression régulière comme suit :

```
<span class="temperature">(\d+\.\.*\,*\d*).*</span>
```

De manière que la valeur 54 de l'exemple sera récupérée, ou possibles valeur avec des positions décimaux.

La directive de capture complète serait comme suit :

```
getValue;solar panel temperature;generic_data;<span  
class="temperature">(\d+\.\.*\,*\d*).*</span>
```

Les commandes qui génèrent des modules sont :

- getValue : Extrait une valeur.

```
getValue;module_name;module_type;REGEX_string_match
```

- getVariable : (Version NG 753 ou supérieure) Il extrait une valeur spécifique d'une variable.

```
getVariable;module_name;module_type;var_name
```

- getScreenshot : Extrait une capture d'écran.

```
getScreenshot;module_name
```

getValue est une commande qui appartient au type qui **gènèrent les modules** Pandora FMS. Visitez le lien précédent pour en savoir plus.

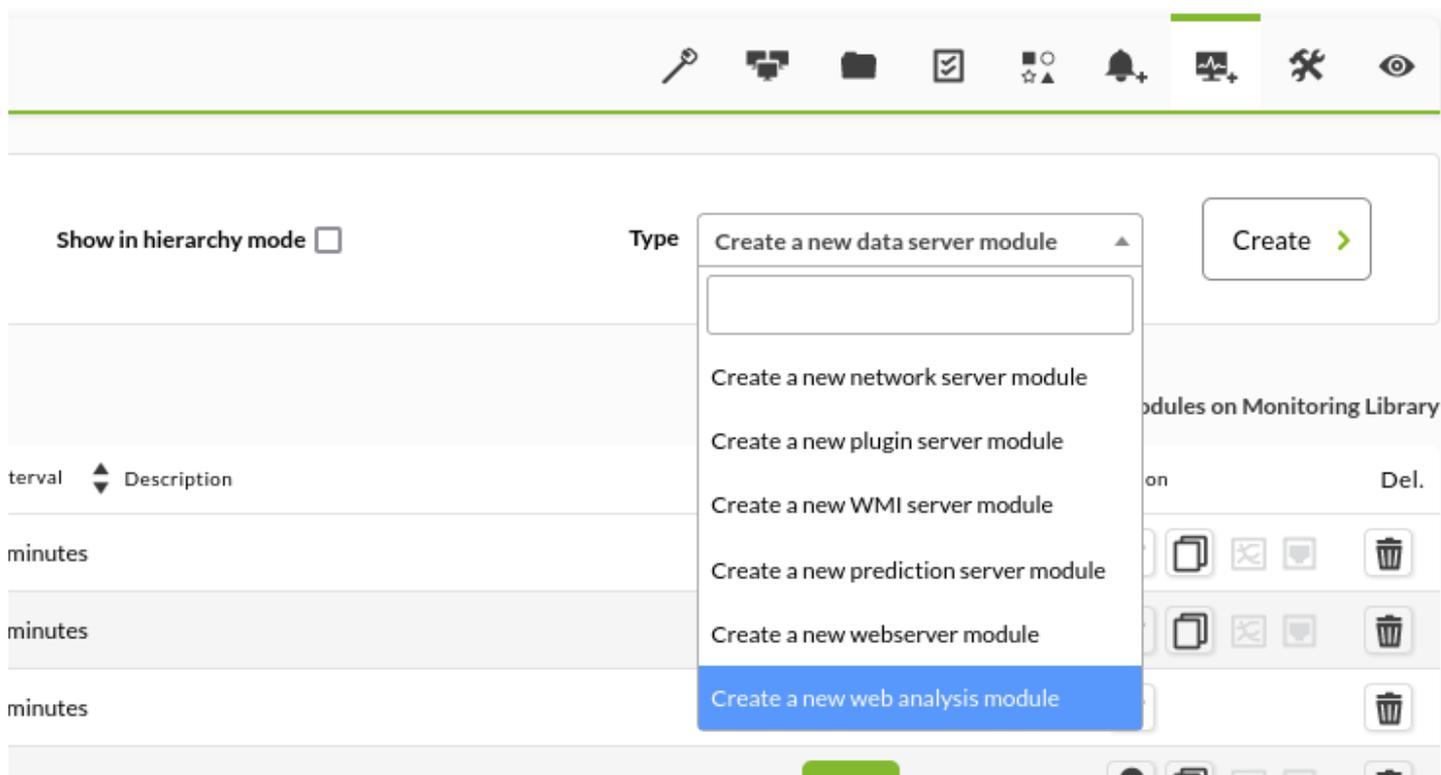
Tous les commandes propres doivent apparaître dans le fichier de vérification (.side) en tant que commentaires. Au cas contraire, il échouera lors de la vérification dans l'extension de Selenium.

Gestion et affichage de données

Créer un module d'analyse web dans la Console Pandora FMS

Une fois vous avez une session de navigation enregistrée, il faut le déployer en tant que Module dans la Console Pandora FMS.

À cette fin, accédez à n'importe quel des agents associés au serveur avec une fonctionnalité WUX activée et créez un module du type Analyse web en choisissant l'option Create a new web analysis module :



The screenshot shows the Pandora FMS console interface. At the top, there is a navigation bar with various icons. Below it, there is a section for creating a new module. On the left, there is a checkbox labeled "Show in hierarchy mode". In the center, there is a "Type" dropdown menu with the following options:

- Create a new data server module
- Create a new network server module
- Create a new plugin server module
- Create a new WMI server module
- Create a new prediction server module
- Create a new webserver module
- Create a new web analysis module (highlighted in blue)

To the right of the dropdown menu is a "Create" button with a right-pointing arrow. Below the dropdown menu, there is a table with columns for "Description" and "Del.". The table contains several rows, each with a "minutes" value in the "Description" column and a trash icon in the "Del." column.

Cliquez sur Create et après remplissez tous les champs dans le formulaire :

✓ **Base options**

Name Disabled
ID 107

Run performance tests

Target web site

Execute tests from

Browser

User data dir

Profile

Accept insecure certificates

Keep counters

FF threshold Change all statuses :
 Change each status : To `normal` To `warning` To `critical`

Historical data

Paste your selenium test, exported as HTML, here

No file selected.

Name

Module WUX (Père) ; rappelez-vous que tous les sousmodules avec les résultats de la supervision dépendront de ce nouveau module.

Run performance tests

Il spécifie le lancement des vérifications de performance, c'est à dire de superviser pas seulement l'expérience utilisateur mais aussi les statistiques de performance de l'accès au site web cible.

Execute tests from

Il indique le serveur WUX qui exécutera la vérification.

Browser

Navigateur web avec lequel exécuter la vérification.

Le navigateur indiqué doit apparaître dans la grille ou serveur Selenium utilisé par le serveur WUX.

User data dir

(*Version 764 ou ultérieure*) Facultatif : il permet de spécifier un répertoire pour les données.

L'option User data dir ne fonctionne que sur le navigateur web Google Chrome.

Profile

(*Version 764 ou ultérieure*) Facultatif : il permet de spécifier un profil utilisateur. Si le nom du profil n'existe pas, utilisez le profil par défaut.

L'option Profile ne fonctionne que sur le navigateur web Google Chrome.

Accept insecure certificates

Si est activé, n'importe quel certificat non-sécurisé sera accepté (auto-signé, expiré, etc.) qui fait partie de la navigation.

Cette option est seulement disponible pour les navigateurs Google Chrome® et Mozilla Firefox®, et seulement sera appliquée si la vérification exécutée par un serveur Selenium 3.

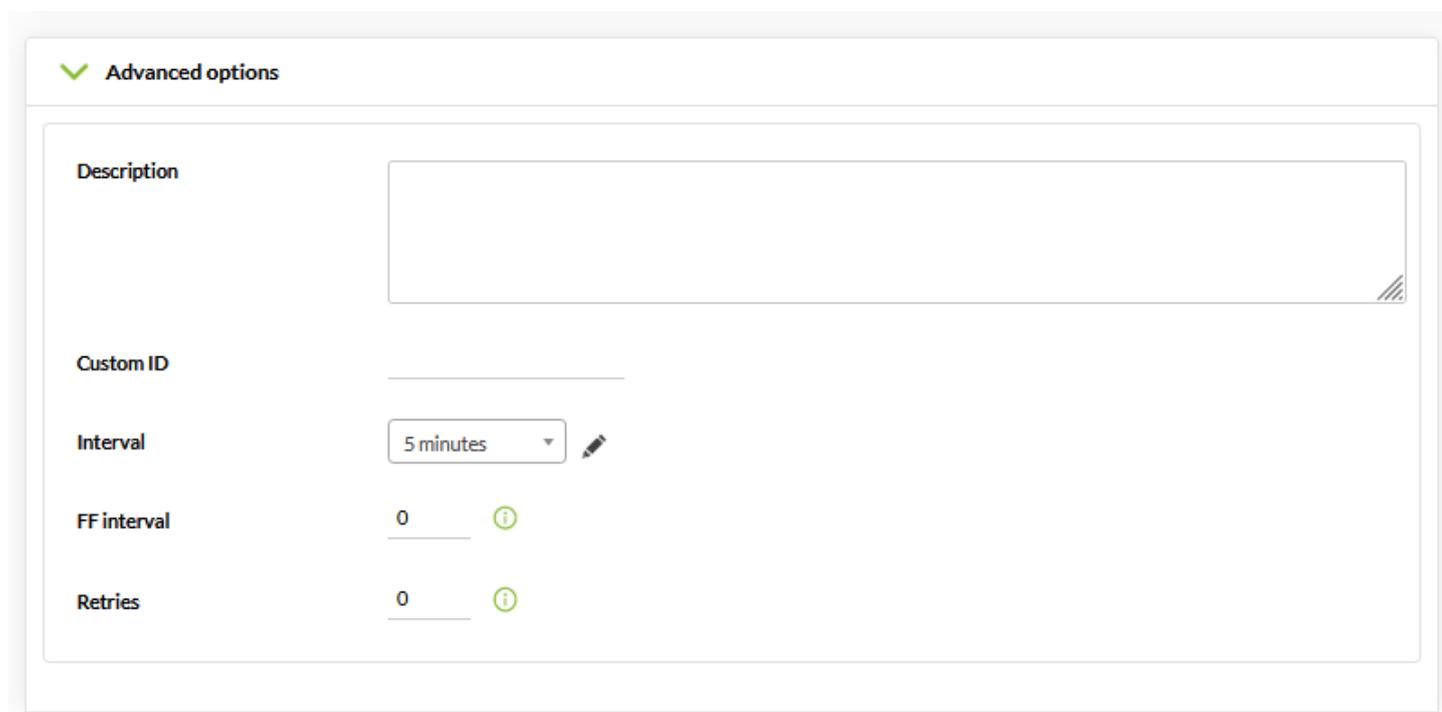
Historical data

Sauvegardez ou pas l'information historique des modules qui supervisent l'expérience utilisateur.

Section de texte

Copiez (ou téléchargez avec le bouton Browse...) le contenu des fichiers de la session de

navigation que vous avez enregistré précédemment.



Advanced options

Description

Custom ID

Interval 5 minutes

FF interval 0

Retries 0

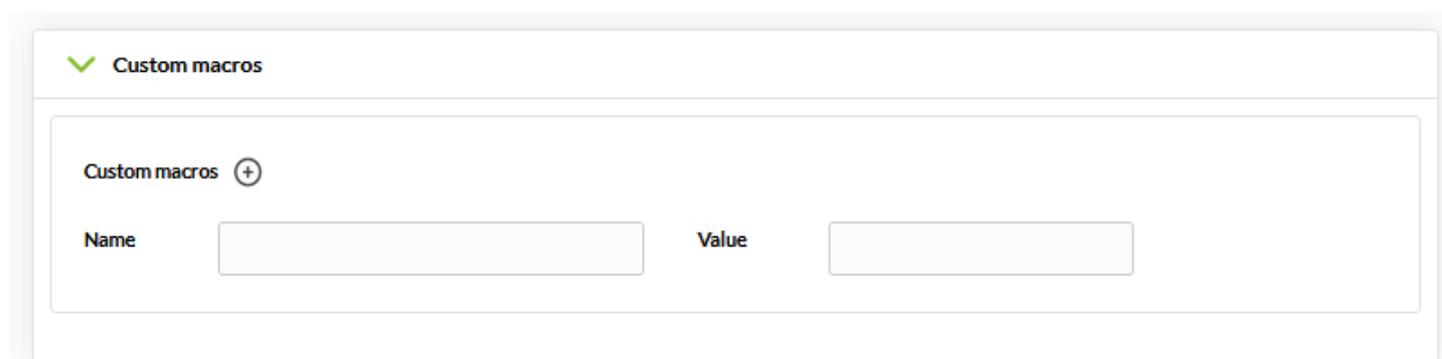
FF interval

Intervalle de temps Flip Flop d'exécution du module (en secondes).

Retries

Nombre de tentatives pour lancer le module.

Macros personnalisées



Custom macros

Custom macros (+)

Name Value

Le fonctionnement des macros personnalisées comprend le remplacement de certaines chaînes de texte présentes dans le fichier de session navigation par certaines valeurs personnalisables.

Dans les cas de modules d'analyse web, cette fonctionnalité a été améliorée, permettant des macros dynamiques qui permettent que ces valeurs soient traduits en dates et heures variables.

Pourquoi cette fonctionnalité ?

Supposez que vous avez besoin de superviser, par le biais d'une session de navigation, le correcte fonctionnement d'une application web de réservation de salles de réunion.

Si vous établissez une date et heure fixée lors de remplir les données du formulaire, il est possible que le système annule la réservation parce à quelque moment il est dans le passé ; par exemple, vous pourriez réserver une salle pour la semaine prochaine.

Il est aussi possible de trouver un temps maximale pour pouvoir faire la réservation et que le système vous fait réserver la salle dans un temps concret, par exemple pas supérieur au mois en vigueur.

Pour éviter d'éditer la macro tous les quelques jours, et ne pas s'inquiéter à cause de la section de configuration, vous pouvez utiliser les macros dynamiques, en indiquant le système que vous réservez la salle toujours pour le jour suivant au moment de faire la vérification.

Afin de pouvoir utiliser cette fonctionnalité, les valeurs doivent avoir un format spécifique, en obtenant les remplacements possibles suivantes :

- @DATE_FORMAT: date/heure actuelle avec format défini par l'utilisateur.
- @DATE_FORMAT_nh: heures.
- @DATE_FORMAT_nm: minutes.
- @DATE_FORMAT_nd: jours.
- @DATE_FORMAT_ns: secondes.
- @DATE_FORMAT_nM: mois.
- @DATE_FORMAT_nY: années.

Ou " n " peut être un numéro sin signe positif ou négatif.

Et FORMAT suit le standard de [strftime de Perl](#)

Exemples :

```
@DATE_%Y-%m-%d %H:%M:%S
@DATE_%H:%M:%S_300s
@DATE_%H:%M:%S_-1h
```

Affichage des données

L'information générée par WUX sera affichée sous forme de modules comme suit. L'activation de la vue en mode hiérarchique, dans la liste de modules aidera à afficher l'information de manière beaucoup plus claire :

Total de elementos 18

F.	P.	Tipo ▲▼	Nombre módulo ▲▼	Descripción	Estado ▲▼	Advertencia	Datos	Gráfico
○			coddns			N/A - N/A	1	
			L coddns_Global_Status	Test OK		N/A - N/A	1	
			L coddns_Global_Time	Test OK		N/A - N/A	6,6	
			L coddns_Phase 0: Wordpress_Status			N/A - N/A	1	
			L coddns_Phase 0: Wordpress_Time			N/A - N/A	4,7	
			L coddns_Phase 1: Application_Status			N/A - N/A	1	
			L coddns_Phase 1: Application_Time			N/A - N/A	1,9	
			L coddns_UX_Stats_DNS			N/A - N/A	4	
			L coddns_UX_Stats_TSSL			N/A - N/A	604	
			L coddns_UX_Stats_TST			N/A - N/A	829	
			L coddns_UX_Stats_TT			N/A - N/A	862	
			L coddns_UX_Stats_TTC			N/A - N/A	33	
			L coddns_UX_Stats_TTCP			N/A - N/A	51	
			L coddns_UX_Stats_TTR	Time global spent in retrieve https://coddns.es (selected it...		N/A - N/A	9,111	
			L coddns_UX_Stats_TTR_css	Time spent in retrieve css resources from https://coddns.es		N/A - N/A	2,649	
			L coddns_UX_Stats_TTR_image	Time spent in retrieve image resources from https://coddns.e...		N/A - N/A	2,230	
			L coddns_UX_Stats_TTR_js	Time spent in retrieve js resources from https://coddns.es		N/A - N/A	3,398	
			L coddns_UX_Stats_TTR_Main	Time spent in retrieve main HTML from https://coddns.es		N/A - N/A	834	

Dans cette hiérarchie vous trouverez les modules suivants :

- *módulo_Global_Status* : Indiquera l'état global de la navigation complète.
 - Au cas où il existe un enregistrement il donne l'état de l'enregistrement WUX.
 - Dans le cas où les **run performance Test** sont activées mais il n'y a pas d'enregistrement inclu, l'état du module Global Status est celui de la vérification fait lors de l'obtention de ces modules.
- *módulo_Global_Time* : Il indiquera le temps global employé dans la navigation complète.
- *módulo_Global_Screenshot* : Il contient une image avec le résultat de l'échec de navigation, il ne sera généré qu'en cas d'échec.
- *módulo_Phase X : Nom de étape_Status* : Il indiquera l'état de la navigation pendant l'étape X.
- *módulo_Phase X : Nom de étape_Time* : Il indiquera le temps employé dans l'étape X.

Exemple : Vue de la capture d'erreur.

The screenshot displays the Pandora FMS console interface. On the left, there is a sidebar with navigation options like 'Full list of monitors' and 'List of modules'. The main area is split into two parts: a top section showing a 'Current data at 2017-09-08 01:45:59' snapshot of a website (CONTACTO) and a bottom section showing a list of monitoring tests. A red arrow points from the 'Current data' section to the 'Description' column of the test list.

Description	Status	Warn	Data	Graph
[Acceso al portal]: Failed to execute clickAndWait at link=M...	Green	N/A - N/A	1	100%
Test OK	Green	N/A - N/A	1	100%
	Green	N/A - N/A	17,6	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	5,7	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	3,3	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	2,5	100%

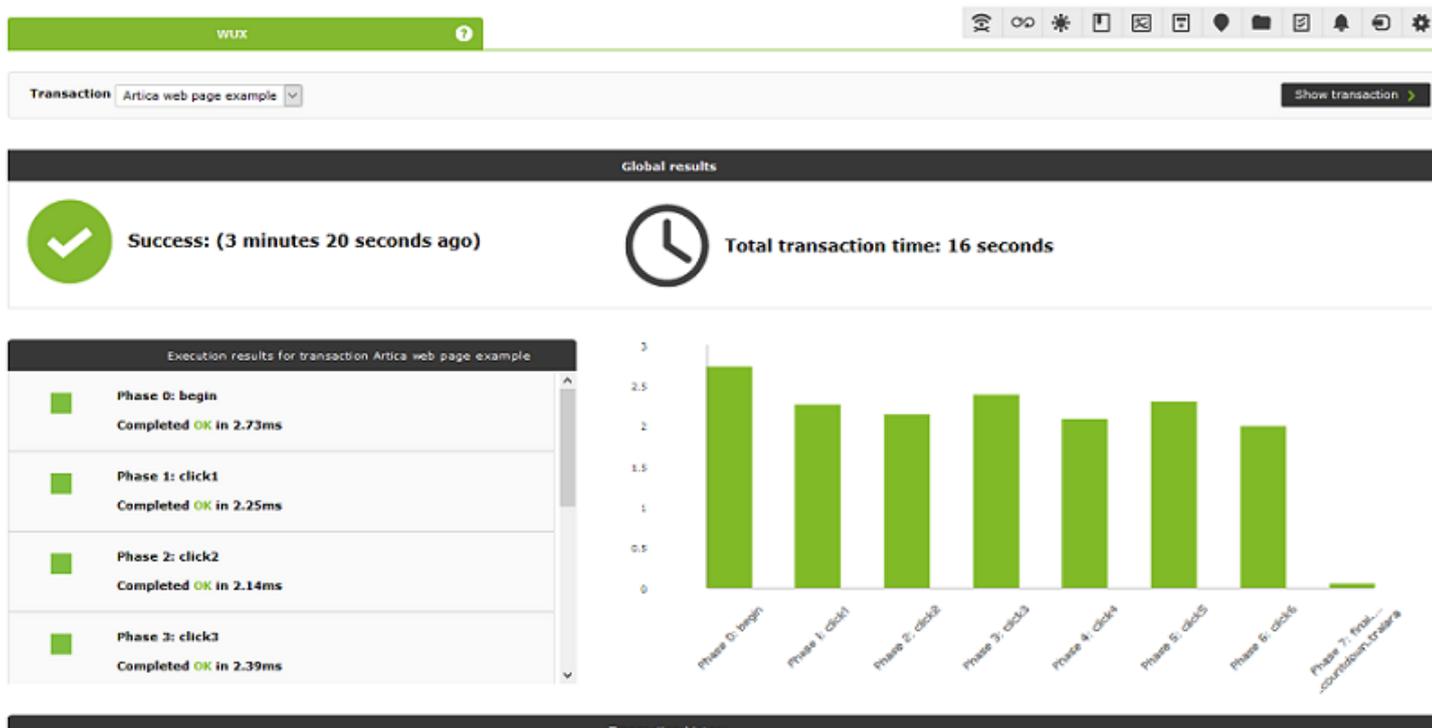
Attention : Si vous avez mis à jour depuis une version précédente à Pandora FMS 7.0NG 712 vous devez faire un léger changement.

Afin de pouvoir stocker les captures d'écran générées par le serveur WUX, vous devez lancer les opérations suivantes dans votre schéma de base de données :

```
alter table tagente_estado modify column datos mediumtext;
alter table tagente_datos_string modify column datos mediumtext;
```

Si vous ne faites pas ces actions, les captures pourraient être incomplètes.

En accédant à la section WUX de l'agent vous pouvez voir des détails supplémentaires de la transaction et ses étapes :



Les statistiques du site web sont résumées dans les concepts suivants :

- Stats_TT : Temps total pour obtenir le site web.
- Stats_TDNS : Temps total pour résoudre l'adresse IP du cible.
- Stats_TTCP: Temps employé pour connecter par le biais de TCP.
- Stats_TSSL : Temps employé pour établir communication SSL.
- Stats_TST : Temps passé jusqu'au démarrage du transfert de données.
- Stats_TTC : Temps passé en transférant des données ; il groupera tous les temps de transfert de ressources.
- Stats_TTR : Temps employé pour transférer le contenu de la page.
- Stats_TTR_main : Temps employé pour transférer le code HTML.
- Stats_TTR_image : Temps employé pour transférer les ressources du type image (png | jpg | jpeg | bmp | tiff | gif | webp | svg).
- Stats_TTR_css : Temps employé pour transférer les feuilles de styles.
- Stats_TTR_js : Temps employé pour transférer les fichiers JavaScript.

Attribuer des alertes à un module d'analyse web

Les alertes associées aux modules d'analyse web suivent la même distance dynamique de fonctionnement que tout le système d'alertes Pandora FMS.

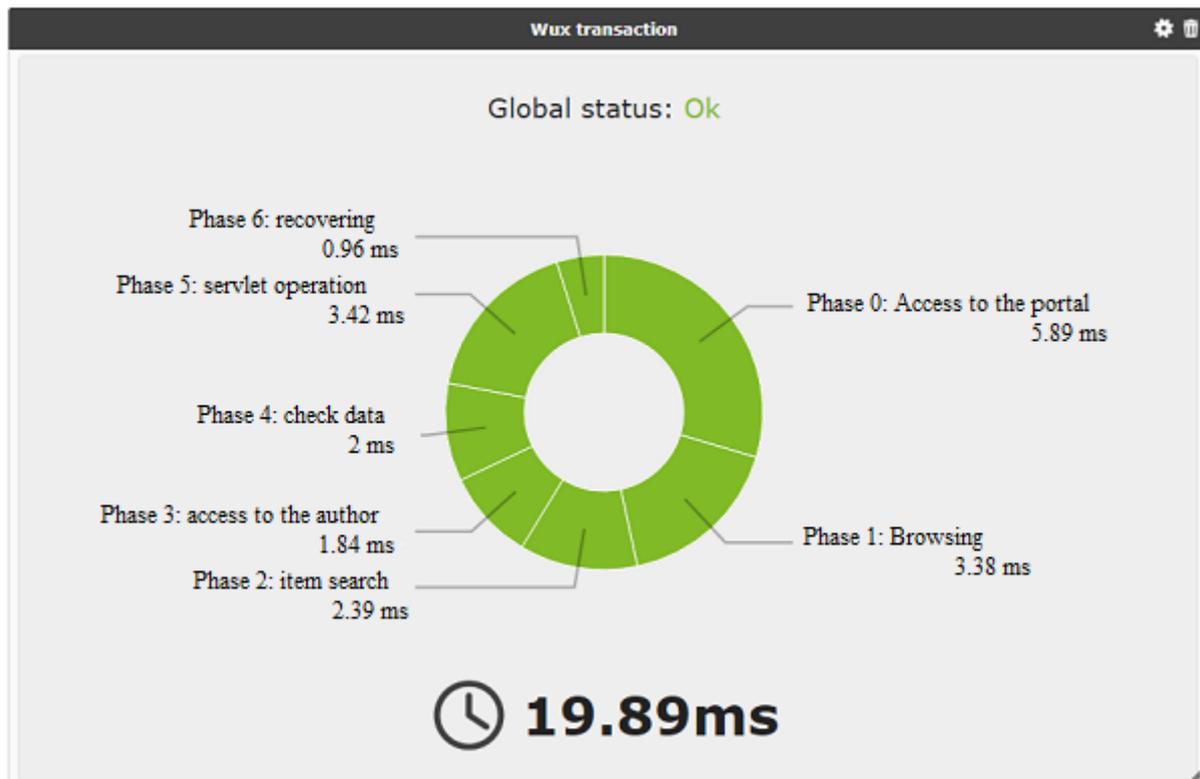
Pour compatibilité il est recommandé d'attribuer les modèles d'alertes sur les sous-éléments autogénérés par le module d'analyse web, comme par exemple :

- L'état de navigation global.
- Alertes sur les seuils de temps.
- Alertes avec modèle d'avertissement mis sur " toujours " pour les modules capture d'image de résultats.

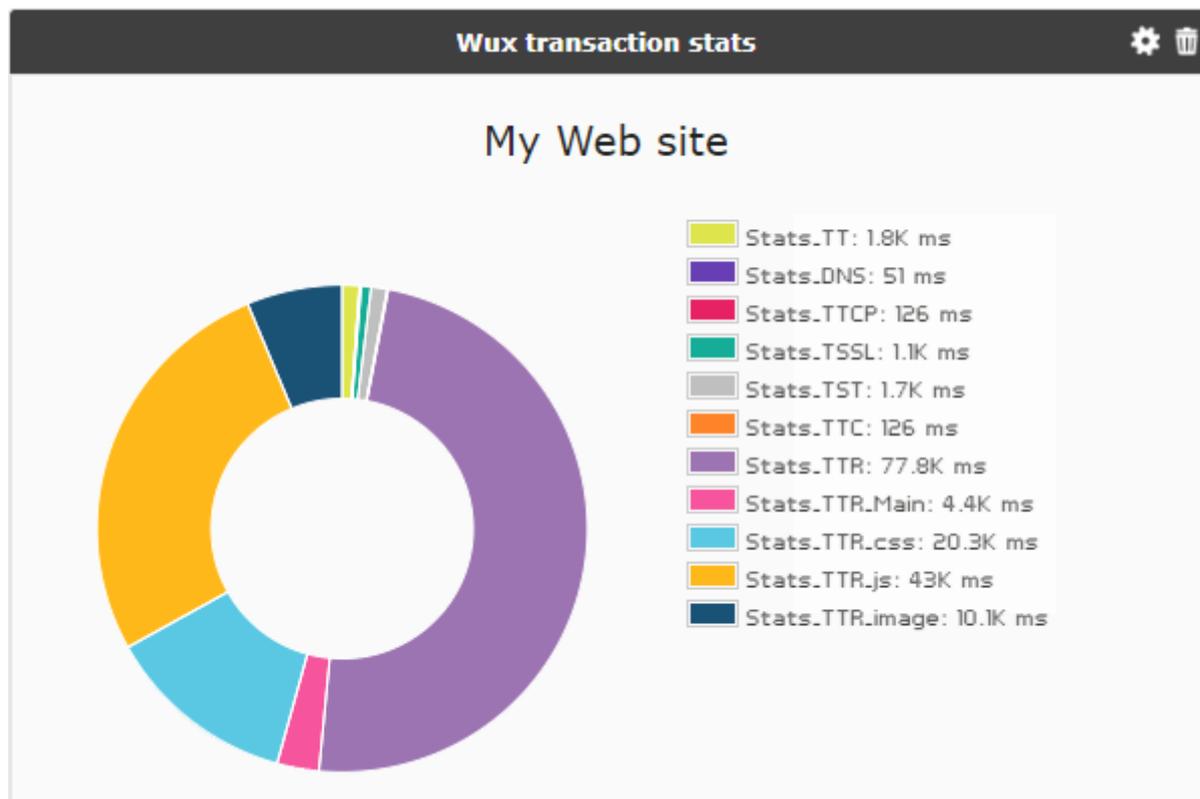
Nouveaux widgets pour Dashboards

Afin de pouvoir représenter les informations de la navigation d'utilisateurs, de nouveaux types de widget ont été ajoutés :

Widget d'état des étapes de navigation



Widget de statistiques du site web



Expérience utilisateur bureau(PDR)

Déploiement du système PDR

E

Le système PDR fonctionne uniquement dans des systèmes Windows®, et une fois en exécution par l'Agent de Pandora FMS, celui-ci doit être exécuté en mode processus, car être exécuté en mode service ne fonctionnera pas. En outre, il ne sera pas possible de bloquer la session de bureau, donc il est recommandé de l'utiliser dans des machines virtuelles.

Exigences préalables :

- Système Windows® en mode bureau avec auto-démarrage et auto-connexion.

Pour y parvenir à cette configuration :

Pour des versions précédentes à Windows 10® :

```
control userpasswords2
```

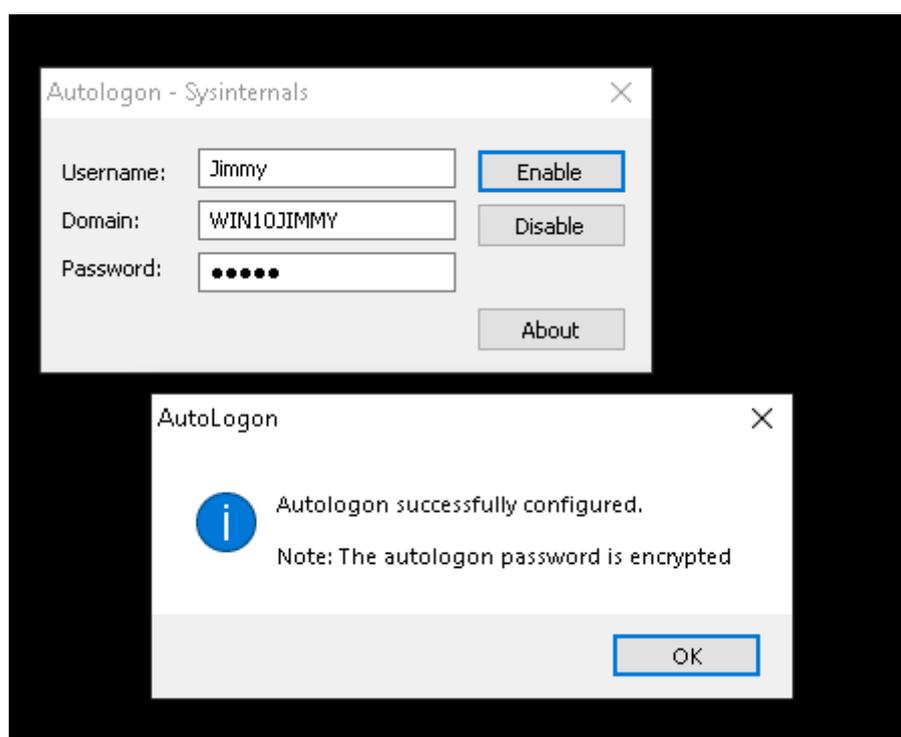
Et décochez Users must enter a user name and password to use this computer :



Cliquez sur Ok et insérez le nom d'utilisateur et votre mot de passe.

Pour Windows 10® :

Une manière d'activer l'auto-démarrage sur MS Windows®, version 10, c'est d'utiliser Powertoyoys de Sysinternals (disponible dans la [section de téléchargements](#) de la documentation de Microsoft®) et activer l'utilisateur qui supervisera, par exemple :



Pour préparer l'environnement créez le répertoire suivant :

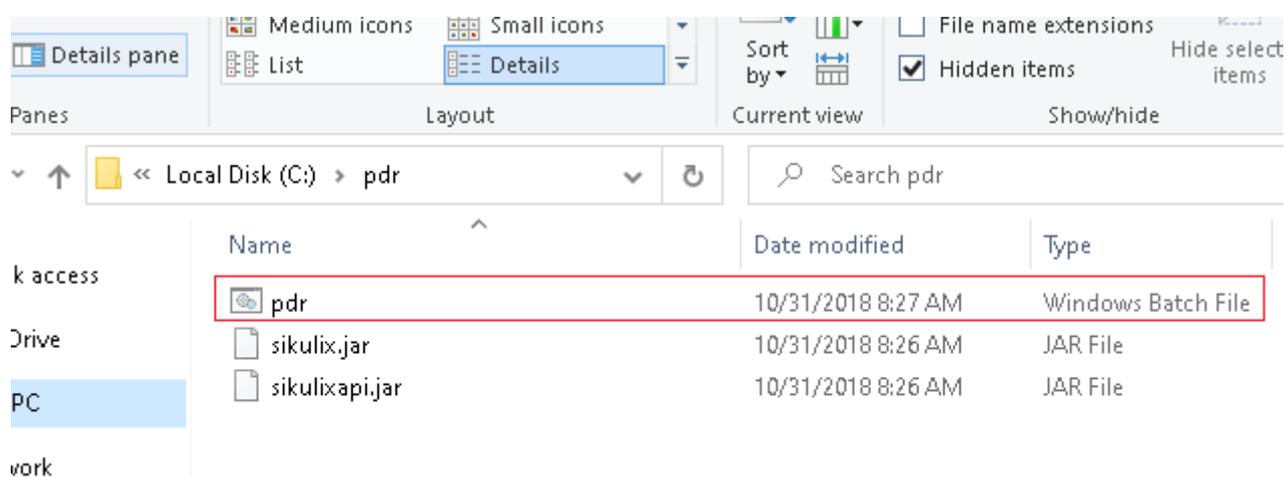
```
C:\PDR
```

Et décompressez le fichier PDR.rar (fournit depuis ce lien <https://pandorafms.com/library/pdr-cmd-for-ux-monitoring/>) dans ledite dossier.

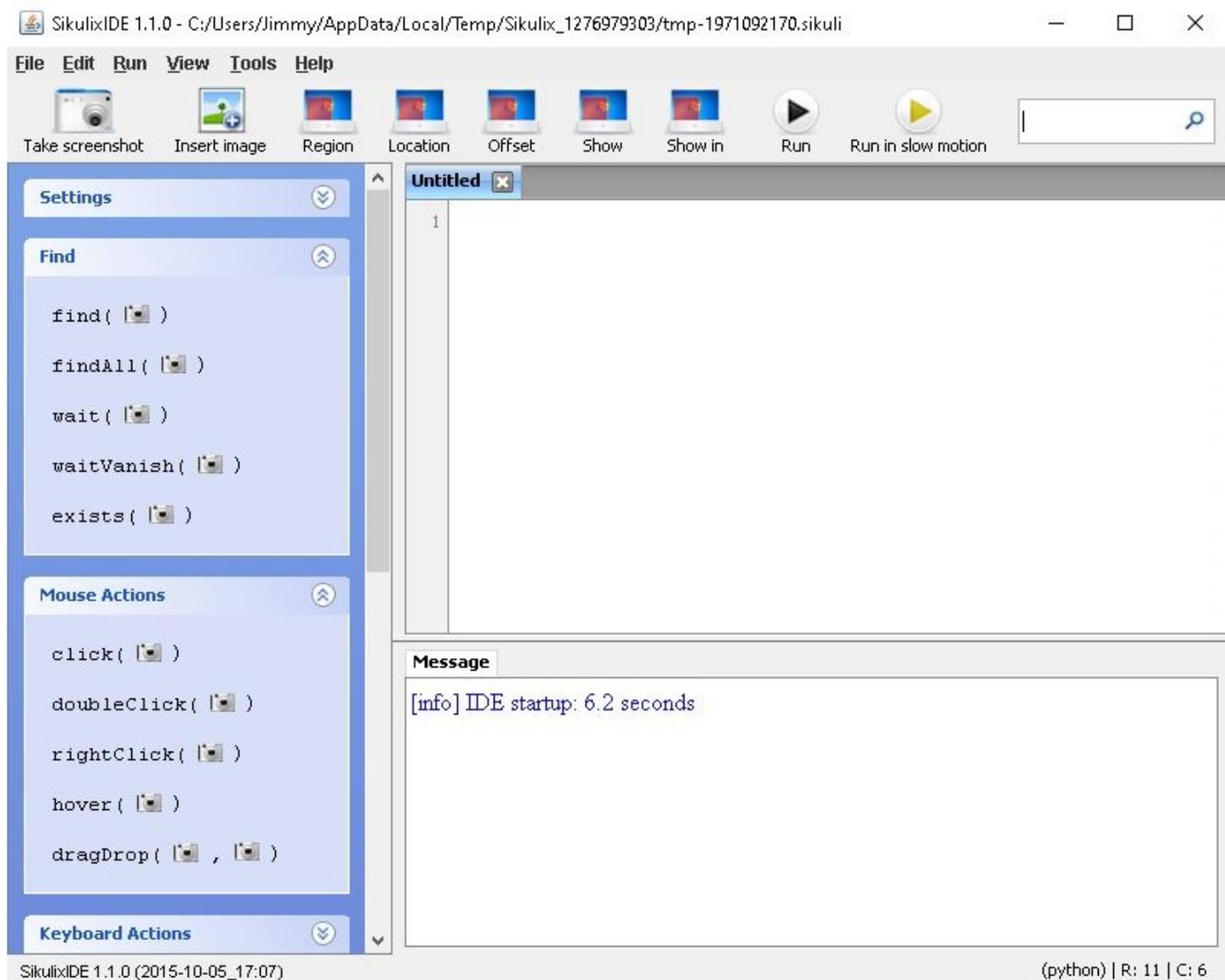
Exécution de sessions PDR

Pour démarrer l'enregistreur, exécutez le script suivant:

```
C:\PDR\pdr
```



Après le chargement, entrez dans l'environnement d'enregistrement:

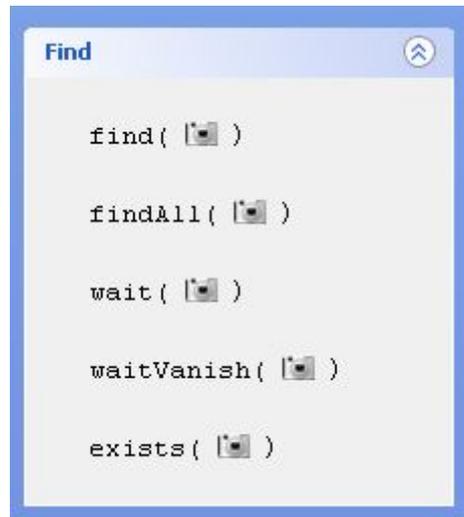


Sélectionnez les actions que vous souhaitez effectuer et faites une capture d'écran dans la zone où vous souhaitez les appliquer. Vous trouverez ci-dessous une liste des actions les plus courantes.

Actions générales :



Actions de contrôle des flux:



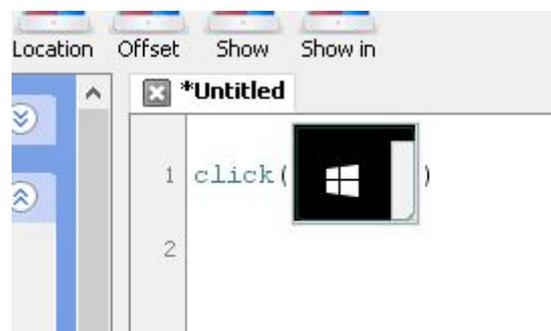
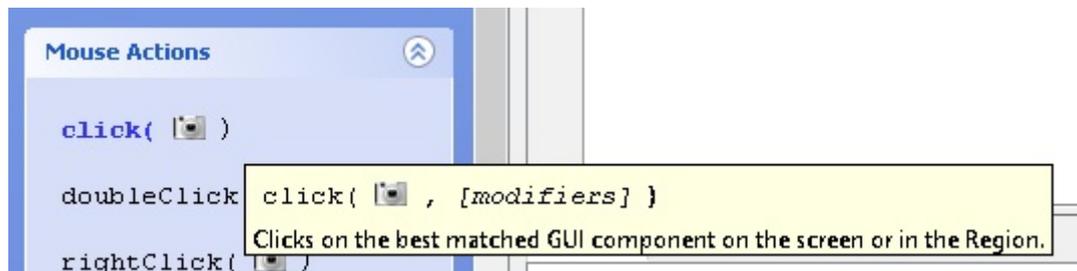
Plus de soutien linguistique: <http://sikulix-2014.readthedocs.io/en/latest/index.html>

Lorsque le projet est enregistré, un dossier contenant les éléments suivants est créé :

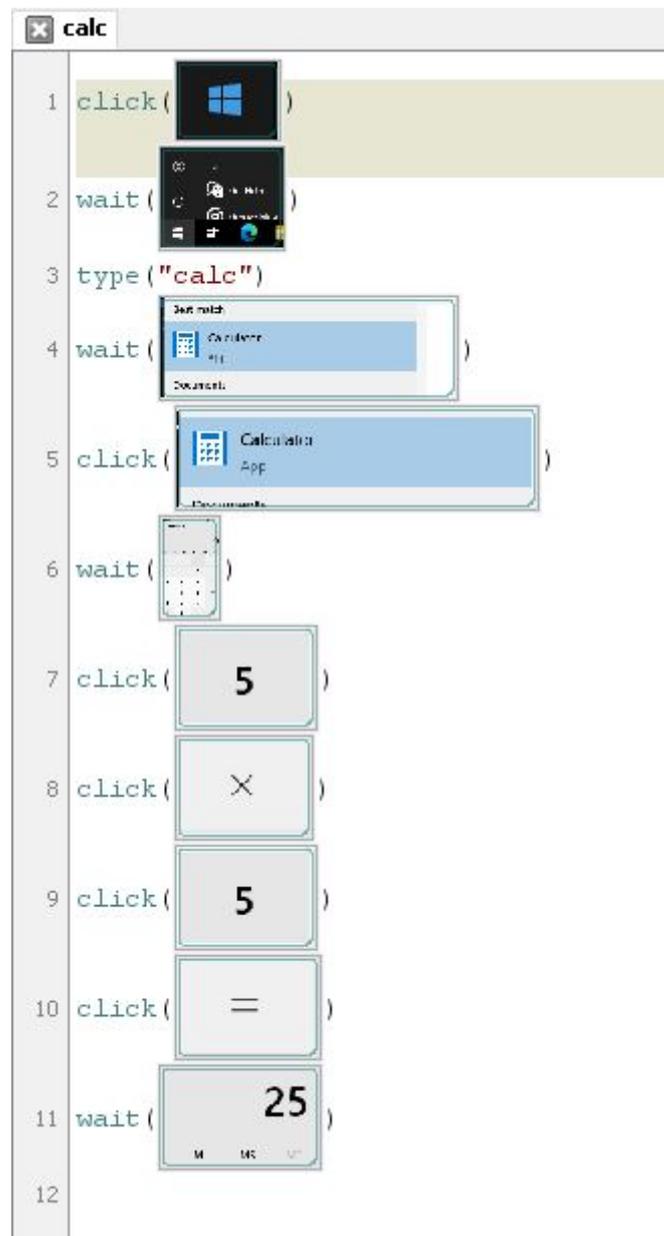
- Fichier .py avec le code du script d'automatisation.
- Images pour contrôler la navigation.

Un exemple simple d'exécution pourrait être le contrôle du bon fonctionnement de votre calculatrice Windows. Nous montrerons l'ensemble du processus à l'aide de captures d'écran.

1. Choisissez l'action "click" et la zone où vous souhaitez appliquer l'action. L'écran passe en mode "area selection" :



2. Saisissez l'action *type* et le texte "calc". Entrez ensuite les actions pour attendre que la zone "calculatrice" apparaisse et cliquez dessus. Attendez ensuite que la fenêtre de la calculatrice apparaisse pour entrer les actions suivantes de cette manière :



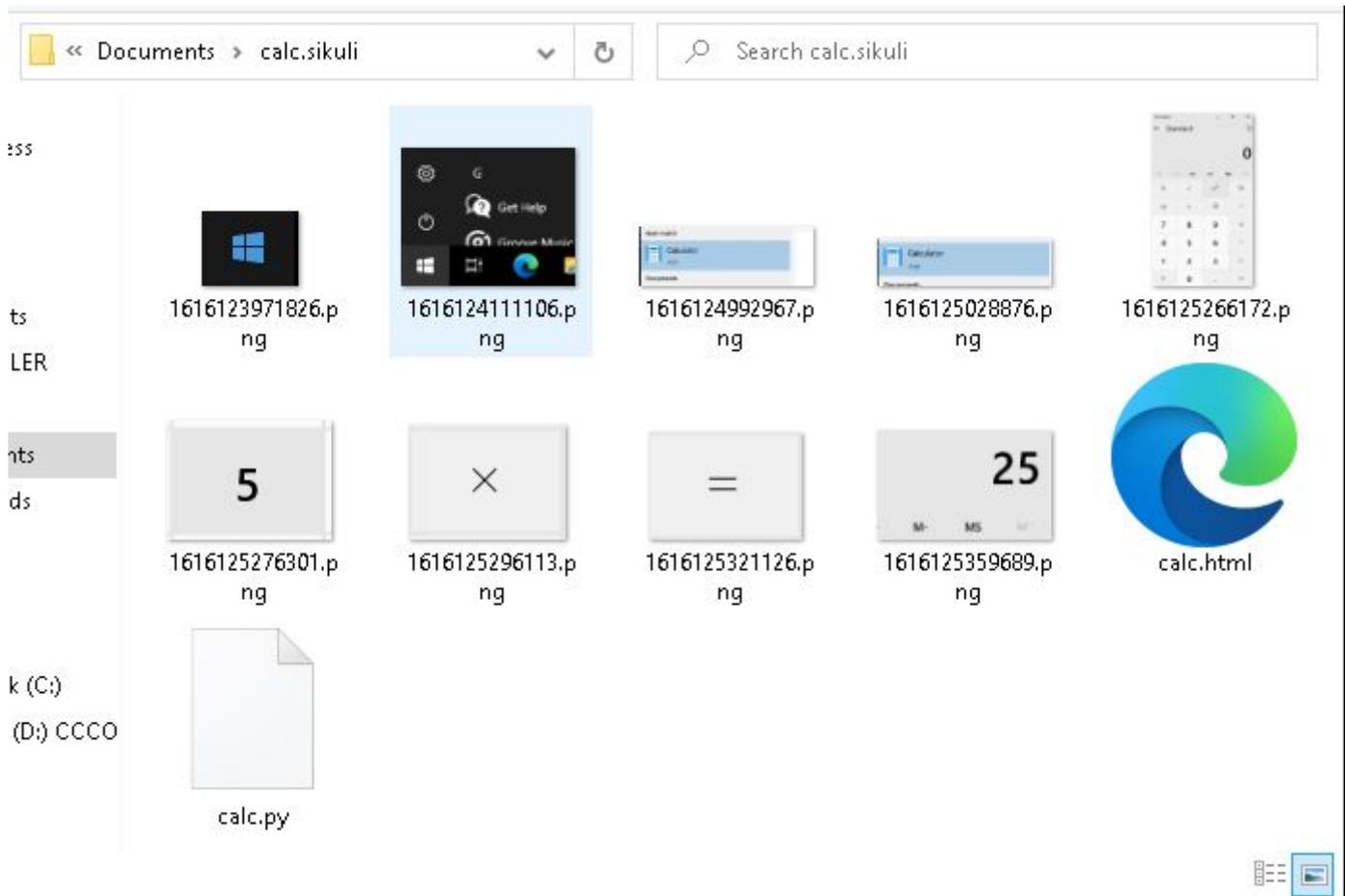
3. Cliquez ensuite sur les boutons correspondants, en sélectionnant les zones comme dans les étapes précédentes.

4. Enfin, sauvegardez le processus et exécutez-le en utilisant la fonction du bouton Run.

Remarques importantes :

- En double-cliquant sur l'une des images de l'environnement d'enregistrement, vous pouvez ajuster les détails de la capture d'écran de contrôle, par exemple en sélectionnant le point exact sur lequel cliquer.
- Il est fortement recommandé de saisir des actions telles que *wait* après chaque clic pour s'assurer que l'exécution ne s'arrêtera pas en raison d'un retard du système d'exploitation.
- L'enregistreur recherchera la zone qui ressemble à la capture d'écran, il faut donc faire très attention au cas où quelque chose serait mis en évidence lorsque le bouton est en haut (par exemple, les boutons de la calculatrice deviennent orange lorsque la souris les survole).

Dans le dossier où vous avez sauvegardé l'enregistrement, vous trouverez les fichiers images et le fichier Python.



Note: Vous pouvez personnaliser la capture effectuée par le moteur d'enregistrement PDR en utilisant le code suivant:

```
## OPTIONAL ##capture file names
import shutil
file = "C:\probes\screenshot_prueba.sikuli.png"
focusWindow = App.focusedWindow()
regionImage = capture(focusWindow)
shutil.move(regionImage, file)
```

Cela créera une image de la fenêtre active à la destination indiquée par le fichier, à partir de laquelle vous pourrez faire une capture d'écran quand vous le souhaitez.

Vous pouvez personnaliser la capture d'écran à l'aide de coordonnées. Pour ce faire, spécifiez la syntaxe suivante :

```
capture(x, y, w, h)
```

Où : X : position horizontale du rectangle dans lequel la capture d'écran doit être effectuée :

X : position horizontale du rectangle pour la capture d'écran.

Y : position verticale du rectangle à capturer.

W : largeur de l'écran.

H : hauteur de l'écran.

Enregistrer une section transactionnelle avec Pandora FMS UX PDR

Il n'est pas nécessaire d'enregistrer une session spécifique ; tout simplement enregistrez les sessions dont vous avez besoin dans des scripts différents. Il sera Pandora FMS UX qui gère le retournement de résultats pour l'organiser en tant que transaction complexe. Dans le point suivant, on verra comment faire l'appel de l'exécution pour qu'il fonctionne correctement.

Si vous allez faire d'enregistrements pour créer un processus transactionnel vous devez faire attention lorsque vous faites l'enregistrement les éléments que vous cherchez soient présents. Il est recommandé d'exécuter manuellement les enregistrements que vous voulez utiliser dans un seul processus transactionnel continu, pour vous assurez que les actions se passent selon vos attentes.

Dans l'exemple suivant, un processus d'enregistrement a été fait, qui constituera la deuxième étape du processus employé dans l'exemple précédent. Nous allons prendre le résultat retourné par la calculatrice et le coller dans un bloc-notes. L'enregistrement présente l'apparence suivante :

```

1 rightClick( 25 )
2 wait( Copy )
3 click( Copy )
4
5 click( Windows )
6 wait( Start Menu )
7 type( "notepad" )
8 wait( Notepad App )
9 click( Notepad App )
10 click( Edit )
11 wait( Edit Menu )
12 click( Paste Ctrl+V )

```

Et nous avons inclus une autre séquence qui consistera à sauvegarder le fichier dans une location particulière, en remplaçant celui précédent. Celui-ci offre une grande flexibilité, car il ouvre la possibilité de superviser parallèlement ces fichiers reflétant les informations dont vous avez besoin dans tout type de variété de vérifications de grands applications de bureau. Cette troisième séquence comprend les étapes suivantes :

```

13 click( Paste Ctrl+V )
14 click( File Menu )
15 click( Save As )
16 wait( Save As Dialog )
17 type( "cal_latest" )
18 click( Save )
19

```

Exécution de sessions PDR

Exécution standard

Pour lancer les sessions pré-enregistrées PDR, indiquez que le mode de travail es le chemin au fichier `pdr.cmd` déployé dans le point correspondant, l'argument du fichier `-r`, le fichier qui contient les directrices de la session (`-script`), le répertoire où stocker les captures d'écran (`-folder`) terminé en `" \ "`, celui qui est optionnel, où sauvegarder les captures dans le dossier où se trouve le `pdr`. Le nombre de réessais consecutifs peut être ajouté en cas d'échec (paramètre optionnel).

Dans l'exécution suivante, la capture d'écran est personnalisée pour collecter uniquement la fenêtre active :

```
pandora_ux_x64 -exe C:\PDR\pdr -args -r -script C:\pandora_ux\calculadora.sikuli -folder C:\pandora_ux\ -ss_config active -retries 3
```

Les modules suivants seront retournés :

- `UX_Time_nom_projet`.
- `UX_Status_nom_projet`.
- `UX_Control_Snapshot_nom_projet` (seulement dans la première exécution).

S'il y a quelque étape avec erreur, le module suivant sera aussi créé :

- `UX_Snapshot_nom_projet`.

Et montrera une image de la fenêtre active (avec `-ss_config active`) du moment de l'erreur.

Exemple de sortie avec exécution correcte :

```
<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 rebuild ]]></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[data:image/png;base64,
IBCAIAAAA0CnfhAAAAAXNSR.../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>

```

Exemple de sortie avec exécution échouée :

```

<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[data:image/png;base64,
iVBORw0KGgoAAAANSUgAA...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>

```

Si tout a été correcte, vous pouvez ajouter la ligne d'exécution à l'agent Pandora FMS installé dans la machine Windows® en tant que plugin d'agent. La ligne à ajouter au fichier de configuration de l'agent aura une apparence similaire à celle-ci (dans une seule ligne) :

```

module_plugin C:\Users\artica\Documents\Producto\UX-Trans\ux\pandora_ux_x64.exe
-exe C:\PDR\pdr.bat -args -r -script C:\PDR\calc.sikuli -folder C:\PDR\ -
ss_config active -checkpoint -post "taskkill /F /IM calc.exe"

```

Comme vous pouvez le voir, dans cette exécution, vous avez plus d'options. Le paramètre `-checkpoint` montre une capture du résultat final même s'il n'y a pas d'échec ; `-post` exécutera les actions après la reproduction de la session. Dans ce cas, nous allons fermer le processus de calculatrice initié par l'enregistrement, pour éviter que le système finit par échouer à cause d'excès de fenêtres ouvertes ou processus similaires.

Bien que l'agent a la ligne d'exécution `module_plugin lista`, vous devez le lancer en mode processus, exécutant le suivant depuis la ligne de commande en tant qu'administrateur :

```
"%ProgramFiles%\pandora_agent\PandoraAgent.exe" --process
```

Lors du lancement vous pourrez voir comment les actions sont exécutées automatiquement selon l'enregistrement. Depuis ce moment, le processus d'agent ne doit pas se fermer et vous ne devez pas vous connecter à la machine par bureau distant, ou les exécutions pourraient être interrompues. La machine ne doit pas être manipulée, donc il est recommandé de l'utiliser dans des machines virtuelles.

Si vous avez déjà des vérifications d'automatisation, vous pouvez les inclure dans Pandora FMS avec l'exécution suivante.

```
pandora_ux.64 -exe <exe of the automation system> -args <system arguments> -script <test file path>
```

- `<exe of the automation system>`: exe (exécutable) du système d'automatisation.
- `<system arguments>`: Arguments du système.
- `<test file path>`: Chemin d'accès au fichier de test.

Exécution fondée sur transaction

Si vous avez de différents processus enregistrés avec PDR et vous avez vérifié qu'ils fonctionnent lorsque vous les reproduisez continuellement, faites l'exécution de la manière suivante :

```
C:\Users\artica\Documents\Producto\UX-Trans\ux\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
```

Comme vous pouvez le voir, tout simplement indiquez le chemin du nouveau `script` dans le paramètre `-script` séparé par une virgule comme du script précédent et utilisez le paramètre `-t` avec le nom de la transaction qui comprend les différentes étapes. Si vous avez un processus avec plus d'étapes, suivez la même logique, par exemple :

```
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
```

La ligne à ajouter au fichier de configuration de l'agent, pour ce cas serait la suivante :

```
module_plugin C:\Users\artica\Documents\Producto\UX-Trans\ux\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"
```

Grâce au paramètre -checkpoint vous pouvez voir des captures du résultat finale de chaque étape dans la console Pandora FMS.

Les modules suivants seront retournés par étape :

- UX_Time_nom_projet.étape_ordre.
- UX_Status_nom_projet.étape_ordre.

S'il y a quelque étape avec erreur, le module suivant sera aussi créé :

- UX_Snapshot_nom_projet.étape_ordre.

Une image de la web sera aussi montrée au moment de l'erreur, au cas où il se produit.

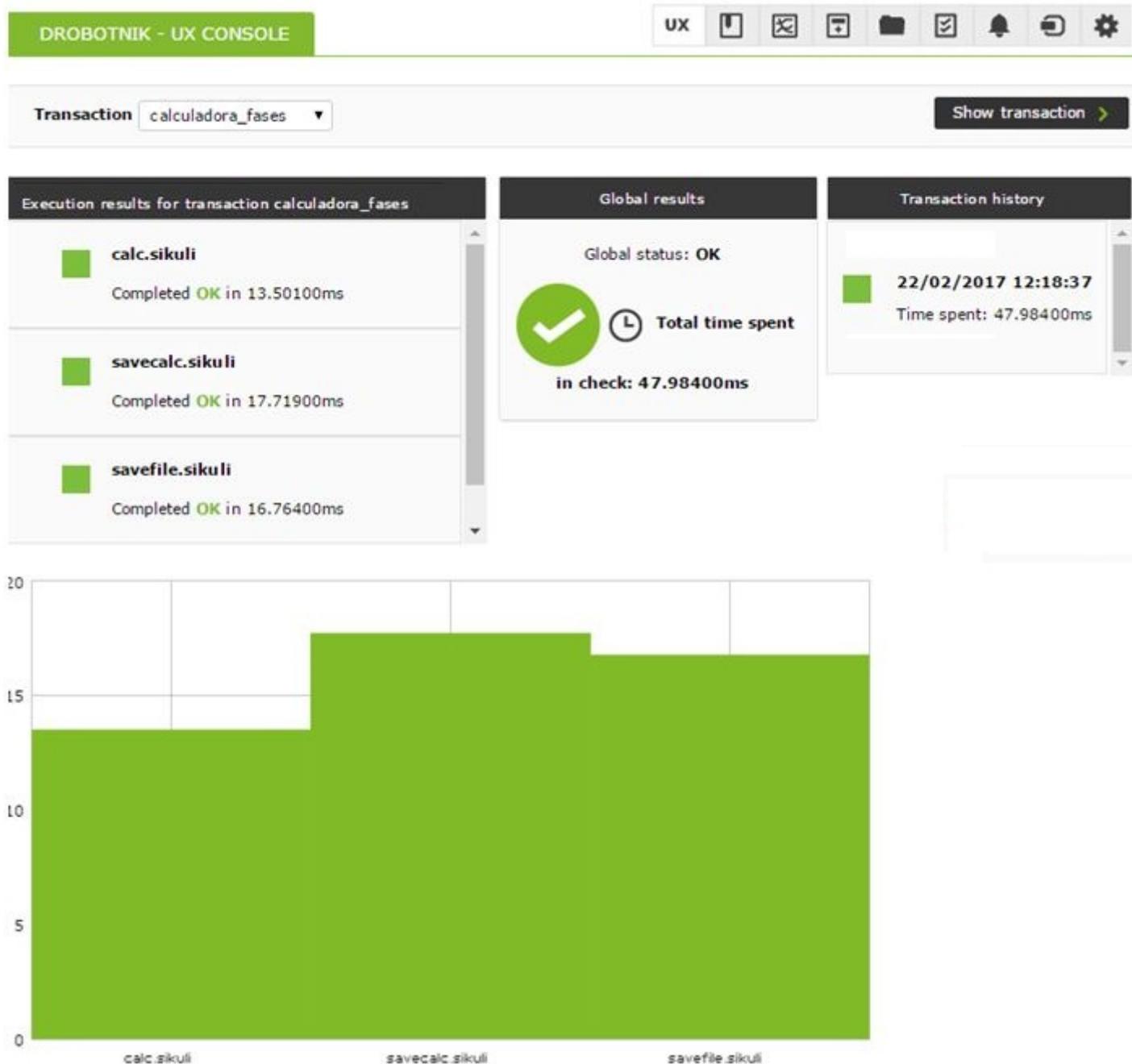
Les modules de résumé globales seront aussi retournés indetifiés ave les noms suivants :

- UX_Global_Time_nom_projet.
- UX_Global_Status_nom_projet.
- UX_Global_Snapshot_nom_projet.

Lorsque l'agent s'exécute en mode processus et les modules sont récupérés par Pandora FMS vous le verrez dans la console. À nouveau, par le biais de la vue en mode hiérarchique vous pourrez montrer clairement la relation entre les modules montrés, limitant clairement les différentes étapes :

	UX_Global_Status_calculadora_fases	Test OK		N/A - N/A	1			5 seconds
	└ UX_Global_Time_calculadora_fases	Test OK		N/A - N/A	48			5 seconds
	└ UX_Status_calc.sikuli_0	C:\PDR\calc.sikuli execution completed Control snapshot rebu...		N/A - N/A	1			5 seconds
	└ UX_Time_calc.sikuli_0	C:\PDR\calc.sikuli execution completed Control snapshot rebu...		N/A - N/A	13.5			5 seconds
	└ UX_Control_Snapshot_calc.sikuli_0	Control image rebuilt		N/A - N/A				5 seconds
	└ UX_Status_savecalc.sikuli_1	C:\PDR\savecalc.sikuli execution completed Control snapshot ...		N/A - N/A	1			5 seconds
	└ UX_Time_savecalc.sikuli_1	C:\PDR\savecalc.sikuli execution completed Control snapshot ...		N/A - N/A	17.7			5 seconds
	└ UX_Control_Snapshot_savecalc.sikuli_1	Control image rebuilt		N/A - N/A				5 seconds
	└ UX_Status_savefile.sikuli_2	C:\PDR\savefile.sikuli execution completed Control snapshot ...		N/A - N/A	1			5 seconds
	└ UX_Time_savefile.sikuli_2	C:\PDR\savefile.sikuli execution completed Control snapshot ...		N/A - N/A	16.8			5 seconds
	└ UX_Control_Snapshot_savefile.sikuli_2	Control image rebuilt		N/A - N/A				5 seconds

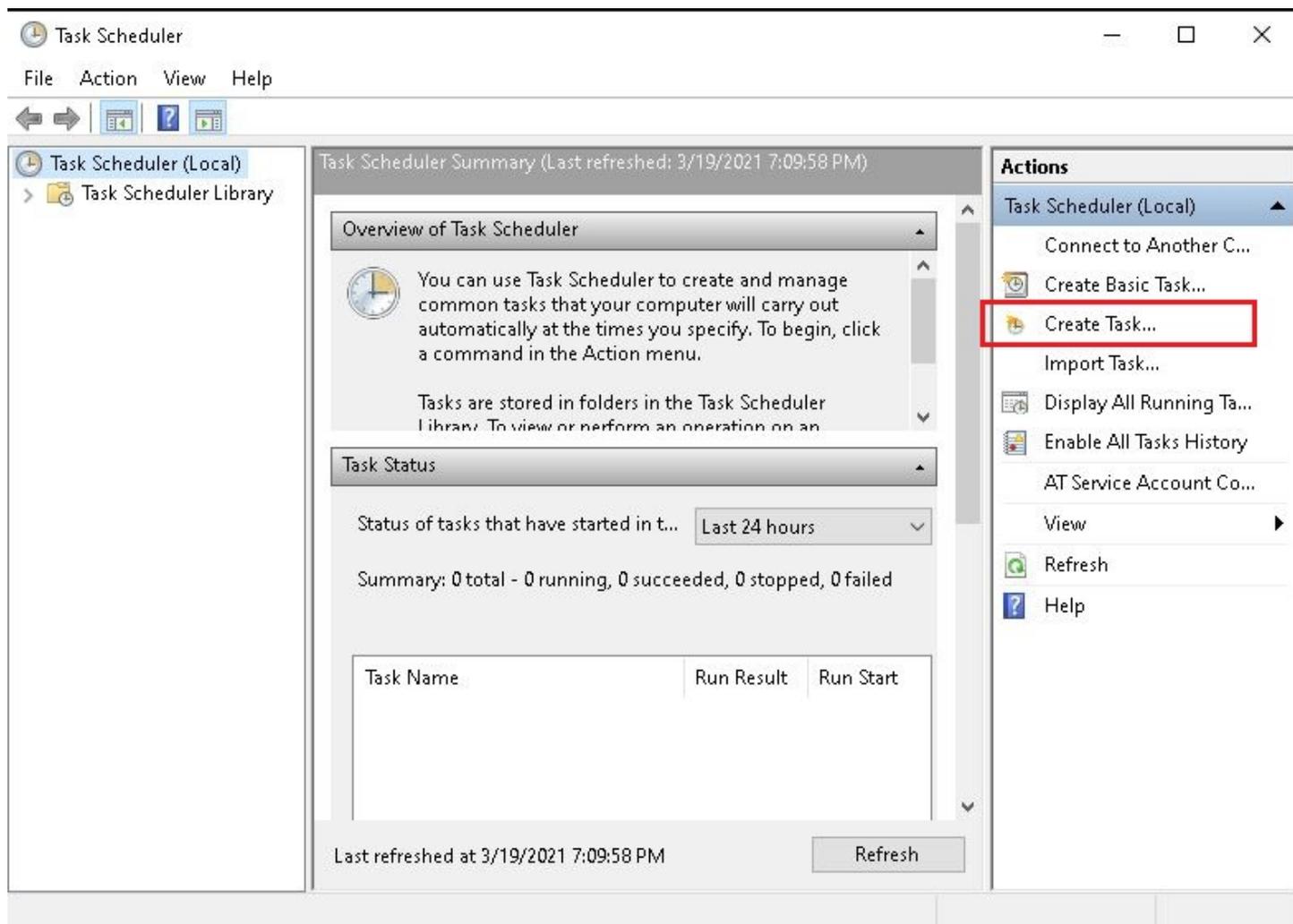
Et dans le viseur des transactions, vous pouvez voir le détail des étapes et la graphique avec les temps :



Les captures d'erreur seulement seront montrées lorsque le client UX (PWR) et le serveur PWR s'exécutent dans la même machine. Au cas contraire, le répertoire de livraison des images par le serveur PWR doit être accessible par le client afin de pouvoir montrer l'image Pandora FMS.

Il sera sur ces modules sur lesquels vous pourrez créer des alertes et regarder l'historique pour construire des graphiques et montrer l'évolution dans le temps des systèmes à superviser.

Il est recommandé de créer une tâche programmée qui démarre l'agent Pandora FMS en mode processus lorsque l'équipement démarre et ainsi les exécutions ne seront pas interrompues bien que le système aie un redémarrage forcé. Puisqu'il est en mode connexion automatique sans mot de passe, l'agent s'exécutera toujours bien que la machine est redémarrée.

 Task Scheduler

Task Scheduler

File Action View Help

Task Scheduler (Local)

Task Scheduler Library

Task Scheduler Summary (Last refreshed: 3/19/2021 7:09:58 PM)

Overview of Task Scheduler

You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.

Tasks are stored in folders in the Task Scheduler Library. To view or perform an operation on an

Task Status

Status of tasks that have started in t... Last 24 hours

Summary: 0 total - 0 running, 0 succeeded, 0 stopped, 0 failed

Task Name	Run Result	Run Start
-----------	------------	-----------

Last refreshed at 3/19/2021 7:09:58 PM Refresh

Actions

Task Scheduler (Local)

- Connect to Another C...
- Create Basic Task...
- Create Task...
- Import Task...
- Display All Running Ta...
- Enable All Tasks History
- AT Service Account Co...
- View
- Refresh
- Help

Create Task

General Triggers Actions Conditions Settings

Name: Pandora FMS Agent

Location: \

Author: WIN10JIMMY\Jimmy

Description: Starting Pandora FMS Agent, process mode.

Security options

When running the task, use the following user account:

WIN10JIMMY\Jimmy Change User or Group...

Run only when user is logged on

Run whether user is logged on or not

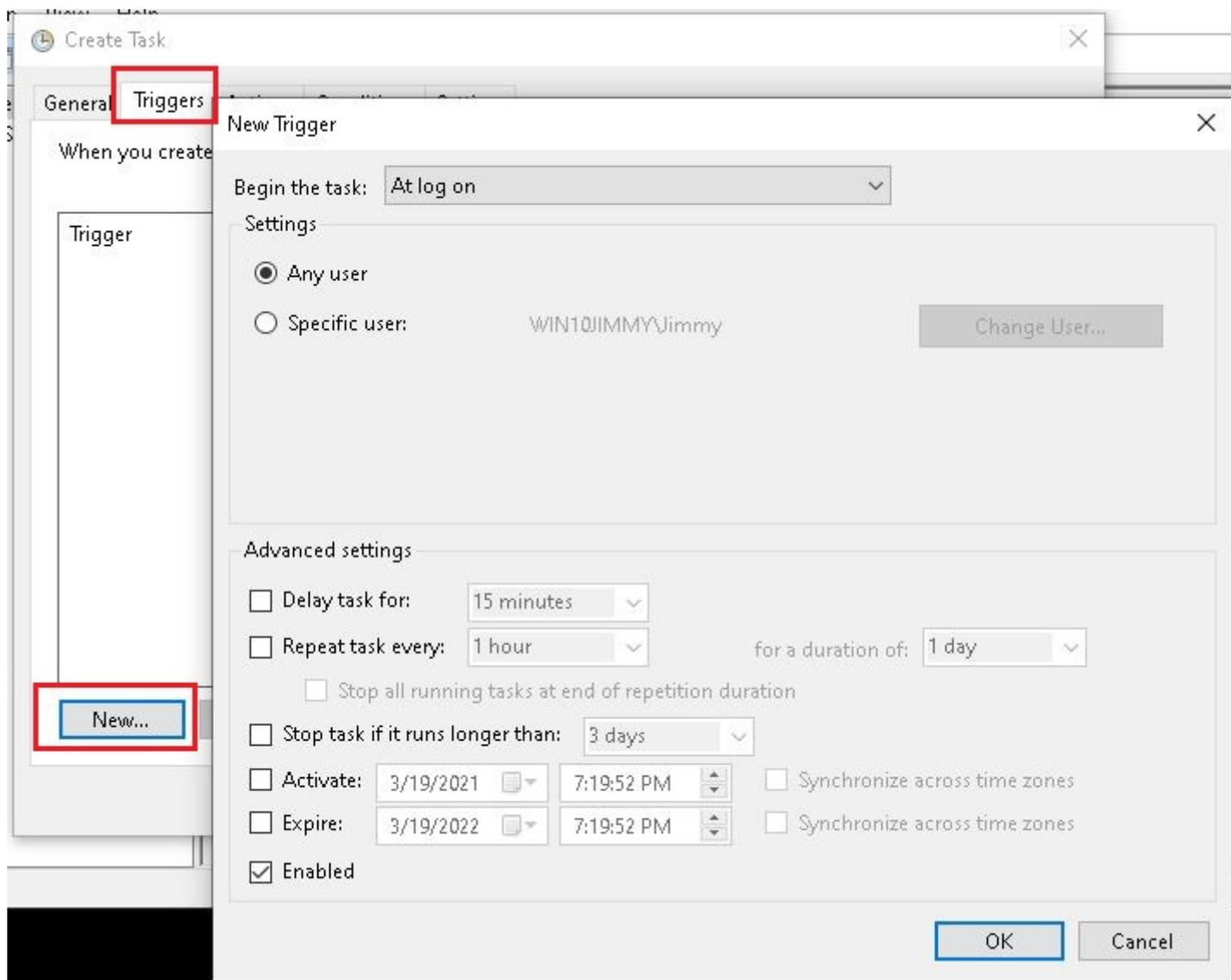
Do not store password. The task will only have access to local computer resources.

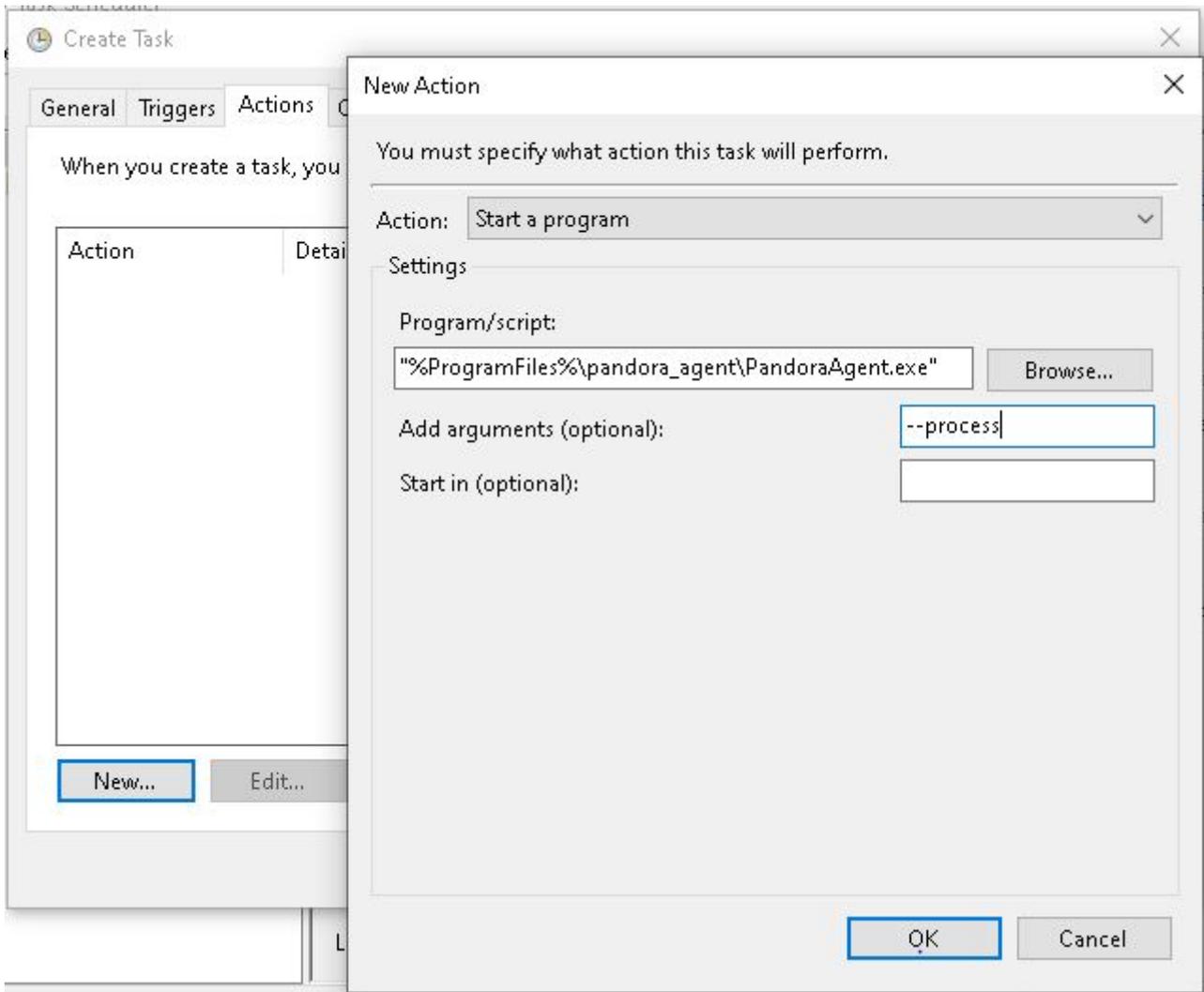
Run with highest privileges

Hidden

Configure for: Windows® 7, Windows Server™ 2008 R2

OK Cancel





Create Task

General Triggers Actions Conditions Settings

Specify the conditions that, along with the trigger, determine whether the task should run. The task will not run if any condition specified here is not true.

Idle

Start the task only if the computer is idle for: 10 minutes

Wait for idle for: 1 hour

Stop if the computer ceases to be idle

Restart if the idle state resumes

Power

Start the task only if the computer is on AC power

Stop if the computer switches to battery power

Wake the computer to run this task

Network

Start only if the following network connection is available:

Any connection

OK Cancel

Create Task

General Triggers Actions Conditions Settings

Specify additional settings that affect the behavior of the task.

Allow task to be run on demand

Run task as soon as possible after a scheduled start is missed

If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

Stop the task if it runs longer than: 3 days

If the running task does not end when requested, force it to stop

If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

OK Cancel

De cette manière nous garantissons que dans ce système Windows l'agent Pandora FMS s'exécutera en mode processus même si la machine se redémarre, en envoyant les informations collectées par la sonde PDR.

Systèmes avec quelques bureaux peuvent devenir problématiques, donc il est toujours recommandée d'utiliser la configuration décrite précédemment, dans une machine en mode bureau avec autodémarrage et un seul bureau.

[Retour à l'index de documentation du Pandora FMS](#)