



ユーザエクスペリエンス監視 (UX および WUX)



om:

<https://pandorafms.com/manual/!778/>

Permanent link:

https://pandorafms.com/manual/!778/ja/documentation/pandorafms/monitoring/13_user_monitorization

2024/12/03 19:30



ユーザエクスペリエンス監視 (UX および WUX)

[Pandora FMS ドキュメント一覧に戻る](#)

概要

ユーザエクスペリエンス監視は、ウェブブラウジング (PWR / WUX) と Windows システムのデスクトップアプリケーション (PDR) における操作を自動的に記録する仕組みから成ります。これらの記録は、ブラウザ上でのマウスクリック、テキスト入力、ウェブブラウジングでの検索、デスクトップ上でのアプリケーションを開くことなどです。記録した処理を自動的に実行し、処理にかかった時間などの結果を得ることができます。これにより、後で結果を検索して実行できるように自動的に記録することができます。

UX および WUX 監視の違い

両方の監視の仕組みは Pandora Web Robot Daemon (PWRD) システムを通してウェブブラウジングタスクを実行するために利用されます。

UX システムは、マシンにインストールされたソフトウェアエージェントによって監視を実現します。WUX は Pandora FMS の内部に統合されたサーバを用いて監視をします。

Pandora Web Robot Daemon (PWRD) は、ウェブブラウジングセッションを自動実行するために必要なサービスです。これは、ユーザの Web ポータルをナビゲートするのに必要なアクションのリストを含むファイルを使用します。

加えて UX 監視は、Windows デスクトップおよびシステムアプリケーションの自動化タスクを実行することができます。このタイプのタスクは、WUX では実行できません。

Pandora FMS UX は、自動化されたタスクを実行するシステムであり Pandora FMS に実行結果、所要時間、およびエラーが検出された可能性のあるスクリーンショットを含むレポートを提供します。

UX 向けの Pandora FMS PWR テクノロジーは、ウェブサイトの閲覧、ユーザアクションのシミュレーションを可能にします。コンテンツ、テキストなどを確認できます。これは、非フラッシュベースの Web サイトおよびアプリケーションを監視するのに最適です。利点としては、より高速でバックグラウンドで実行できることです。

タスクの自動化システムがある場合 Pandora FMS UX では、すでにあるスクリプトを実行し、その実行を監視することもできます。

どのタイプの監視を行うかを指定することが重要です。どちらの方法も、対応するマシンにインストールされているソフトウェアエージェントのプラグインによる監視実行に基づいています。

ウェブランザクションの実行には Selenium エンジンを利用します。

- Selenium バージョン 2.
- Selenium バージョン 3.

PWR UX 監視

UX および WUX 監視の違いを考慮して UX の監視は、監視対象の環境に関連する一連の前提条件に従う必要があります。

- Java® のインストール
- Mozilla Firefox® のプロファイル設定
- Selenium® サービスのインストール
- システム上への PWR の展開
- Mozilla Firefox® 用の Selenium® IDE の展開
- PWR セッションの保存

事前に保存されたセッションの標準的な実行には、Pandora UX プラグインの最新バージョン 28-04-2022 を使用します。

Selenium 3 による Web User Experience (WUX)

WUX サーバでの Selenium デプロイでは、コンテナベースのスタックを使用して、迅速なデプロイと容易なスケーリングを実現します。

事前設定

Docker と Docker Compose が事前にインストールされている必要があります。CentOS を基本オペレーティングシステムとして使用することをお勧めします。

このインストールは、以下にある Docker ドキュメントに従うことをお勧めします。

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

公式 Selenium イメージは、スタックのインストールと展開に使用されます。それらは次の場所で見つけることができます。

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

Selenium リポジトリでは、ブラウザを使用したさまざまなイメージがあります。Pandora FMS の場

合は、Firefox® および Chrome® コンテナを推奨します。

Selenium スタックのデプロイ

Selenium スタックをデプロイするには、まず必要な構成を含む YAML ファイルを作成する必要があります。

```
# 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
```

前の例に対して、メモリ制限、タイムゾーン、ノード数など、ケースごとに必要な変更を行います。

設定した内容でコンテナを有効化するには、次のコマンドを実行します。(<compose-file> は、選択した名前に置き換えてください。)

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

コンテナで実行されているサービスを確認する必要がある場合は、次のコマンドを使用します。

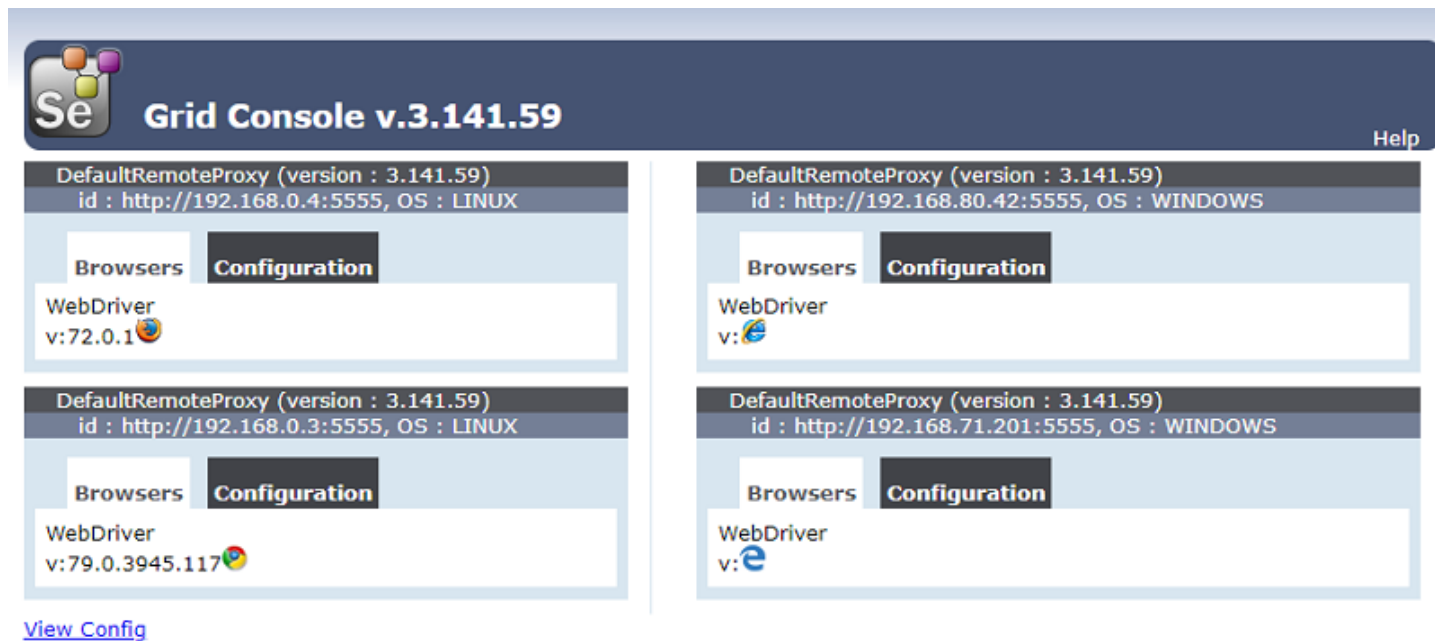
```
docker-compose -f <compose-file> ps
```

Selenium サービスの状態とログを表示するには、次のコマンドを使用します。

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

適切なチェックが行われた後、グリッドが適切に機能し、ワーカーが設定ファイルでの定義どおりにサインアップしたことを確認するには、次のURLにアクセスします。

```
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 located at the top right. The main area is divided into four panels, each representing a node. Each panel has a header with 'DefaultRemoteProxy (version : 3.141.59)' and an 'id' field. Below the header are two tabs: 'Browsers' and 'Configuration'. The 'WebDriver' version is displayed in each panel.

Node ID	OS	WebDriver Version
http://192.168.0.4:5555	LINUX	v:72.0.1
http://192.168.80.42:5555	WINDOWS	v:72.0.1
http://192.168.0.3:5555	LINUX	v:79.0.3945.117
http://192.168.71.201:5555	WINDOWS	v:79.0.3945.117

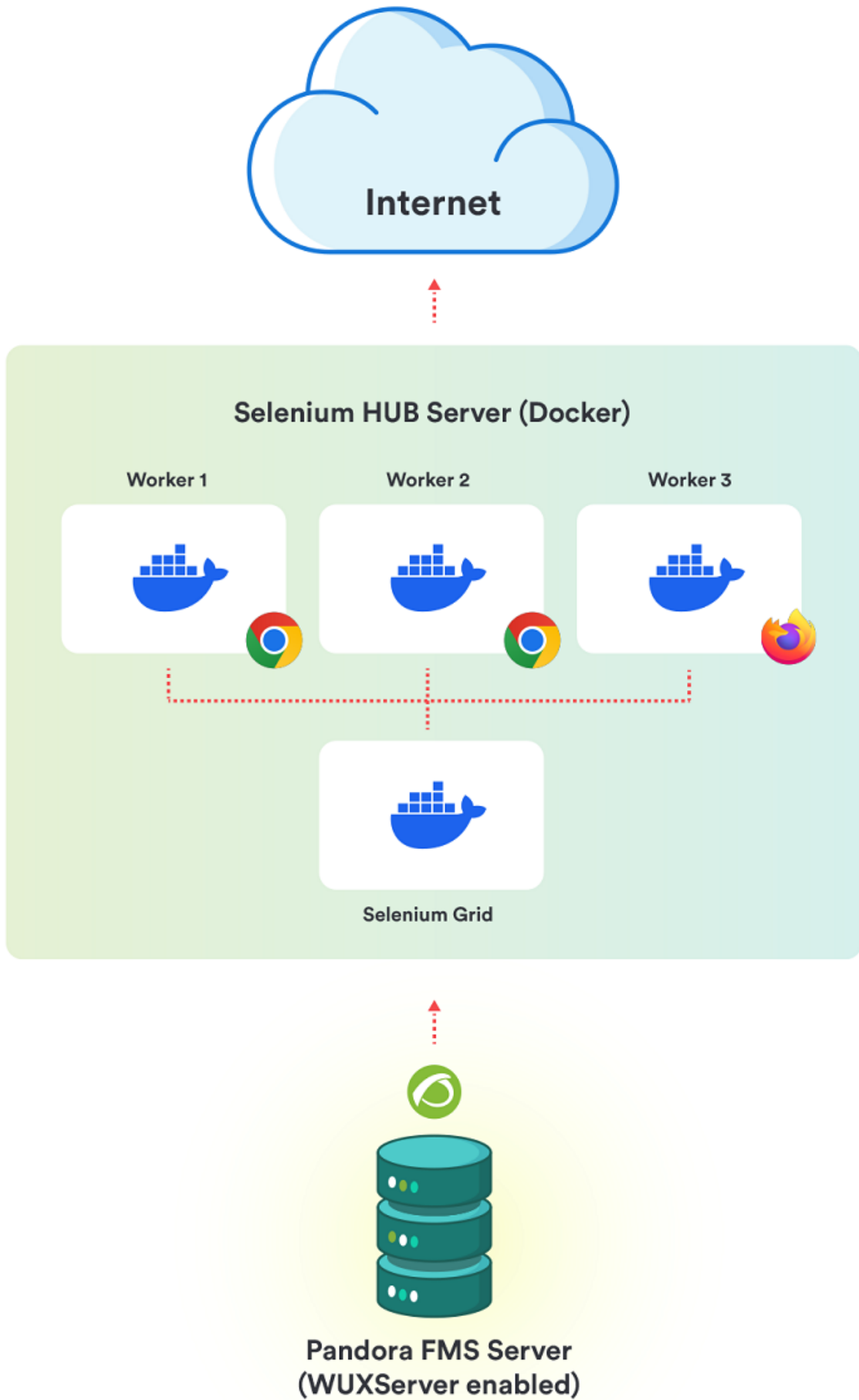
[View Config](#)

ワーカー数を増やす必要がある場合は、次のコマンドを実行します。

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

Selenium サービスインフラストラクチャ

Selenium は、必要なワーカーコンテナをデプロイするグリッドとして機能するコンテナが有効になっているハブとして機能します。



Pandora FMS 設定

集中モード(WUX)を使用するにはPandora FMS サーバに以下の設定を適用する必要があります。

選択したモードに関係なくWUX サーバ設定パラメータを Pandora FMS サーバの設定ファイルに追加することにより、ブラウジングセッションから実行の割り当てができます。

ファイル/etc/pandora/pandora_server.confの最後に次の設定を追加します (<ip_wux_host> をサーバのIPアドレスに置き換えますPandora FMS が動作しているのと同じサーバの場合は、127.0.0.1 です。):

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

Pandora FMS サーバの設定ファイルには、キューに入れられたナビゲーションセッションを削除するための新しいトークンがあります。

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

Pandora FMS スレッド管理

- wuxserver スレッド管理は、pandora_server サービスを起動した際に自動的に行われます。
- これはSelenium hub にある MINOR ブラウザのノード数を考慮して行われます。例:
 - hub に 2 つの Firefox ノードと 2 つの Chrome ノードが設定されている場合、wuxserverスレッドの数は 2 になります。
 - hub に 1 つの Firefox ノードと 4 つの Chrome ノードが設定されている場合、スレッドの数は 1 になります。
 - hub に 6 つの Firefox ノードが設定されている場合、スレッドの数は 6 になります。

各スレッドは、wuxserver から Selenium hub に同時に送信できるセッションを示していることに注意してください。

付録: Internet Explorer および Microsoft Edge のワーカー追加

Microsoft ブラウザに対して Web トランザクションを起動する必要がある場合は、目的の Windows バージョンのマシン(物理または仮想)を設定し、以下の公式ドキュメントに従ってドライバーを設定する必要があります。

Internet Explorer ドライバインストールドキュメント:

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

32bit のバージョン 3.141 のドライバの利用をお勧めします

64bit 版はパフォーマンスの問題があります。

Microsoft Edge ドライバーインストールドキュメント:

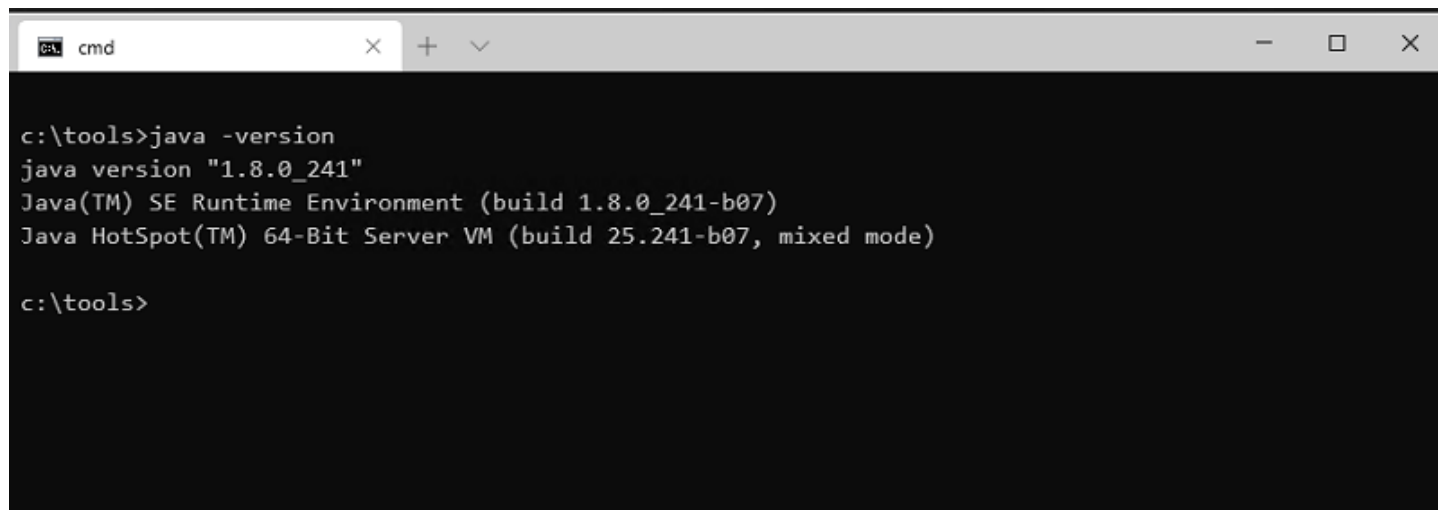
```
https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/
```

Selenium を実行するには Java が Windows デバイスにインストールされている必要があります。

Java がインストールされているかどうかを確認するには、次のコマンドを実行します。

```
java -version
```

次のような出力が得られます。



```
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>
```

サーバをローカルで実行してグリッドに登録するには Selenium JAR ファイルも必要です。

以下から入手できます。

```
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 [3.141.59](#)

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](#)

Microsoft Edge サーバを有効化するには、JAR ファイルがあるディレクトリのターミナルで次を実行する必要があります。

```
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"
```

このコマンドは Internet Explorer サーバを有効化するのと似ていますが、ダウンロードしたドライバーのパスを指定する必要があります。

```
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
|
```

Windows ファイアウォールは、コマンド実行時に指定されたポートの通信を許可するように設定する必要があります。例では 5555 および 4444。

記録

Selenium 2 で作成された記録は正しく機能しない場合があります。ことに注意してください。

Selenium バージョン 3 では、新旧両方の記録の両方がサポートされます。

新しいセッションを記録するには、目的のブラウザに Selenium IDE 拡張機能をインストールする必要があります。

Firefox

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

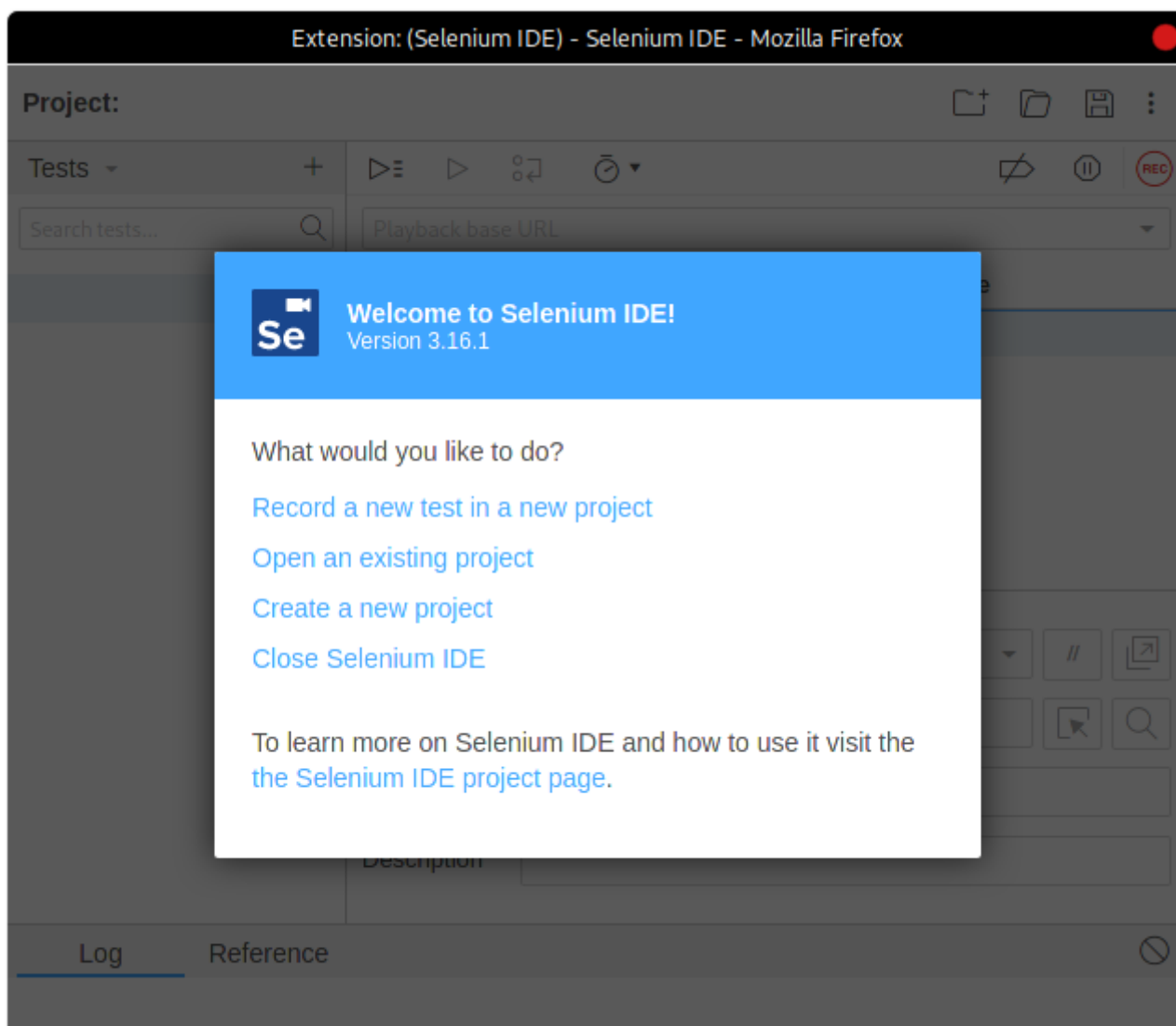
Chrome

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

次のアイコンは、拡張機能をインストールしたあとに表示される、記録を開始するアイコンです。



新しい記録プロジェクトを開始するためのメニューが開きます。



記録が行われると、次のような結果になります。

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

Pandora FMS には Selenium の機能とは別に、監視対象から情報を取得するためのカスタムコマンドがあります。

コマンド

利用可能なコマンドの一覧は、[こちら](#)を確認してください。

録画を行う際の推奨事項

次に、Selenium IDE とのトランザクションを記録し、最も複雑なコマンドを Pandora FMS と統合するための、一連の推奨事項とアドバイスについて詳しく説明します。

- 可能な場合はいつでも、トランザクションを段階的に分割します。このようにして、状態、時間、スクリーンショットで作成されたモジュールがセグメント化され、トランザクションが失敗したときにそれらを見つけやすくなります。
- 誤検知を回避するには Selenium コマンド “set speed” および “wait for” を使用します。トランザクションを実行するとき Selenium コマンドには、1つのコマンド実行終了から次のコマンドが実行されるまでのデフォルトの遅延がなく、一部のコマンドにもタイムアウトがありません。これにより、トランザクションは可能な限り最短時間で実行されますが、非常に高速に実行することにより Web がやや遅かったり、ロードに数秒余分にかかる場合、チェックが失敗する可能性があります。たとえば “click” コマンドを実行して他のページに移動した後、新しいページの要素を使って動作するコマンドがあった場合、読み込みが 1秒遅れると、その要素は見つかりません。新しいページのチェックは失敗に終わります。このような状況を回避するために、コマンド “set speed” があります。これは、各コマンドの間に Target フィールドに指定されたミリ秒数の遅延を追加します。トランザクションの開始時に設定することをお勧めします。ページの読み込みや要素の表示に数秒かかることがわかっている場合は、コマンド “wait for element present” もあります。“wait for visible” と “wait for test” では、トランザクションを失敗として判断する前に、要素がページに表示されるのを待つ時間をミリ秒単位で設定できます。これらのコマンドを使用すると、チェックの信頼性が大幅に向上しますが、トランザクションの実行にかかる時間も長くなることを考慮することが重要です。
- 既存の要素を確認します。このためには “assert” や “verify” などのコマンドをさまざまな場面で使用します。たとえば、クリックでトランザクションを終了しても、クリックした要素を新しいページが開くことは保証されません。要素をクリックできることを保証するだけです。クリック後にのみ読み込まれることがわかっている “verify test” を導入すると、クリックによって送信されたページが使用可能かどうかを確認することができます。
- ウィンドウまたはタブ間を移動するトランザクションでは “store window handle” を使用します。初期ウィンドウに識別子をあらかじめ保存していない場合、ウィンドウの変更 (“select window” を使用) は失敗する可能性があります。
- Target by CSS が失敗した場合またはページ内のコンテンツを検索したい場合は、xpath を使用します。デフォルトでは Selenium IDE レコーダは CSS ロケータ要素の Target にロードしますが xpath によってロケータもロードします。レコーダの Target ボックスをクリックすると、保存されているすべてのロケータを表示できます。

Command	assert text	//	📄
Target	css=#\36 fn79666fiykxpysdro91x1ccr_message sp	📄	🔍
Value	css=#\36 fn79666fiykxpysdro91x1c cr_message span	css:finder	
Description	xpath=//div[@id='6fn796 66fiykxpysdro91x1ccr_me ssage']/div/h2/span	xpath:idRelative	
rovided value. T	xpath=//h2/span	xpath:position	
	xpath=//span[contains(, 'Innovadores de la monitorización')]	xpath:innerText	

さらに xpath を使用すると、ページのタグ内のテキストを検索して、より動的なレコーディングができます。上記のキャプチャでは、特定のロケータではなく、ページのすべてのスパンタグで “Monitoring Innovators” というテキストを検索する xpath を使用できることがわかりました。

- “execute script” コマンドの正しい利用。“execute script” コマンドは、現在選択されているフレームまたはウィンドウのコンテキストで JavaScript のスニペットを実行します。このガイド(<https://ui.vision/rpa/docs/selenium-ide/executescript>)を読んで、その使用法と提供されるさまざまなオプションについて学ぶことをお勧めします。ただし、以前に保存された変数の使用(たとえば “store text” コマンドは、Pandora FMS Web ドライバが正しく解釈できるように、二重引用符で囲む必要があります。これは “store text” で保存され、後で “execute script” で使用されるように Pandora FMS サーバが正しく解釈できるようにする変数の例です。引用符で囲まれた変数をこのように使用すると Selenium IDE レコーダではスクリプトを実行できないことに注意してください。

```
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
```

Selenium 2 による Web User Experience (WUX)

事前設定

Selenium

Pandora ウェブロボットデーモン (PWRD) の展開

Pandora Web Robot Daemon は、ユーザーの Web アクセスをナビゲートするために必要なアクションのリストを含むファイルを使用して Web ブラウザセッションを自動化するツールを提供するサービスです。

Pandora サーバに統合され PFMS サーバ(GNU/Linux®) またはモジュールライブラリ(Windows®) のインストール時に /usr/share/pandora_server/util/pwrд に置かれています。または、モジュールライブラリからダウンロードできます(Windows®)

ダウンロードは、<https://pandorafms.com/library/pandora-ux-and-wux-pwr-recorder/> から行います。

以下が含まれます。

- Windows 版 Firefox 47 のバイナリ
- Web ブラウザセッションを実行するために、あらかじめビルドされたプロファイル
- セッション自動化のためのサーバ (Selenium サーバ)

Windows への Selenium サーバの展開

事前準備:

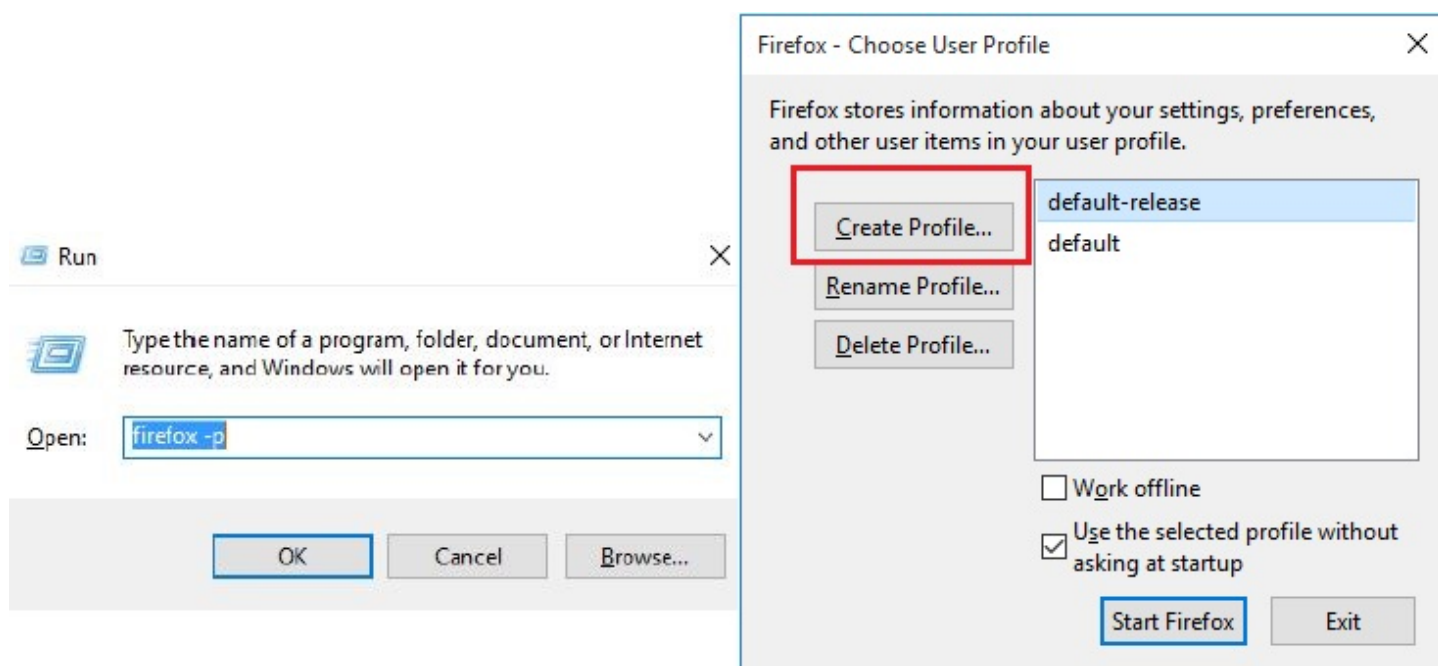
- サービスを提供するサーバへ Java 1.8 をインストール
- Firefox 47.0.1 (<https://ftp.mozilla.org/pub/firefox/releases/47.0.1/> からダウンロード)
- 自動セッションチェックのための Firefox プロファイル(オプション):
<http://toolsqa.com/selenium-webdriver/custom-firefox-profile/>
- ディレクトリの作成: C:\PWR

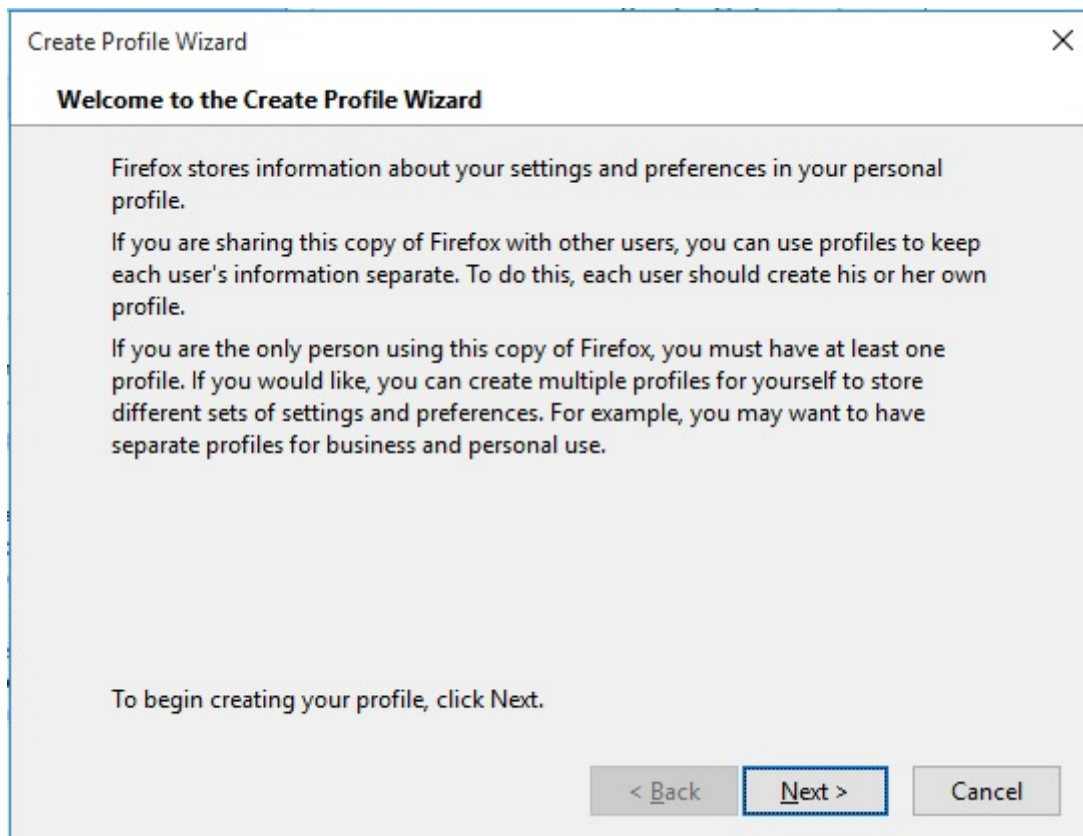
PWR_Server.zip のダウンロードは、<https://pandorafms.com/library/pwr-server-for-ux-monitoring/> から行います。

ファイルの展開:

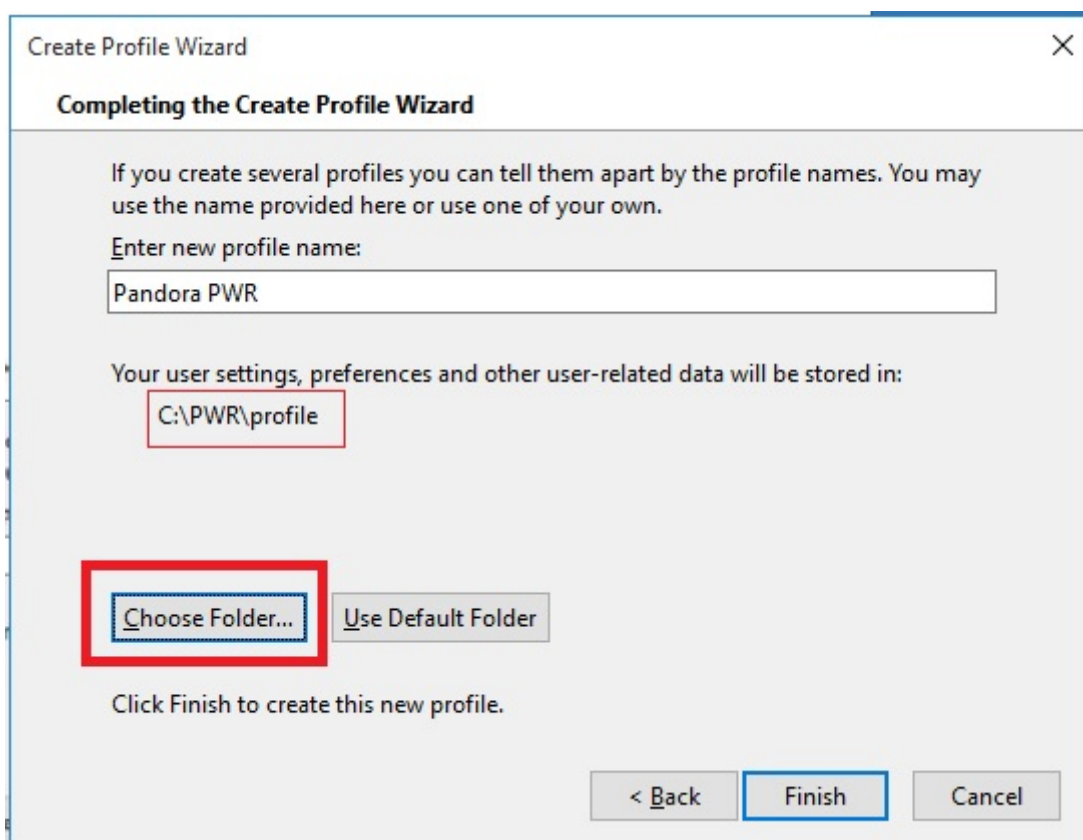
- PWR_Server.zip を C:\PWR\ へ展開
- Firefox プロファイルを C:\PWR\profile へ展開

プロファイルを使用してチェックを実行することは必須ではありませんが、プロキシを使用する場合や、パスワードのオートコンプリートを使用する場合は特に推奨します。Firefox のプロファイルを作成するには、次のようにします。

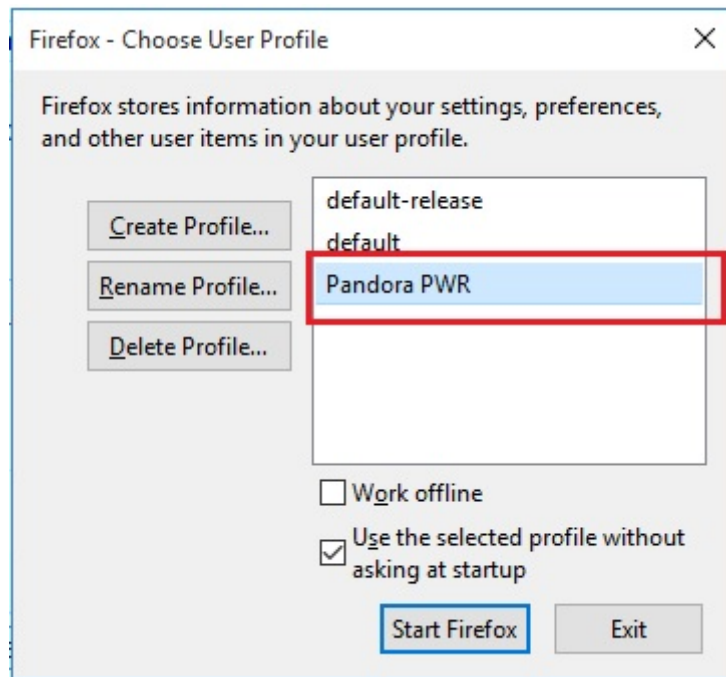




ディレクトリを選択します。



プロキシやポップアップなどのオプションを設定するために、新たなプロファイルで Firefox を起動します。



次に、提供された BAT ファイル `service_installer.bat` を実行してサービスをインストールします。 `service_installer.bat` が正しく機能するように、内容を変更し、存在するパスにインストールされるようにします。たとえば `java` は、PATH が正しく構成されている場合にのみ動作します。それ以外の場合は、ファイル内にフルパスを指定する必要があります。最後に、サービスを開始します。

```
net start PWRSRV
```

Selenium サーバが起動します。デバッグのために(サービスを中断した時は)手動でも実行できます。次のコマンドを使います。

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

Linux への Selenium サーバの展開

事前準備:

- サービスを提供するサーバへ Java 1.8 をインストール
- Firefox 47.0.1 (<https://ftp.mozilla.org/pub/firefox/releases/47.0.1/> からダウンロードできます) をインストール
- 自動セッションチェックのための Firefox プロファイル(オプション): <http://toolsqa.com/selenium-webdriver/custom-firefox-profile/> を準備
- `xorg-x11-server-xvfb` をインストール
- `java` をインストール

PWRD デーモンの Selenium コンポーネントは Java を要求します。以下のようにインストールします。

```
yum install java
```

PWRD で、Linux サーバで Firefox を起動できるようにするには、仮想グラフィック環境を作成するために xorg-x11-server-Xvfb, gtk2 and gtk3 をインストールする必要があります。

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

リポジトリに無い場合は、rpm ファイルは以下から入手できます。

```
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
```

rpm パッケージを手動でインストールするには次のようにします。

```
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
```

必要なパッケージのインストールをしたら `install_pwr.sh` のインストールに進みます。デフォルトのパスは、`/usr/share/pandora_server/util/pwr/install_pwr.sh` で、次のように実行します。

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

インストールが完了したらサービスを開始します。

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

以下のスクリプトで Selenium サーバを起動できます。

```
#!/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
```

または、手動で以下のコマンドを実行します。

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

バージョン 730 以降では、デフォルトのインストールとは異なるユーザとディレクトリでカスタムインストールを実行できるようになりました。

PWRD 動作モード

PWRD には複数の動作モードがあります。

- Standalone: このモードでは PWRD の単一のインスタンスを起動します。インスタンスは Pandora FMS サーバに割り当てられます。
- HUB: このモードでは PWRD サービスはブラウザセッションを直接は評価せず、代わりにタスクを実行するためのノードが登録されます。これは PWRD サービスのクラスタモードです。それぞれの HUB が Pandora FMS サーバに割り当てられます。

Standalone モードの PWRD

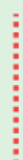
Standalone モードの PWRD は、デーモンを起動し、WUX サーバを通してユーザが指示したアクションを実行できるようになります。



WUXServer PWRD Standalone



PWRD



Pandora FMS Server
(WUXServer enabled)

起動:

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

状態確認:

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

停止:

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

PWRD の "HUB" モード

HUB モードは、ロードバランサーとしてデーモンを起動します。この動作モードでは、システムに登録されたすべてのノード間で負荷をバランシングし、負荷に応じてセッションの実行割り当てをします。

HUB の管理コンソールにアクセスすることにより、HUB の状態を常に確認することができます。

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



WUXServer - PWRD HUB

Node 1



Node 2



Node 3



PWRD



Pandora FMS Server
(WUXServer enabled)

起動:

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

状態確認:

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

停止:

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

PWRD ノードの HUB への追加

新たな PWRD ノードを追加するには、以下が必要です。

- HUB (HUB モードの PWRD)
- 同一または異なるマシン上の PWRD ファイル
- ノードをホストするコンピュータから HUB をホストするコンピュータへの TCP/4444 接続

この動作モードでは、サービスはキューに入れられた HUB からのすべての要求を処理し、実行結果を返します。HUB のみが WUXServer との通信を行い、アクションの実行は透過的です。

起動および HUB への登録の“hub” の部分は PWRD HUB のサーバ IP に置き換えてください:

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

状態確認:

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

停止:

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

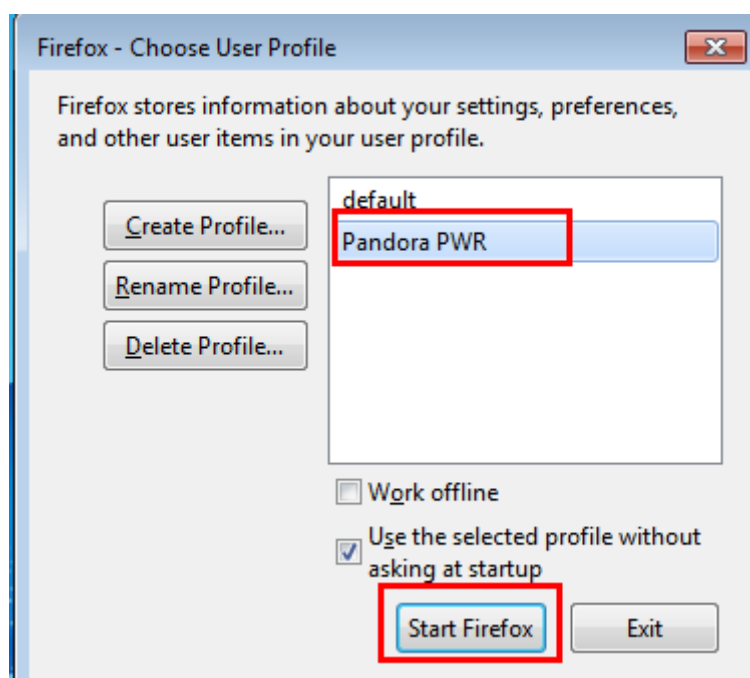
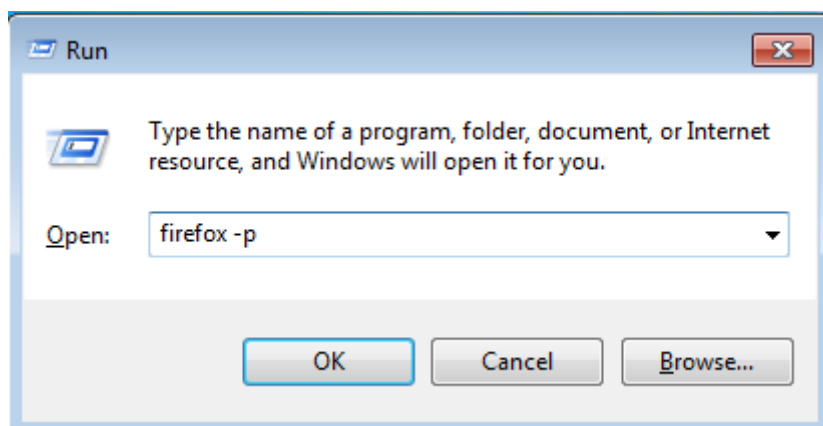
Firefox プロファイルへの証明書アップロード

Firefox のデフォルトの CA に含まれていない、自己証明書や CA による証明書を必要とする Web ページの監視を行うこともできます。そのような場合、利用する Firefox のプロファイルに証明書を読み込ませる必要があります。

それには、グラフィカルな環境でブラウザを起動し、URL にアクセスし SSL 証明書を追加するのが一番簡単です。以下に Windows および Linux でのその方法を説明します。

Windows システムにデプロイした PWRD の場合

この場合 GUI 環境があるのでチェック用のプロファイルを使って Firefox ブラウザを起動するだけです。



起動したら、アップロードする証明書を使用して URL にアクセスし、ブラウザの例外として追加します。



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

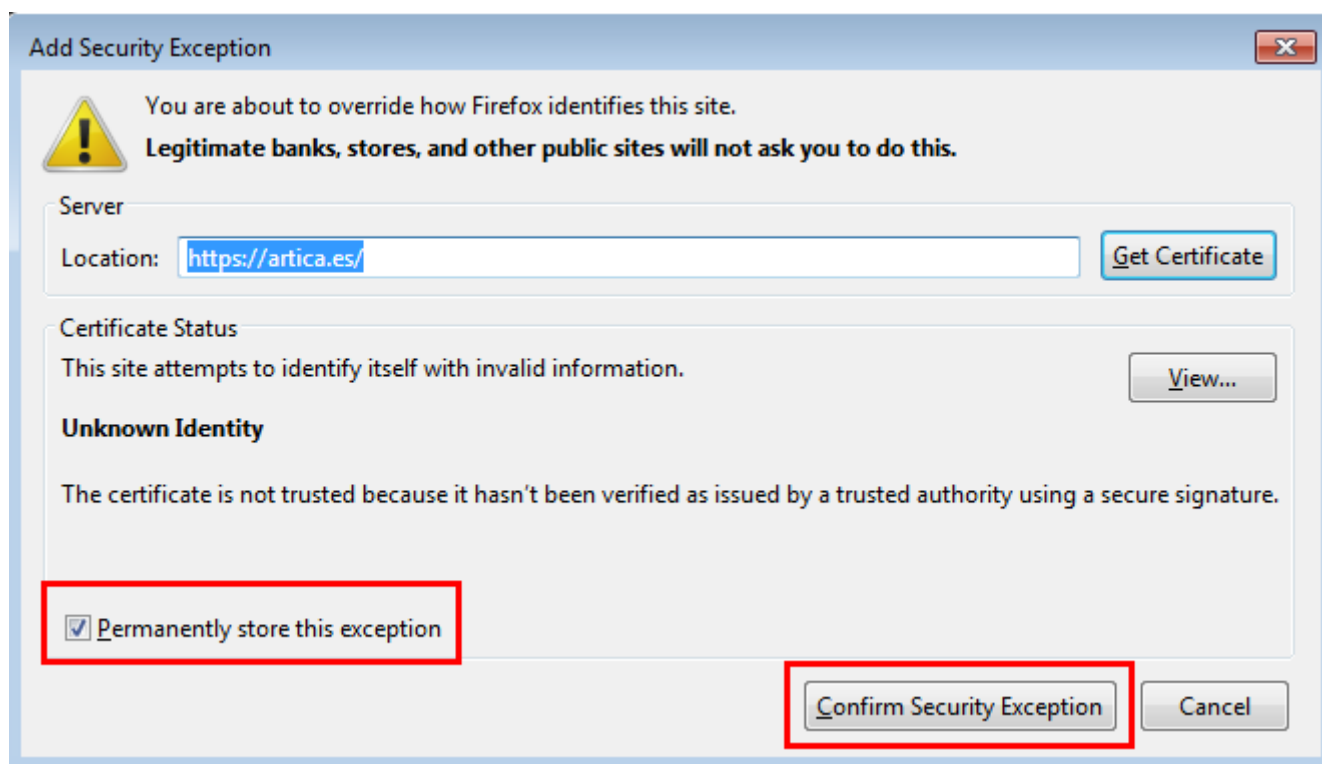
Advanced

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](#)

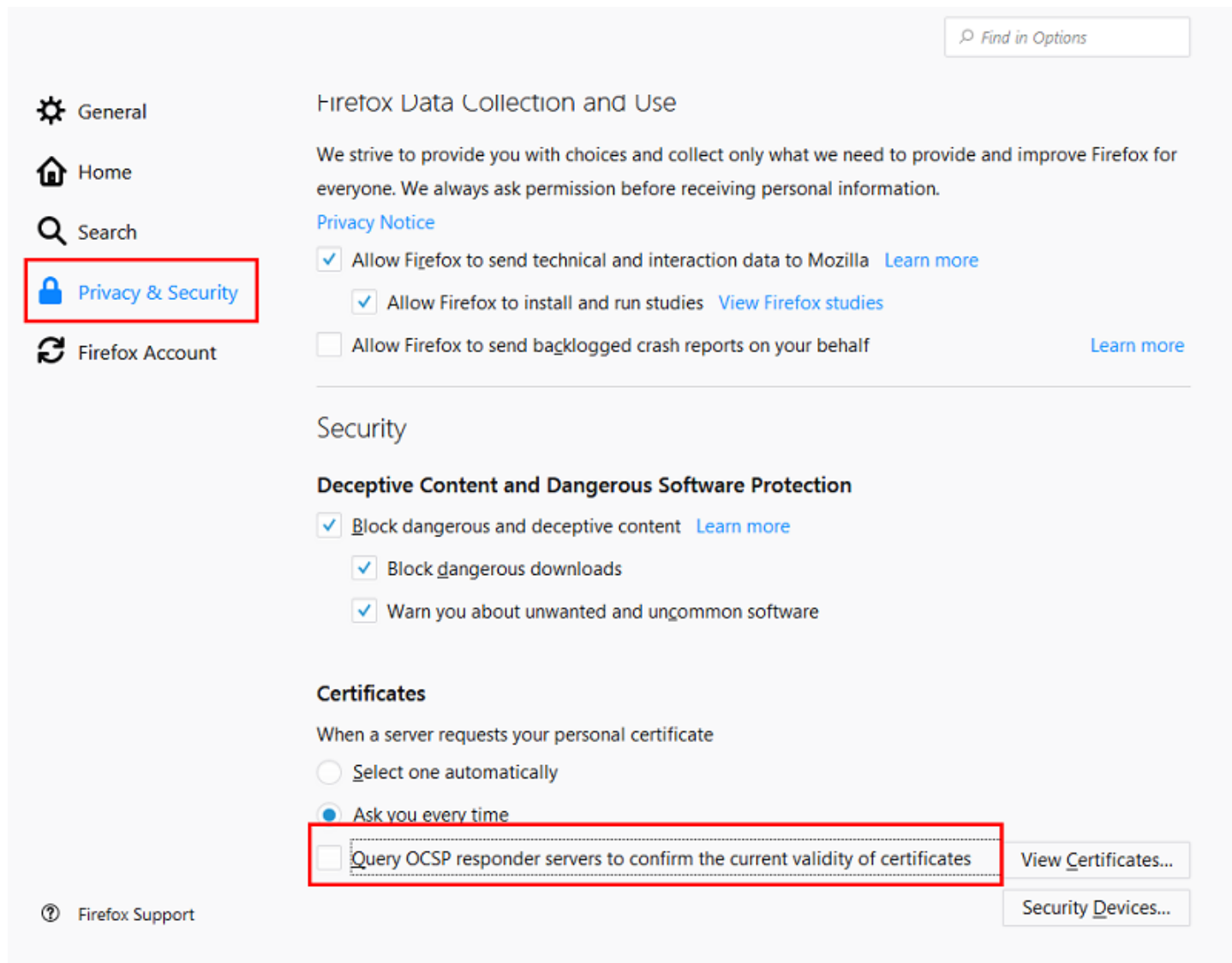
Add Exception...



別の方法として、SSL 証明書を設定するには、Firefox のオプション「Privacy & Security」タブにアクセスし、「Query OCSP responder servers to confirm the current validity of certificates」のチェックを外します。

The image shows a web browser menu with various options. The 'Options' menu item is highlighted with a red rectangular box. The menu items and their keyboard shortcuts are as follows:

- Sign in to Sync
- Content Blocking
- New Window (Ctrl+N)
- New Private Window (Ctrl+Shift+P)
- Restore Previous Session
- Zoom: 100%
- Edit (includes icons for cut, copy, and paste)
- Library (>)
- Add-ons (Ctrl+Shift+A)
- Options** (highlighted)
- Customize...
- Open File... (Ctrl+O)
- Save Page As... (Ctrl+S)
- Print...
- Find in This Page... (Ctrl+F)
- More (>)
- Web Developer (>)
- Help (>)
- Exit (Ctrl+Shift+Q)



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

Linux システムにデプロイした PWRD の場合

一般的に、Linux サーバがインストールされている場合、グラフィカルなデスクトップ環境は含まれていません。前のケースと同じ手順を実行するには X を GUI 環境を持つコンピュータにリダイレクトする必要があります。

Linux デスクトップへの X11 のリダイレクト

PWRD サーバから “-X” オプションをつけて SSH 接続をするだけで十分であるため非常にシンプルです。

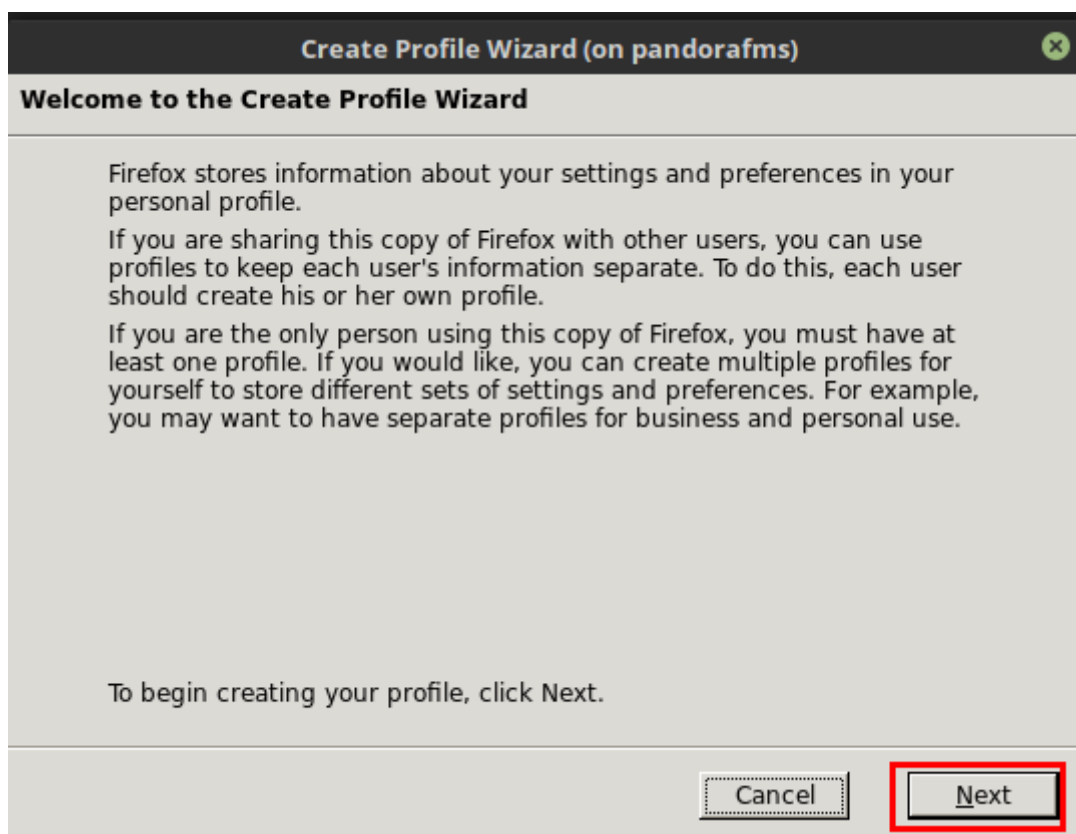
```
ssh -X user@pwrд_ip_address
```

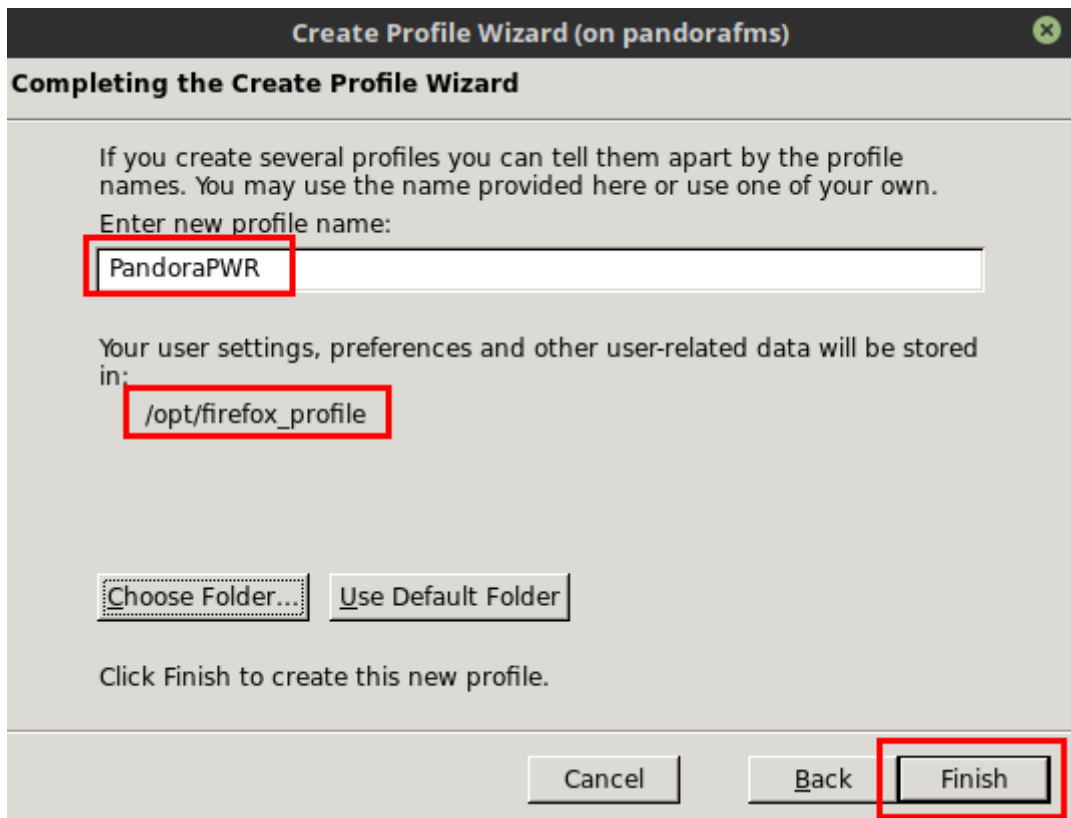
その後、Firefox ブラウザを起動すると、デスクトップに画面表示されます。

```
firefox -p
```

デフォルトのインストールでは、デフォルトのプロファイルのみ見ることができます。Windows への

インストールの場合と同様に、新しいプロファイルを作成して使用することをお勧めします。

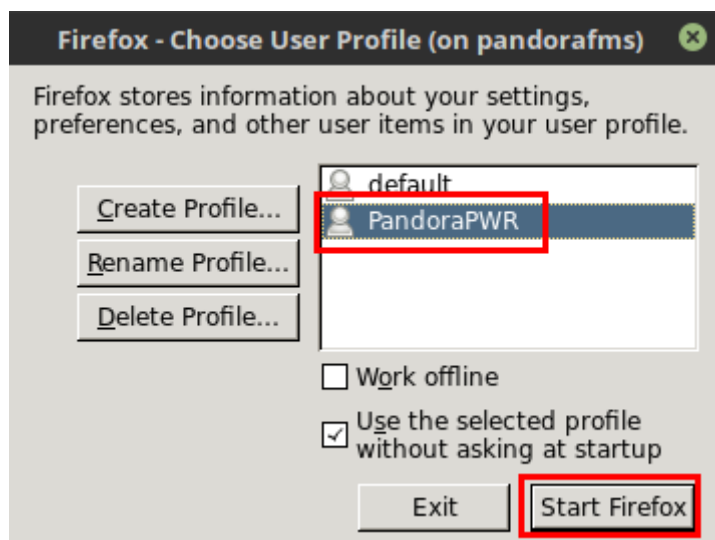




プロファイルが /opt/firefox_profile 以外にある場合、PWRD の起動スクリプト /etc/init.d/pwrdd でプロファイルのパスを指定するように編集する必要があります。

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

プロファイルを作成したら、ブラウザを起動します。



起動したら、アップロードする証明書を使って URL にアクセスし、ブラウザの例外として追加します。



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

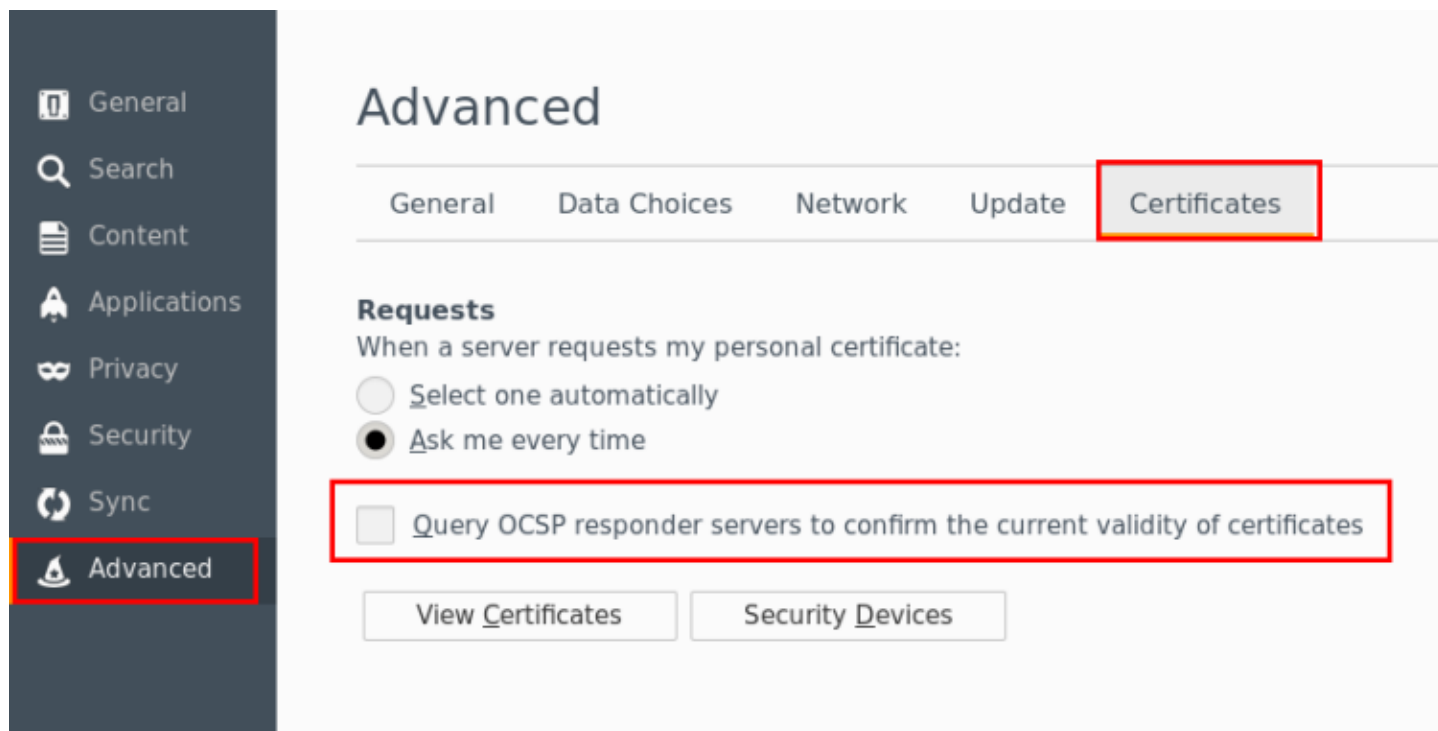
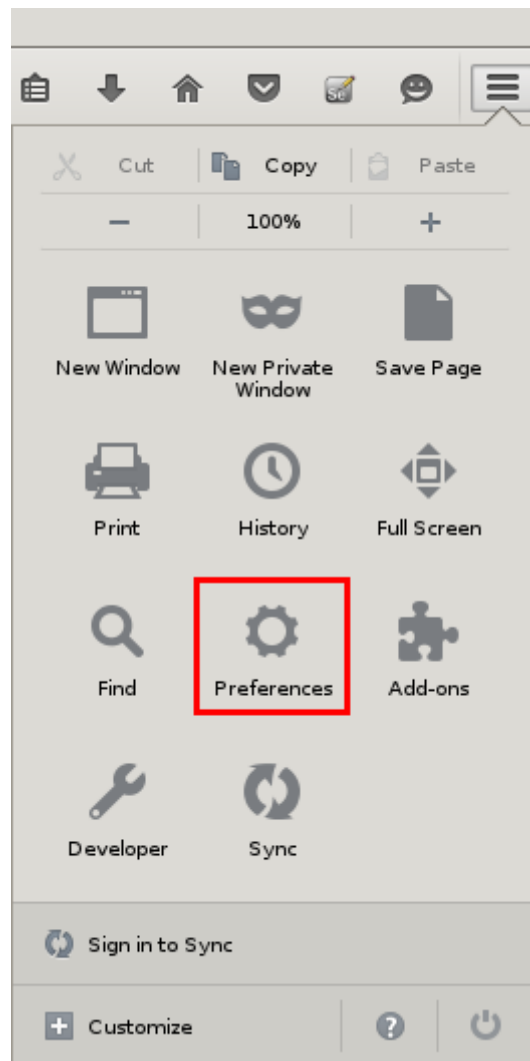
If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

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

Add Exception...



他の方法として、SSL 証明書を受け入れたい場合は、Firefox の “Advanced > Certificates” タブのオプションへアクセスし、“Query the OCSP respondent servers to confirm the current validity of the certificates” を選択します。

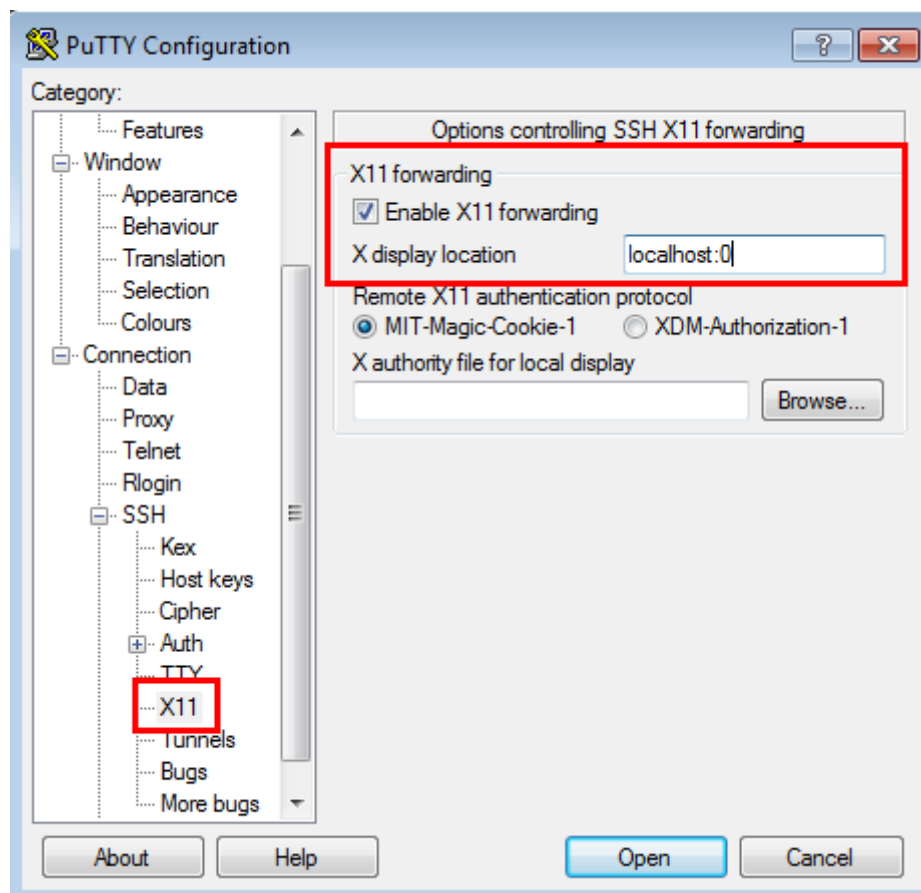


Windows デスクトップへの X11 リダイレクト

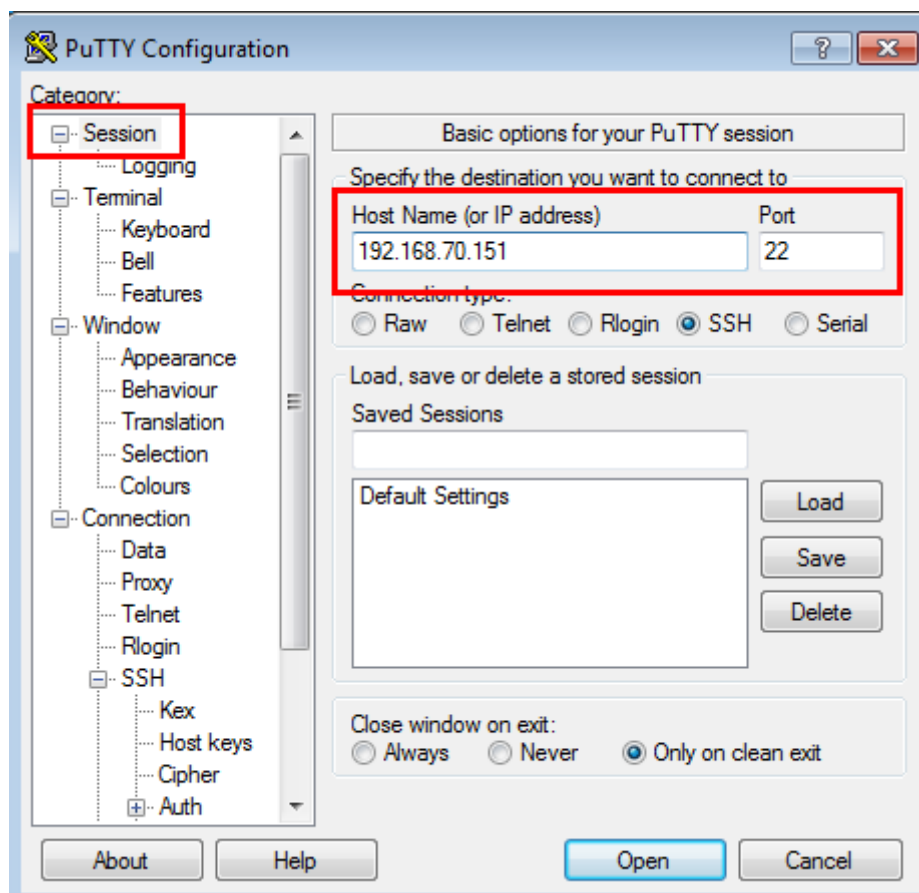
Windows の場合、リダイレクトを行えるようにするために、Xming などの X サーバをインストール

する必要があります。インストールして起動するとXのリダイレクトができるようになります。

SSH Putty クライアントを用いて、接続前に “Connection > SSH > X11” にアクセスします。そして “Enable X11 forwarding” がチェックされ “X display location” が “localhost:0” になっていることを確認します。



そして “Session” セクションに戻り、接続を確立します。

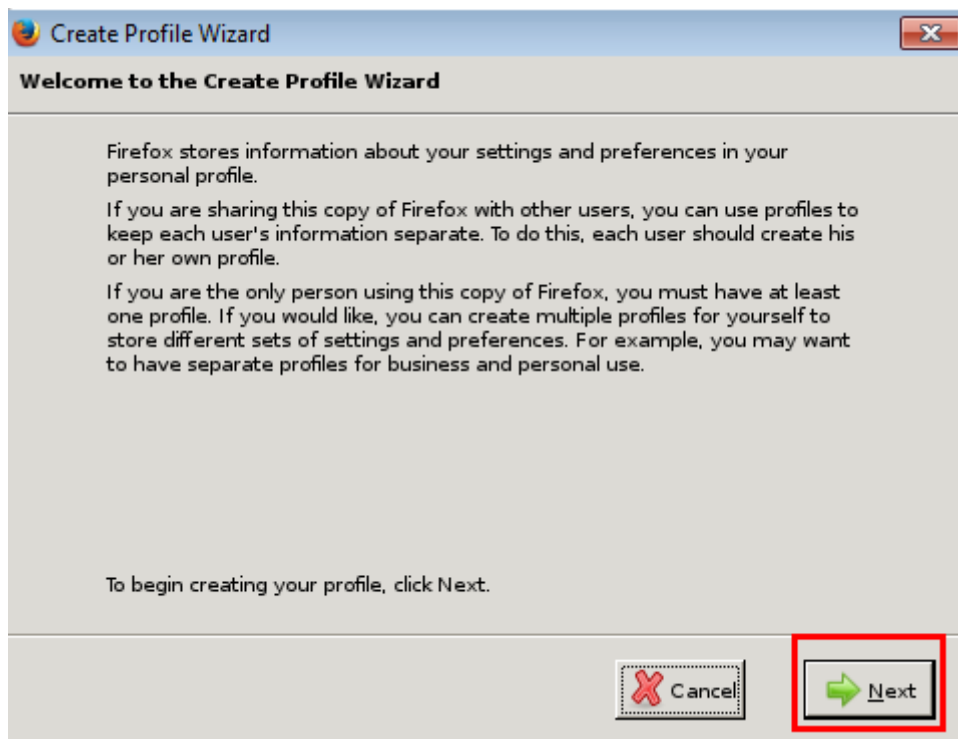


接続したら PWRD サーバ上で Firefox ブラウザを起動すると Windows デスクトップでその画面を見ることができます。

```
firefox -p
```

デフォルトのインストールでは、デフォルトのプロファイルのみ見ることができます。Windows へのインストールの場合と同様に、新しいプロファイルを作成して使用することをお勧めします。

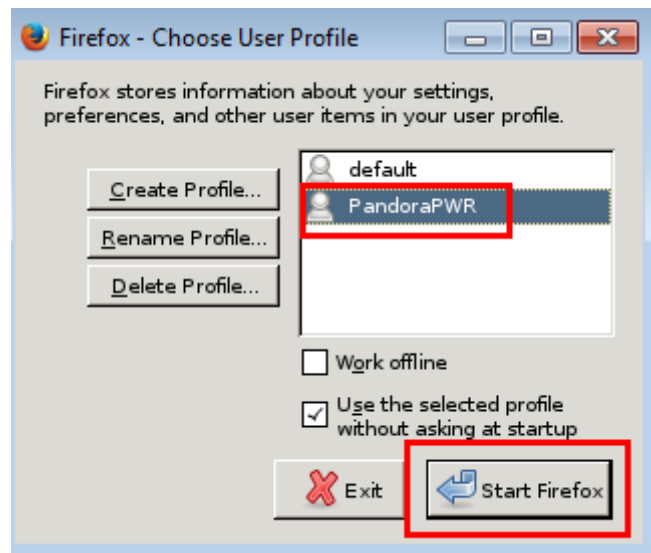




プロファイルが /opt/firefox_profile 以外にある場合、PWRD の起動スクリプト /etc/init.d/pwrld でプロファイルのパスを指定するように編集する必要があります。

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

プロファイルを作成したら、ブラウザを起動します。



起動したら、アップロードする証明書を使って URL にアクセスし、ブラウザの例外として追加します。



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!](#)

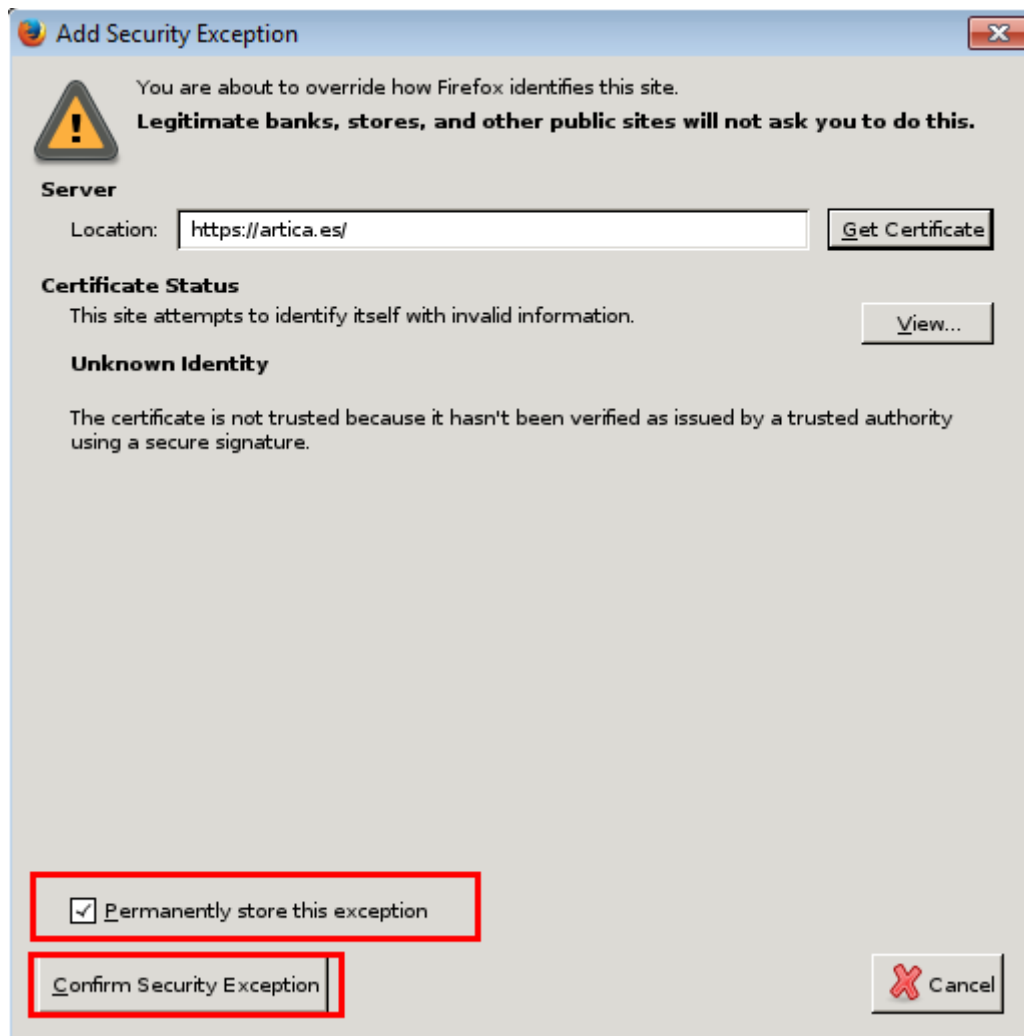
▶ Technical Details

▼ I Understand the Risks

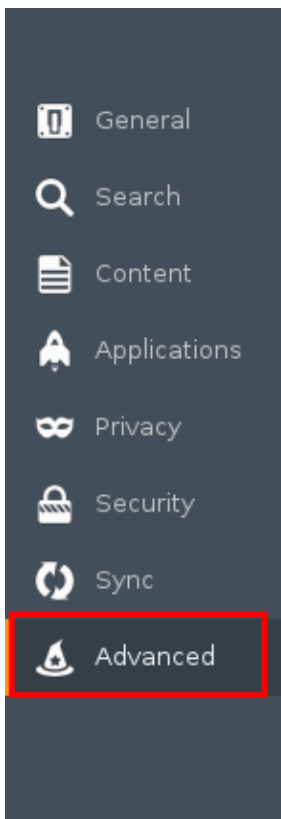
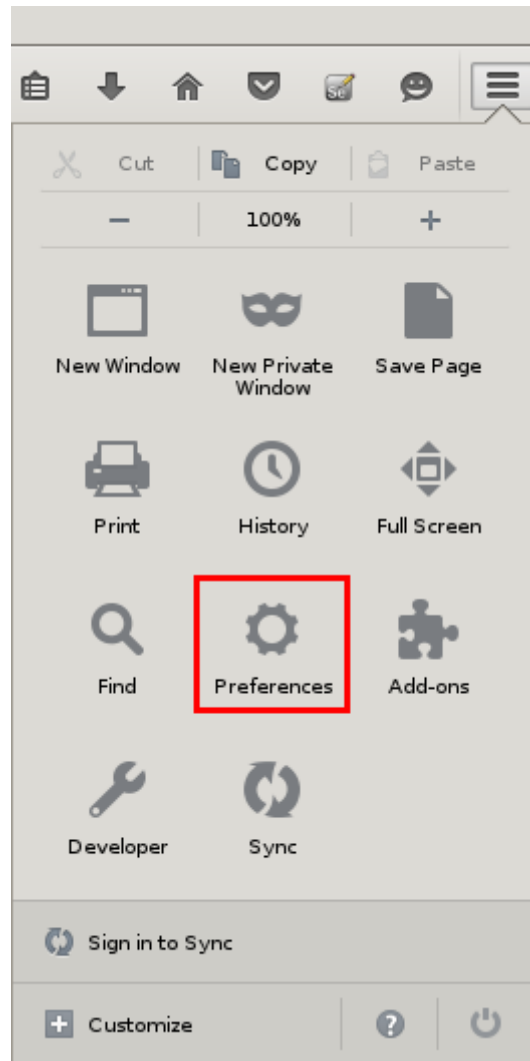
If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

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

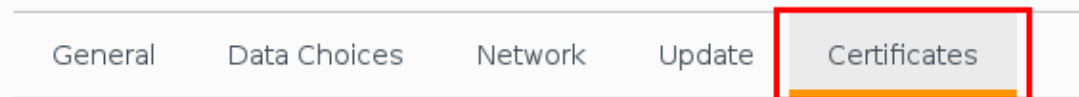
[Add Exception...](#)



他の方法として、SSL 証明書を受け入れたい場合は、Firefox の “Advanced > Certificates” タブのオプションへアクセスし、“Query the OCSP respondent servers to confirm the current validity of the certificates” を選択します。



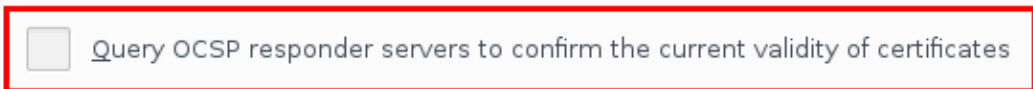
Advanced



Requests

When a server requests my personal certificate:

- Select one automatically
- Ask me every time



[View Certificates](#)

[Security DeVICES](#)

Pandora サーバ設定

選択したモードにかかわらず、一度起動するとPandoraサーバーの設定ファイルに WUXServer の設定パラメータを追加して、ブラウジングセッションの実行の割り当てを開始できます。

PWRD サーバを Pandora FMS サーバと同じサーバに配置するのであれば、次の設定を追加します。
(`/etc/pandora/pandora_server.conf` へ追加します)

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

wux_timeout パラメーターで最大トランザクション時間を 30秒に設定していますが、必要に応じて、この値を環境に合わせて調整します。

Pandora FMS スレッド管理

PWRD を hub モードで利用する場合:

- wuxserver スレッド管理は、pandora_server サービスを起動した際に自動的に行われます。
- これはSelenium hub にある MINOR ブラウザのノード数を考慮して行われます。例:
 - hub に 2 つの Firefox ノードと 2 つの Chrome ノードが設定されている場合、wuxserverスレッドの数は 2 になります。
 - hub に 1 つの Firefox ノードと 4 つの Chrome ノードが設定されている場合、スレッドの数は 1 になります。
 - hub に 6 つの Firefox ノードが設定されている場合、スレッドの数は 6 になります。

各スレッドは、wuxserver から Selenium hub に同時に送信できるセッションを示していることに注意してください。

PWR セッションの記録

Selenium 3 で実装された、新たなバージョンの Selenium IDE は、Pandora FMS バージョン 745 以降で対応しています。Selenium 3 におけるトランザクションの記録には影響しません。

User experience 監視の前に、選択した技術タイプに応じて、適切なレコーディングシステムを使用してください。

Selenium 2 でのセッションの記録は、Firefox でのみ実施可

能です。

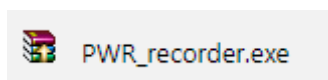
PWR で操作を記録するには、モジュールライブラリにある PWRD レコーダーが必要です。

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

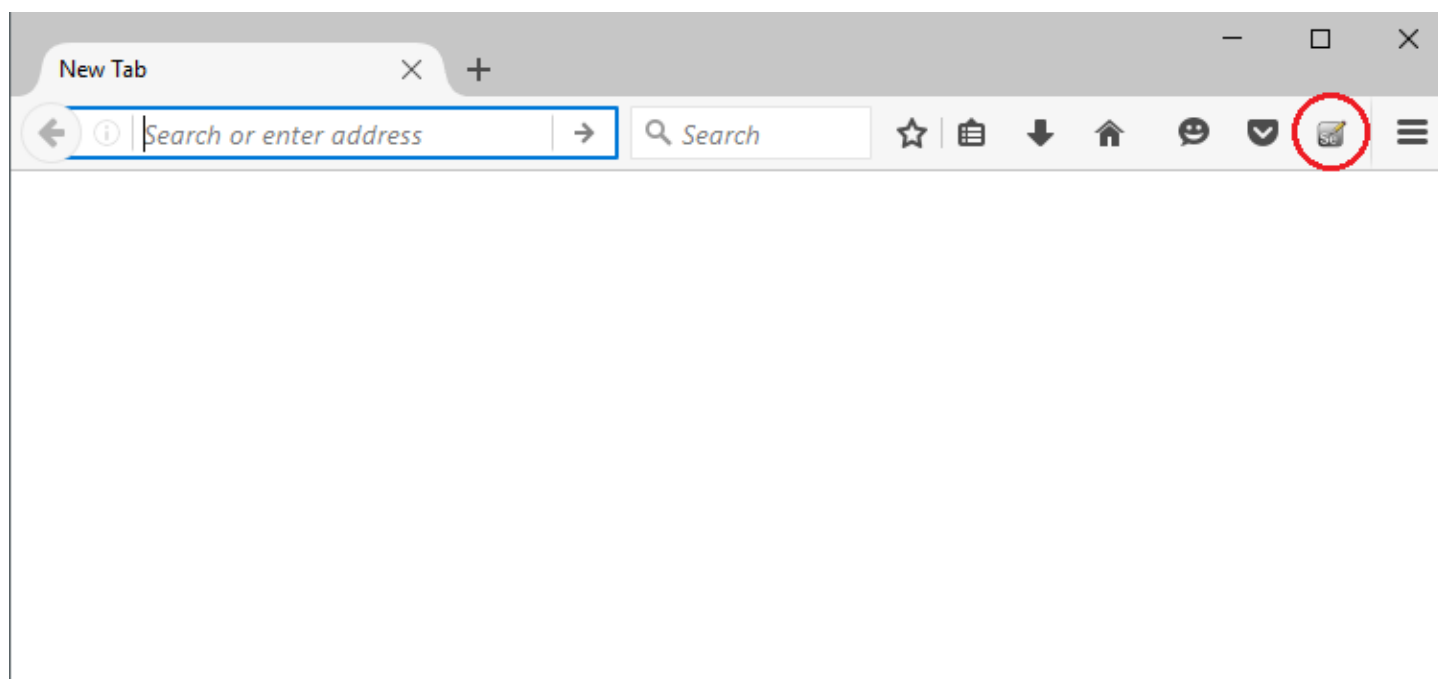
以下が含まれています:

- ウェブブラウザ **Firefox バージョン 47.0.1**.
- 拡張 **Selenium IDE**.

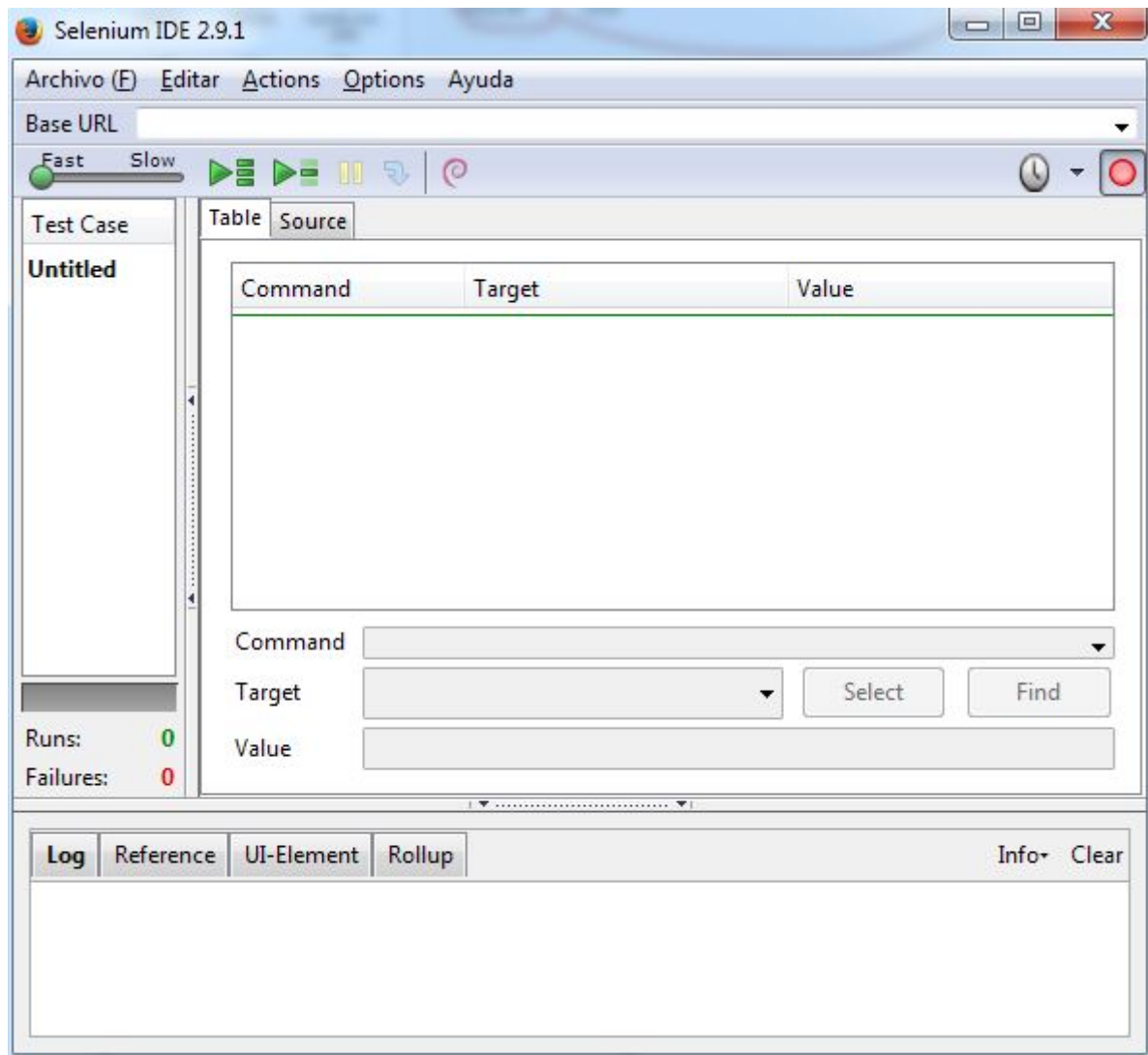
レコーダ PWR_recorder.exe を起動します。



PWR セッションの記録準備ができた状態で Firefox が自動的に起動します。

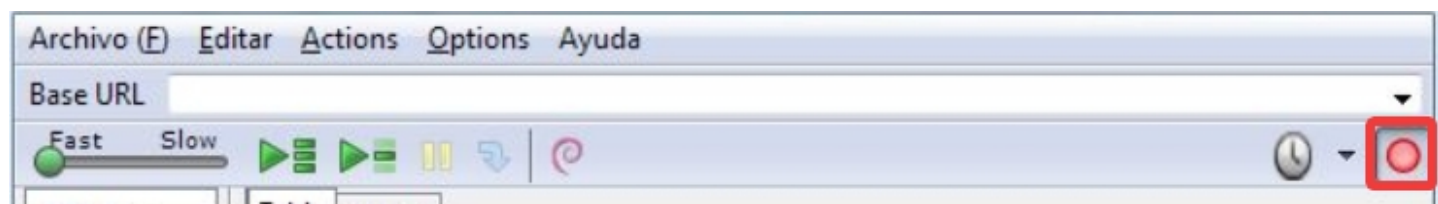


Selenium IDE にアクセスしたあと、あなたの操作のレコーディングを開始できます。

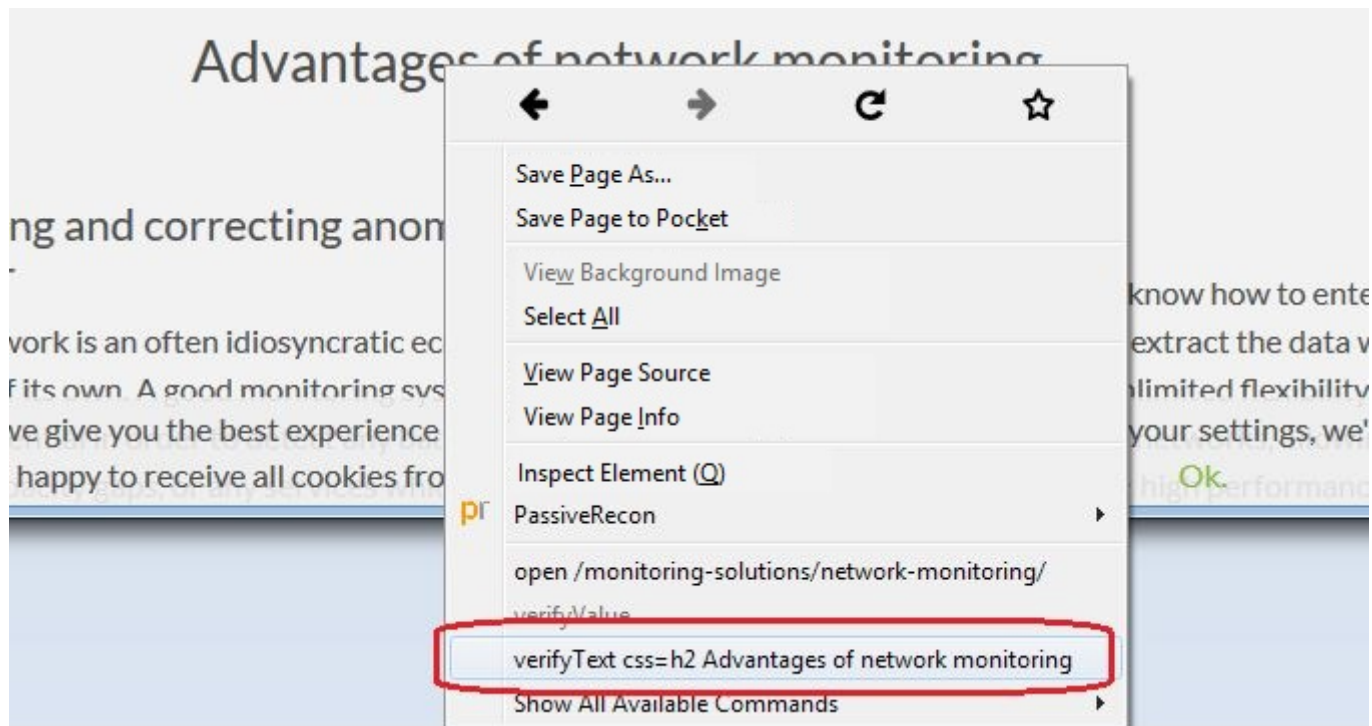


ここから、監視したいウェブサイトをブラウザし、それぞれのステップのアクションがレコーダーに現れます。

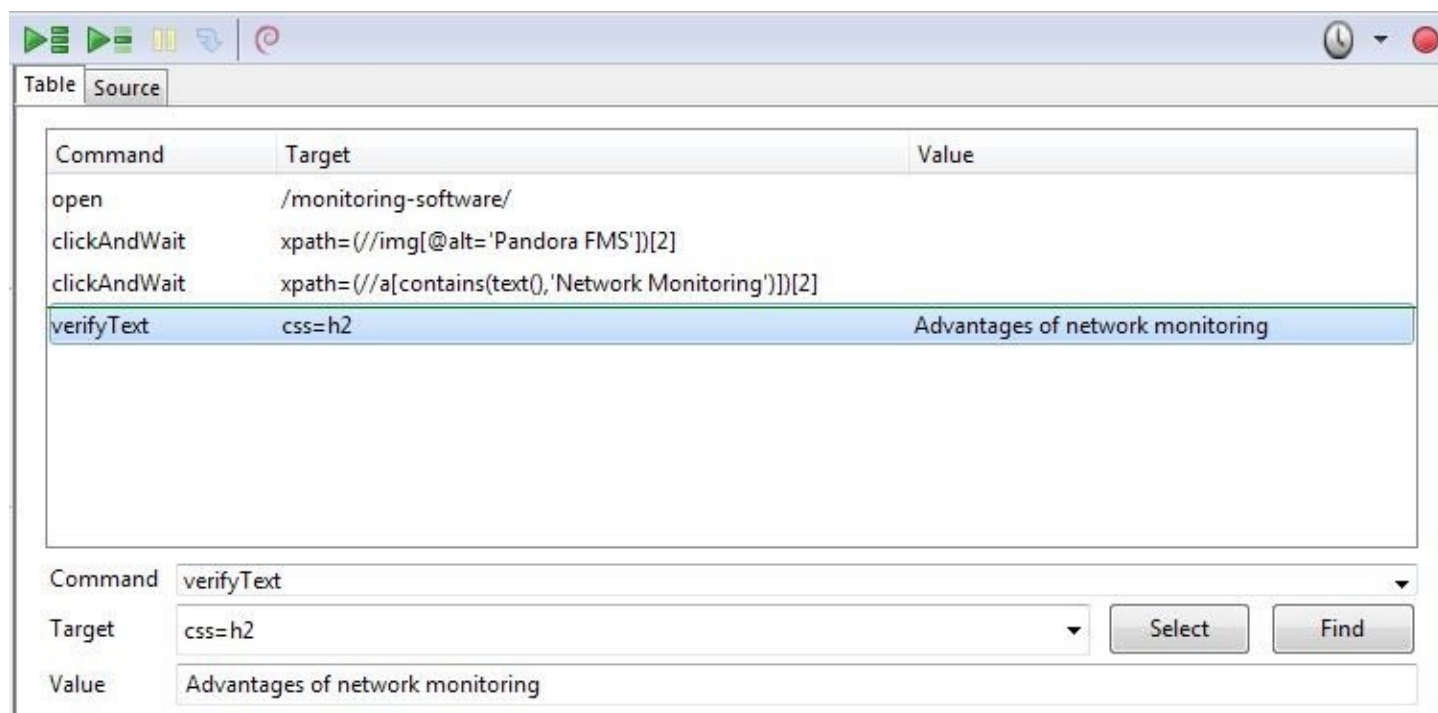
記録を停止するには、レコーダーの右上にある次のボタンを利用します。



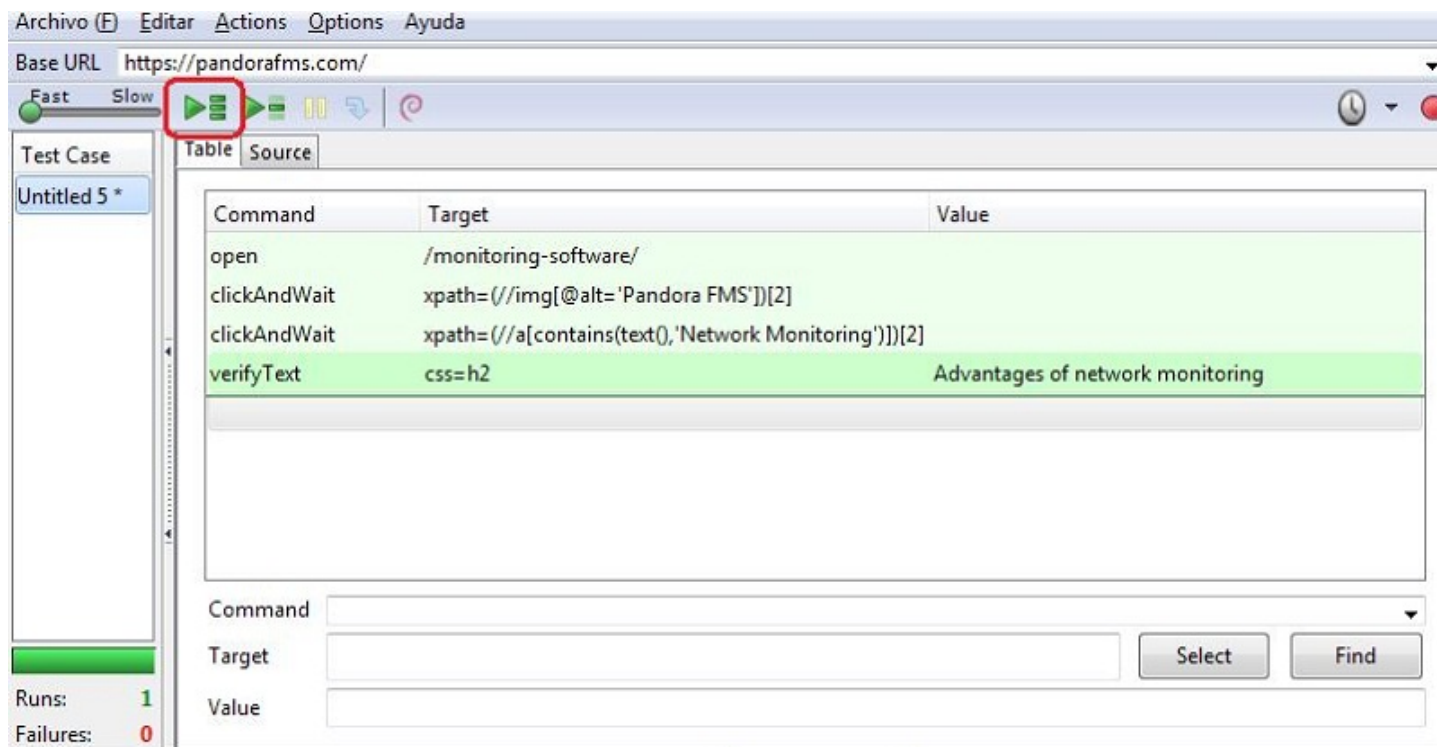
処理が完了したら、ページのチェックを実施できます。たとえば、特定のテキストの存在を確認して、読み込まれたページが正しいことを確認できます。これには、記録を行っているときに、ブラウザウィンドウのテキスト部分を右クリックし、オプション `verifyText` を選択します：



テキストチェックアクションを示す新しいステップがレコーダに表示されます。



Play Entire test suite ボタンで全体の流れを再現し、一通り成功するか確認できます。



The screenshot shows the Pandora FMS WUX interface. The top menu includes 'Archivo (F)', 'Editar', 'Actions', 'Options', and 'Ayuda'. The 'Base URL' is set to 'https://pandorafms.com/'. The interface has a 'Fast' and 'Slow' speed control, and a 'Test Case' sidebar on the left with 'Untitled 5 *'. The main area is a table with columns 'Command', 'Target', and 'Value'. The table contains the following rows:

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

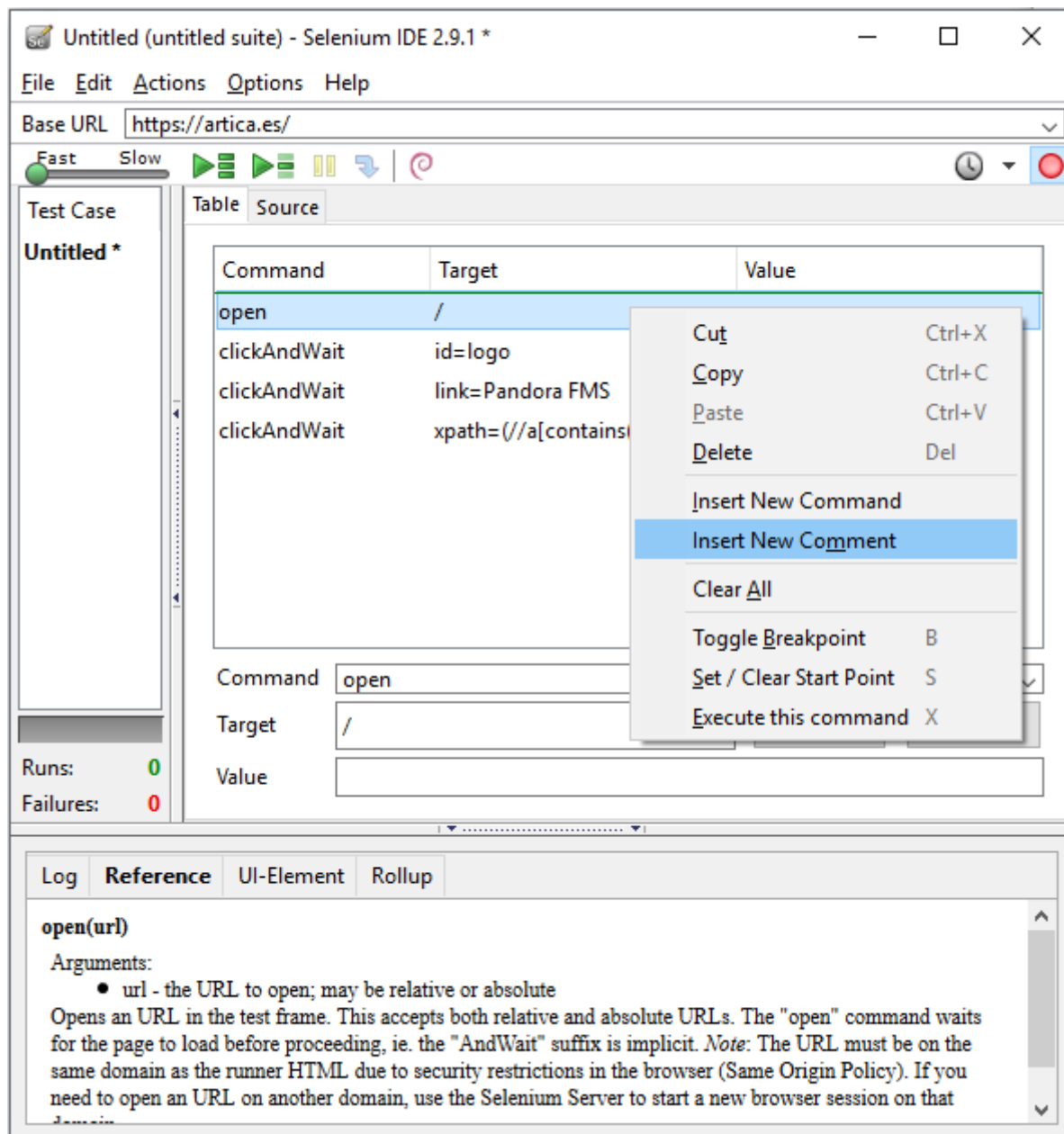
Below the table, there are input fields for 'Command', 'Target', and 'Value', along with 'Select' and 'Find' buttons. At the bottom left, it shows 'Runs: 1' and 'Failures: 0'.

一連の動作を確認したら、それを保存します(File → Save Test Case) これが Pandora コンソールで新たな Web 分析モジュールを作成するときロードするファイルになります。

Pandora FMS WUX のトランザクションセッションを記録する

Pandora WUX は、ウェブサイトのブラウジング監視を、それぞれのステップを表す複数のモジュールに分割することができます。

新たなコントロールポイントを追加しフェーズモジュールを生成するには、フェーズを開始したい場所で右クリックします。



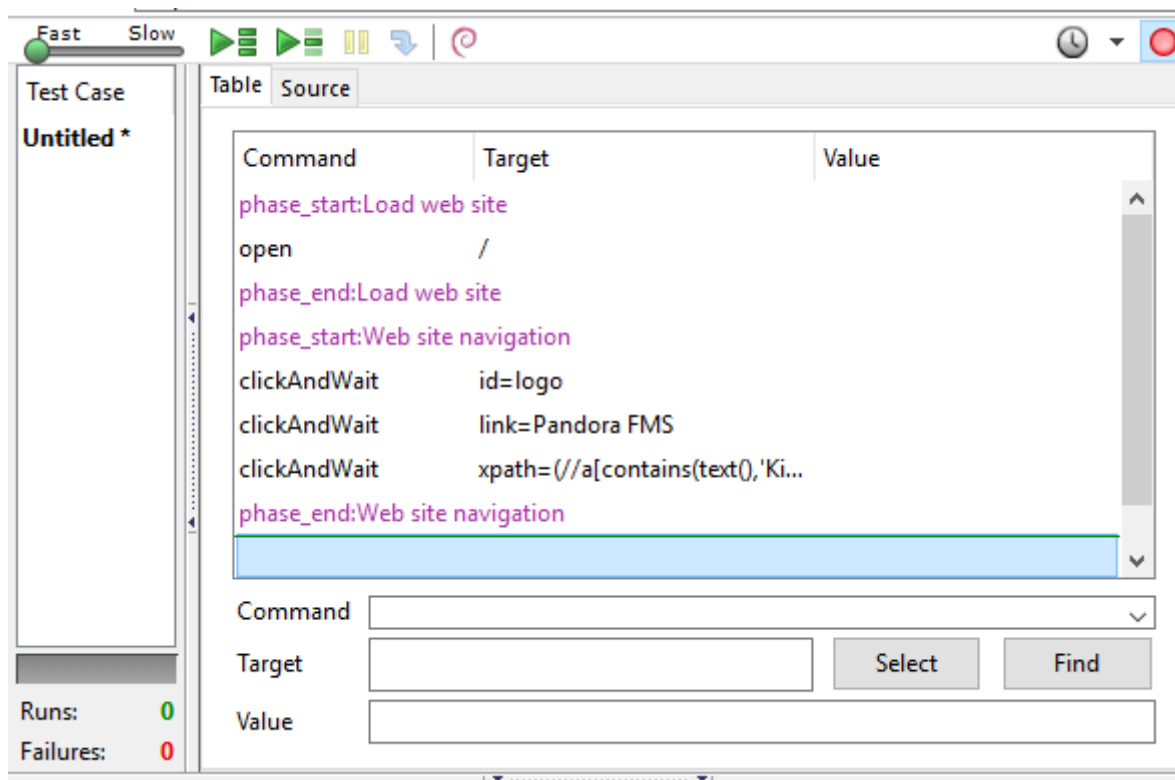
次のテキストをコメントとして書きます。

```
phase_start:<phase_name>
```

フェーズは、次のコメントまでに存在するすべてのコマンドと時間の結果を含みます。

```
phase_end:<phase_name>
```

phase_start と phase_end の間に実行されるすべてのコマンドが、一つのフェーズとして扱われます。



Selenium 2 で作成された記録は正しく機能しない場合があることを考慮してください。

Web トランザクションの実行

標準実行

あらかじめ記録した PWR セッションを起動するには、`pandora_ux_x64.exe` を実行します。これは、<https://pandorafms.com/library/user-experience-pandora-ux> からダウンロードできます。

PWR 動作モードとセッションガイドラインを含んだファイルを選択します。エラーのスクリーンショットが、`-folder` パラメータで指定された場所に保存されます。実行に失敗したときに再実行するための `-retries` パラメータを利用することもできます。Windows での実行は次のようにします。

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

次のモジュールが返ります。

- `UX_Status_project_name`: シーケンスが成功したかどうか
- `UX_Time_project_name`: シーケンスを完了するのにかかった時間
- `UX_Snapshot_project_name`: エラーが発生した直前のスクリーンショット

正常実行結果例:

```
<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>

```

エラー実行結果例:

```

<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,
iVBORw0KGgoAAAANSUhEUgAA...JRU5ErkJggg==]]></data>
  <description><![CDATA[Image (last error)]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>

```

すべて問題なければ、監視を実行するために実行コマンドラインを(マシンにインストール済の) Pandora エージェントに追加できます。例えば以下の通りです。

Windows:


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

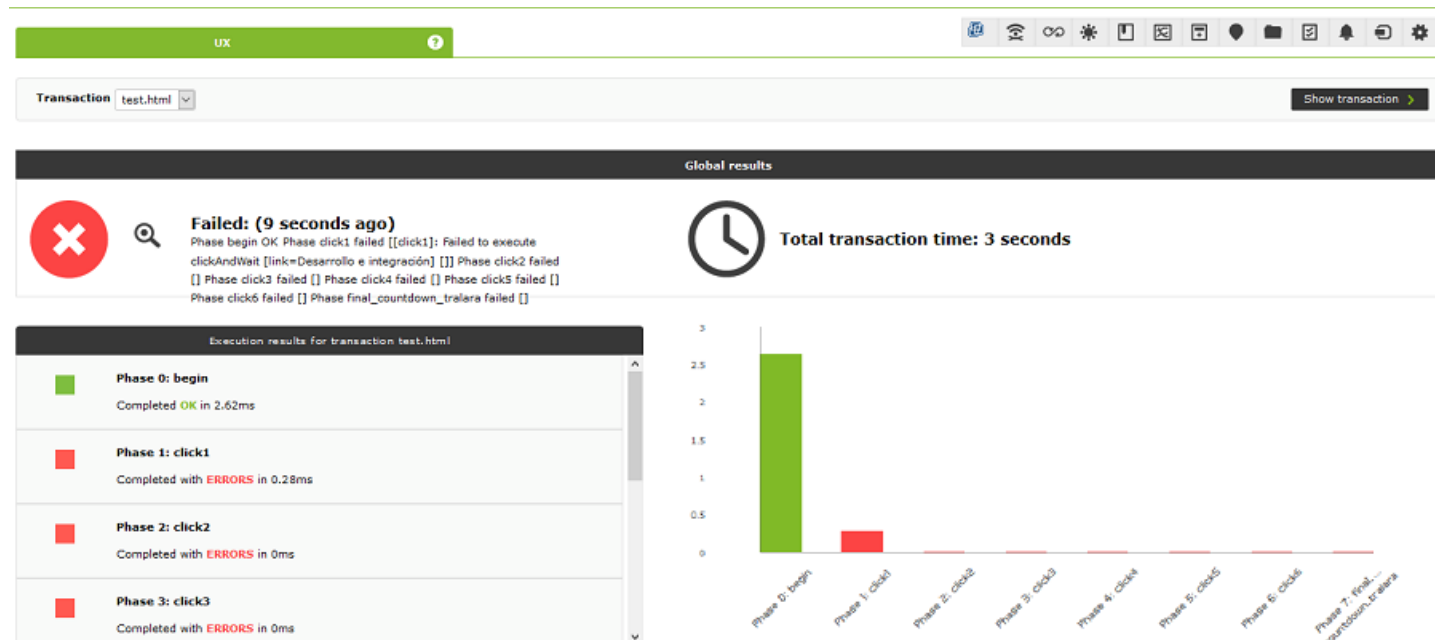
Linux:

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

エージェントが Pandora FMS サーバへデータを送ると、関連するモジュールが作られます。モジュールの関係を見るには、**階層モード**を有効にします。

	UX_Status_sesion.html	Test OK		N/A - N/A	1		101	45 seconds
	UX_Snapshot_sesion.html	Image (last error)		N/A - N/A	No image ava		101	20 hours
	UX_Time_sesion.html	Test OK		N/A - N/A	4.7		101	45 seconds

前述の通り、実行結果(成功 失敗)は、*UX_Status_session.html* モジュールで、所要時間(秒)は、*UX_Time_session.html* モジュールで、失敗時のスクリーンショット(例では空です)は *UX_Snapshot_session.html* に保存されます。アラートを設定したり個々のグラフを表示したりするには、これらのモジュールを使います。



ステージごとの実行

前の章で説明した Pandora UX PWR トランザクションの記録がある場合、各フェーズを特定するのに必要なモジュールはシステムが生成します。これは実行は一つ前と関連性がないことを意味します。対応する htmlファイル(この場合、異なるフェーズを含む)を指定します。

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

次のモジュールが返されます。

- UX_Time_project_name.phase_order
- UX_Status_project_name.phase_order

フェーズでエラーが返る場合は、次のモジュールが作成されます。

- UX_Snapshot_project_name.phase_order

エラー時のスクリーンショットが表示されます。

全体の状態モジュールが、次のラベルで返されます。

- UX_Global_Time_project_name
- UX_Global_Status_project_name
- UX_Global_Snapshot_project_name

エラー時のスクリーンショットが表示されます。





























































エージェントの実行コマンド設定は前述の例と同じですが、フェーズを含む html ファイルがありません。

Pandora FMS サーバにデータが届くと、モジュールとして表示されます。情報を見やすくするには、モジュールの *階層モード* を有効にします。

 	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

トランザクションおよびフェーズの詳細を確認するには、エージェントの *WUX* にアクセスします。

以下の例は、当社のウェブサイトの閲覧処理と複数のアイテムの購入を表しており、5段階に分けて、時間を正確に測定して改善が必要な場所、またはボトルネックが発生している場所がわかるようにしています。

  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

データ取得

Pandora FMS UX は、ウェブのブラウジング監視中にウェブコンテンツのデータを取得できます。

この機能を利用するには、Selenium IDE でテストケースにコメントとしていくつかのディレクティブを追加する必要があります。

The screenshot shows the Selenium IDE 2.9.1 interface. The window title is 'ejemplito (untitled suite) - Selenium IDE 2.9.1'. The Base URL is 'http://fringe.lab.artica.lan/'. The test case 'ejemplito' is selected in the left sidebar. The main area displays a table of commands and their targets and values.

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

Below the table, the Command field contains 'getValue;resultado en texto;generic_data_string;<h1>(.*?)</h1>'. The Target and Value fields are empty. There are 'Select' and 'Find' buttons next to the Target field.

The Log tab is active, showing the following entry:

```

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.

```

この例では、監視の実行後に値を取得しています。そして、モジュール(getValue ディレクティブ)として Pandora に追加されます。

値を取得するプラグインを作成するステップは次の通りです。

- Selenium IDE で、テストケースにコメントを追加します。
- ディレクティブをコメントで記載します。

データ取得ディレクティブ:

```
getValue;<モジュール名>;<モジュールデータタイプ>;<データ取得のための perl 正規表現>
```

各フィールドはセミコロンで区切ります。

- getValue: UX システムのディレクティブです。
- モジュール名: Pandora に表示されるモジュール名です。
- モジュールデータタイプ: Pandora が取り扱える **データタイプ**です。
- Perl 正規表現: perl の正規表現です。カッコ内のデータが取得されます。

以下に例を示します。

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

温度を取得したい場合、正規表現を以下のようにします。

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

すると、例えばシステムは数値 “ 54 ”、または小数点以下の任意の数を取得します。

ディレクティブ全体は次のようになります。

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

モジュールを生成するコマンドは次の通りです。

- getValue: 値を展開します。

```
getValue;モジュール名;モジュールタイプ;文字列マッチ正規表現
```

- getVariable: (バージョン 753 以上) 変数から特定の値を展開します。

```
getVariable;モジュール名;モジュールタイプ;変数名
```

- getScreenshot: 画面をキャプチャします。

```
getScreenshot;モジュール名
```

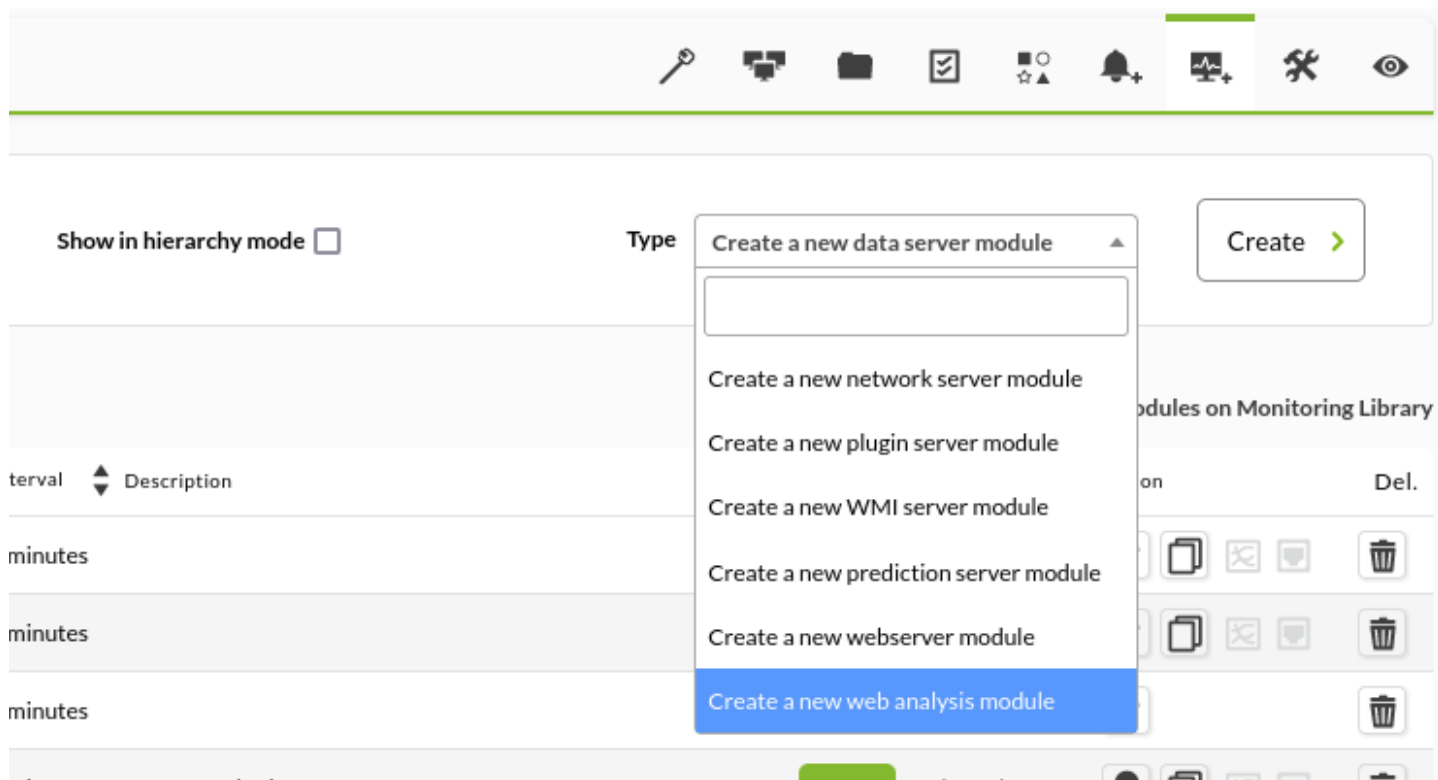
getValue は、[モジュールを生成するコマンド](#) に属するコマンドです。詳細はリンクを確認してください。

データ管理と表示

Pandora FMS コンソールでの Web 分析モジュールの作成

ブラウジングセッションを記録したら Pandora FMS コンソールのモジュールとして展開します。

そのためには、WUX 機能が有効化されたサーバで任意のエージェントにアクセスし、新規ウェブ分析モジュールの作成(Create a new web analysis module) オプションでモジュールを作成します。


















Show in hierarchy mode

Type

- 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

Create >

Modules on Monitoring Library

on	Del.
   	
   	
   	

作成(Create) を選択後、次のフォームに入力します。

✓ 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 : 0
 Change each status : To `normal` 0 To `warning` 0 To `critical` 0

Historical data

Paste your selenium test, exported as HTML, here

No file selected. Upload your selenium test in html or side (only Selenium 3) format

モジュール名(Module name)

モジュールWUX(親)、監視結果を持つすべてのサブモジュールはこの新しいモジュールに依存します。

パフォーマンステスト実行(Run performance tests)

閲覧可否の監視だけでなく、対象のウェブサイトへのアクセスのパフォーマンス統計も監視することを示します。

テスト実行サーバ(Execute test from)

テストを実行する WUX サーバ。

ブラウザ(Browser)

テストを実行するウェブブラウザ。

ブラウザは、WUX サーバが使用する Selenium グリッドまたはサーバ上にある必要があります。

ユーザデータディレクトリ(User data dir)

(バージョン 764 以降) オプション: データディレクトリを指定できます。

ユーザデータディレクトリ(User data dir) オプションは、Google Chrome でのみ動作します。

プロファイル(Profile)

(バージョン 764 以降) オプション: ユーザプロファイルを指定できます。プロファイル名が存在しない場合は、デフォルトのプロファイルを使用します。

プロファイル(Profile) オプションは、Google Chrome でのみ動作します。

安全でない証明書を受け入れる(Accept insecure certificates)

有効にすると、ナビゲーションの際に安全でない証明書(自己署名、期限切れなど)が受け入れられます。

このオプションは、Google Chrome および Mozilla Firefox ブラウザでのみ使用でき、テストが Selenium 3 サーバによって実行される場合にのみ有効になります。

データ保存(Data history)

実行結果のモジュールデータを保存するかどうか。


テキスト入力領域(Text area section)


このテキストボックスには、あらかじめ録画したブラウジングセッションのコンテンツファイルをコピー(またはアップロード)します。


✓ **Advanced options**

Description

Custom ID

Interval 

FF interval 

Retries 

連続障害検知抑制間隔(FF interval)


モジュールの連続障害検知抑制間隔(秒単位)。

リトライ(Retries)

モジュールのリトライ回数。

カスタムマクロ

✓ **Custom macros**

Custom macros 

Name	Value
<input type="text"/>	<input type="text"/>

カスタムマクロは、ブラウジングセッションのファイル内に設定した特定の文字列を、変化する特定の値に置き換える動作をします。

ウェブ解析モジュールの場合、この機能が改善され、値を日時の変数に変換できる 動的マクロ が利用可能になりました。

なぜこの機能が必要なのでしょうか？

会議室を予約するウェブアプリケーションの機能を、ブラウジングセッションを通して監視する必要があります。

固定の日時を設定すると、システムはおそらく予約を取り消すことになります。これは、ある時点で過去に予約されている可能性があるためです。たとえば、先週の予約と同じ予約をしようとしている可能性があります。

また、予約の実行が可能な期間を見つけることも必要です。例えば、システムは当月を超えないなど一定の期間内に部屋を予約する必要があります。

数日おきに設定を気にしたりマクロを編集するのを避け、動的マクロを使用することにより、テストを行う翌日の部屋を常に予約するようにシステムを設定することができます。

この機能を利用するには、次のような特定の形式を用います。

- @DATE_FORMAT ユーザ定義フォーマットでの現在の日付/時間
- @DATE_FORMAT_nh 時間
- @DATE_FORMAT_nm 分
- @DATE_FORMAT_nd 日
- @DATE_FORMAT_ns 秒
- @DATE_FORMAT_nM 月
- @DATE_FORMAT_nY 年

“n” は、符号なしの数値(正の数)または負の数です。

書式は、[perl の strftime](#) に従います。





































例:

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

データ表示

WUX で生成された情報は、次のようにモジュールに表示されます。モジュール一覧の中で階層表示を有効化することにより、情報をより分かりやすく表示します。

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	 

階層表示では、次のモジュールを見ることができます。

- *module_Global_Status*: ブラウズ完了の 全体の 状態を表示します。
 - 記録がある場合は、WUX 記録の状態を示します。
 - **パフォーマンステストの実行**が有効になっており、記録が含まれていない場合、グローバルステータスモジュールの状態は、これらのモジュールの実行結果の状態になります。
- *module_Global_Time*: ブラウズ完了の 全体の 所要時間を表示します。
- *module_Global_Screenshot*: ブラウズでエラーとなった画面キャプチャで、エラーの場合のみ生成されます。
- *module_Phase X*: *フェーズ名_Status*: フェーズ X におけるブラウジングの状態を示します。
- *module_Phase X*: *フェーズ名_Time*: フェーズ X における所要時間を示します。

例: エラー画面スクリーンショット

The screenshot displays the Pandora FMS interface. On the left, there's a sidebar with navigation options like 'Full list of monitors' and 'List of modules'. The main area shows a 'Current data at 2017-09-08 01:45:59' snapshot of a website with a contact form. Below this, a table lists monitoring data with columns for Description, Status, Warn, Data, and Graph.

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%

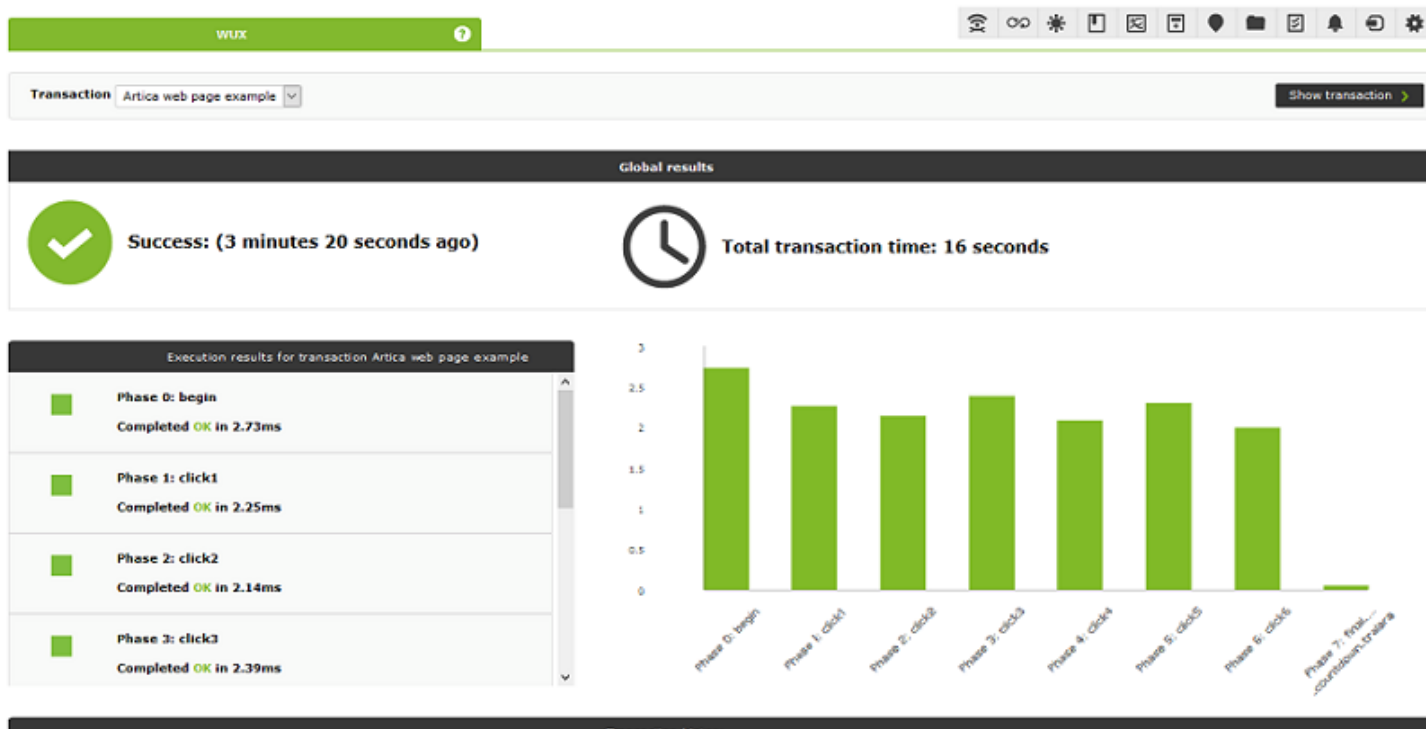
注意: Pandora FMS 7.0NG 712 以前のバージョンからアップグレードした場合は、少し変更する必要があります。

WUXServer によって生成された画面キャプチャを保存するには、データベースにて次の操作を行う必要があります。

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

この操作を行わないと、全てのスクリーンショット見ることができない可能性があります

エージェントの WUX セクションにアクセスすることによって、トランザクションとフェーズの追加の詳細を見ることができます。



ウェブサイトの統計情報は、以下の概念でまとめられています。

- Stats_TT: ウェブサイトの取得にかかるトータルの時間。
- Stats_TDNS: 対象の IP アドレスを名前解決するためにかかるトータルの時間。
- Stats_TTCP: TCP の接続にかかった時間。
- Stats_TSSL: SSL のネゴシエーションにかかった時間。
- Stats_TST: データ転送が開始されるまでにかかった時間。
- Stats_TTC: データ転送にかかった時間。すべての処理にかかった時間の合計です。
- Stats_TTR: ページのコンテンツの転送にかかった時間。
- Stats_TTR_main: HTML コードを転送するのににかかった時間。
- Stats_TTR_image: 画像(png|jpg|jpeg|bmp|tiff|gif|webp|svg)の転送にかかった時間。
- Stats_TTR_css: スタイルシートを転送するのににかかった時間。
- Stats_TTR_js: JavaScript ファイルを転送するのににかかった時間。

ウェブ解析モジュールへのアラートの割り当て

ウェブ解析モジュールに関するアラートは、Pandora FMS の全体のアラートシステムと同様です。

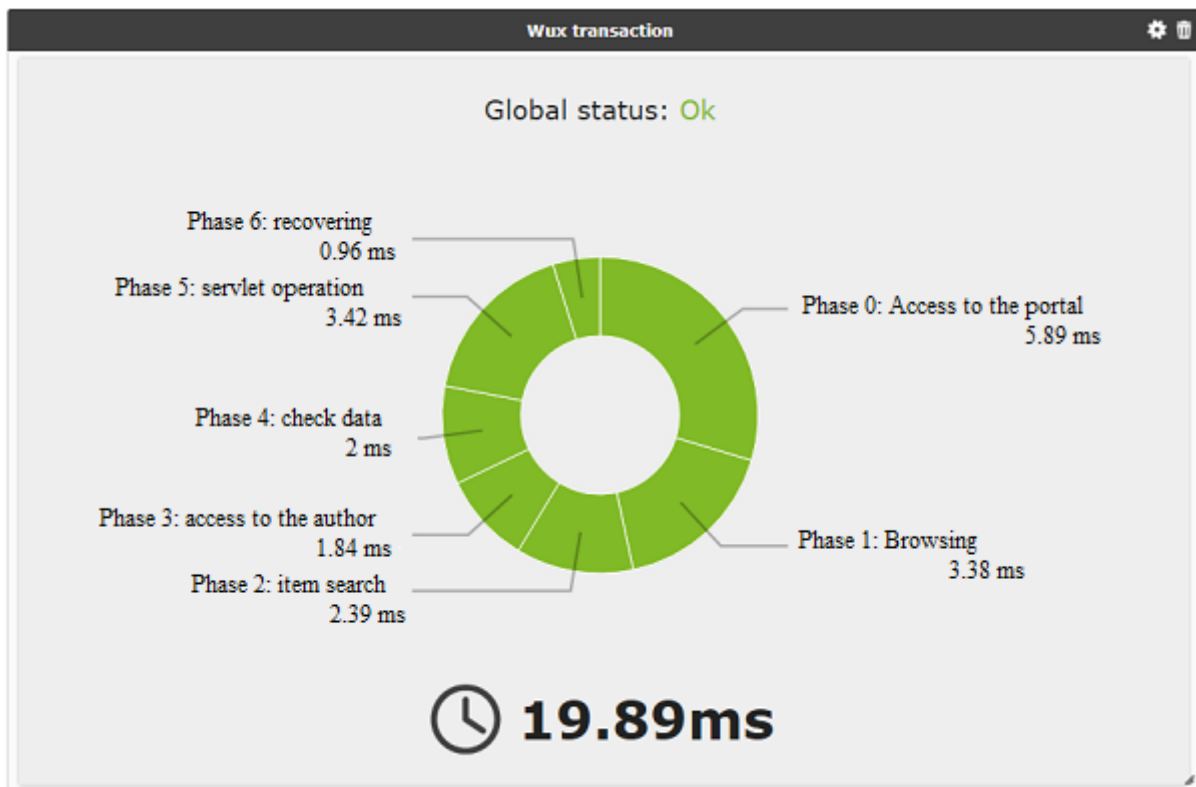
互換性を保つため、次のようにウェブ分析モジュールによって生成されたサブ要素にアラートテンプレートを割り当てることをお勧めします。

- 全体のブラウジングの状態
- 時間閾値に関するアラート
- 結果のスクリーンショットモジュールで、警告テンプレートを “ 常時 ” に設定したアラート

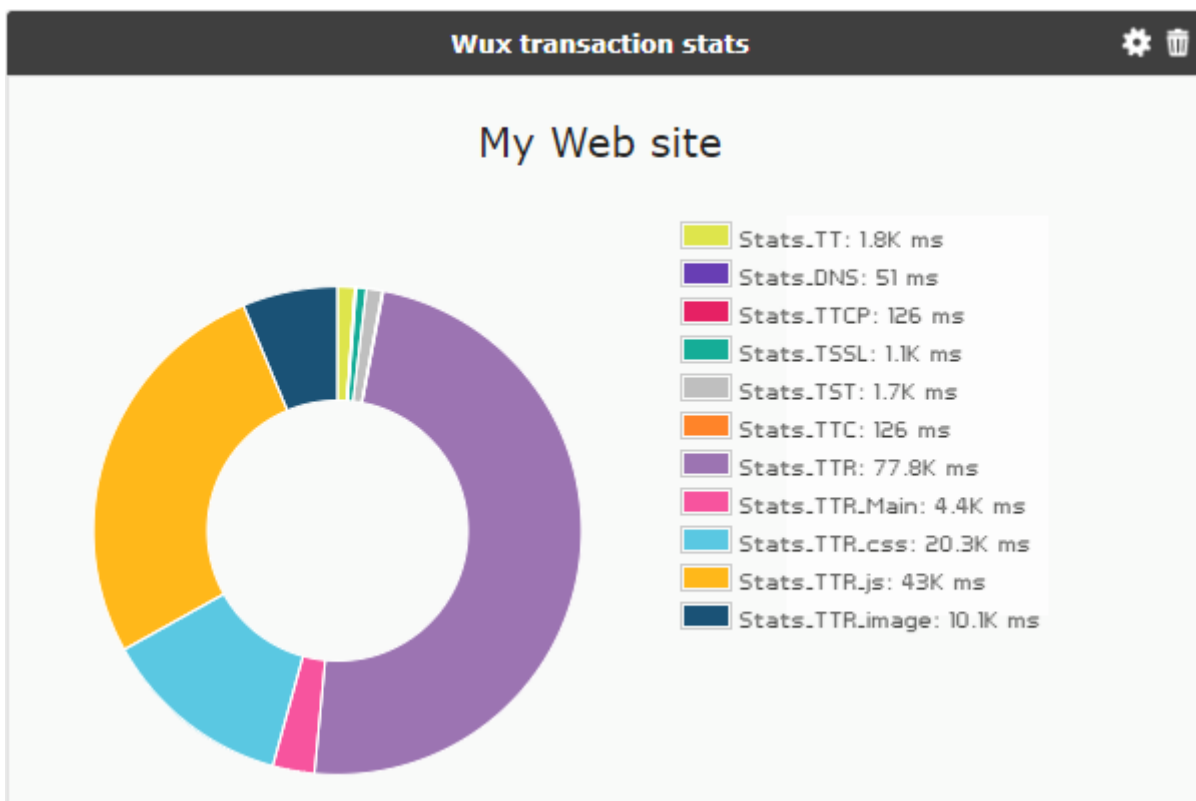
ダッシュボードの新たなウィジェット

ブラウジング情報を表示するための新たな 2つのウィジェットが追加されています。

ブラウジングのフェーズの状態ウィジェット。



ウェブサイトの統計ウィジェット。



デスクトップ UX (PDR)

PDR の展開

PDR は Windows でのみ動作するように設計されています。Pandora エージェントで実行される時プロセスモードで実行する必要があります。サービスモードでは動作しません。デスクトップセッションのブロックはできませんので、仮想マシンの利用をお勧めします。

事前準備:

- デスクトップモード、自動起動、自動ログインでの Windows OS

設定には以下を実行します:

Windows10® より前のバージョンの場合:

```
control userpasswords2
```

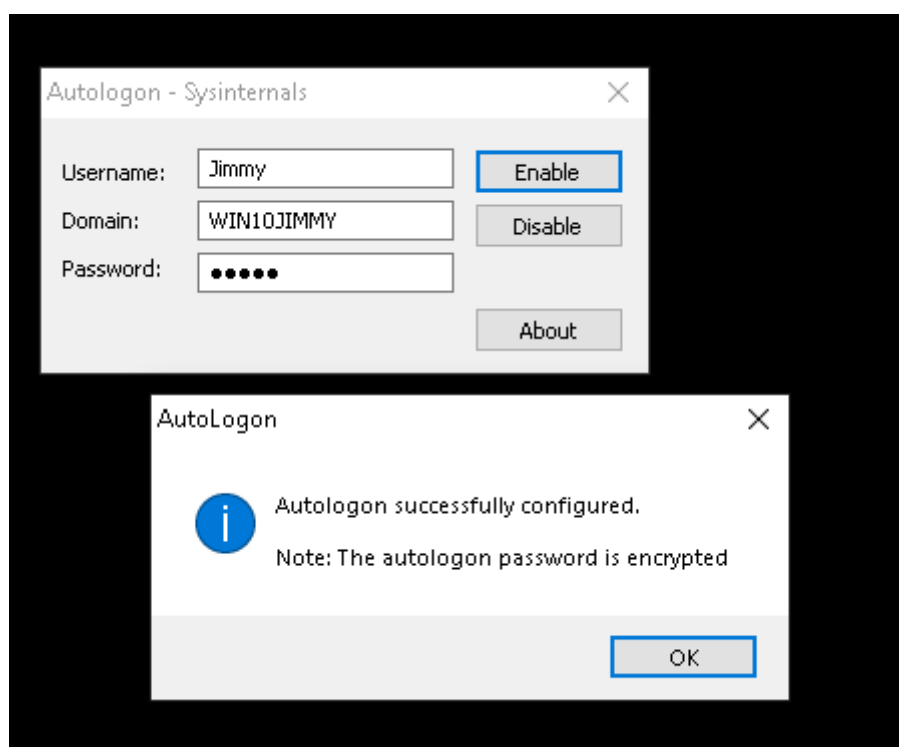
“このコンピュータを使うにはユーザ名とパスワードが必要(Users must enter a user name and password to use this computer)”のチェックを外します。



Windows 10® の場合:

MS Windows® バージョン 10で自動起動を有効にする 1つの方法としては、Microsoft® ドキュメン

トの <https://docs.microsoft.com/en-us/sysinternals/downloads/autologon> [ダウンロードセクション] で入手可能な Sysinternals Powertoy を使用することです。そして、監視を実行するユーザを有効にします。次に例を示します。



環境設定するには、以下のディレクトリを作成します。

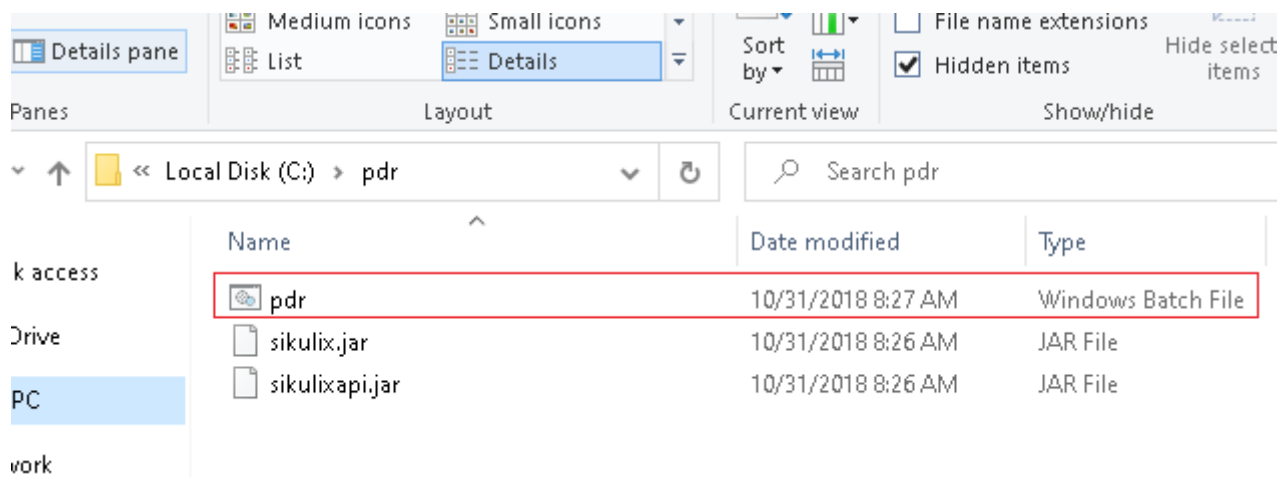
```
C:\PDR
```

<https://pandorafms.com/library/pdr-cmd-for-ux-monitoring/> から入手した PDR.rar ファイルを C:\PDR へ展開します。

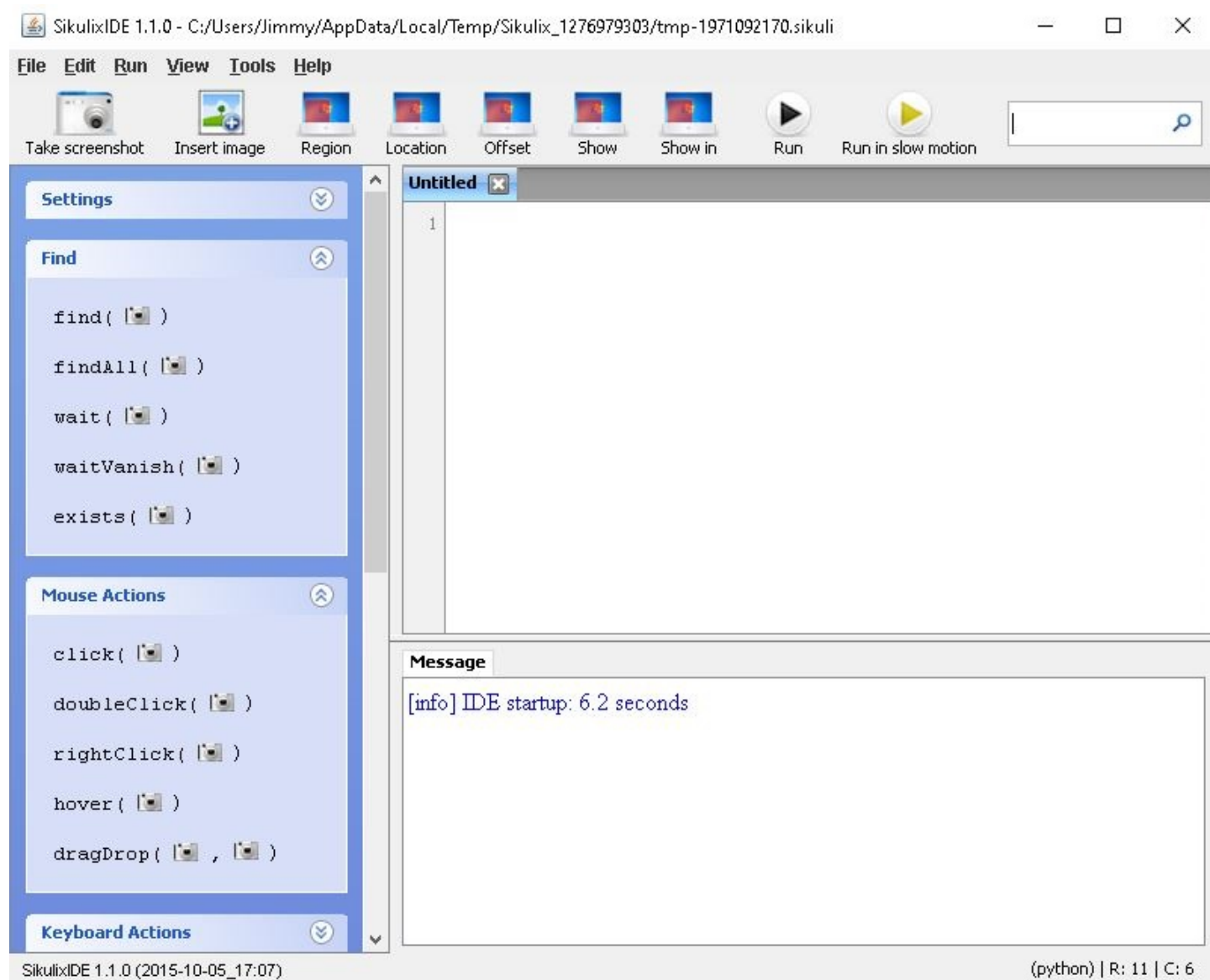
PDR セッションの記録

レコーダーを起動するには、以下のスクリプトを実行します。

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

起動したら、レコーディング環境に入ります。



実行するアクションを選択し、それらを適用する領域でスクリーンショットを実行します。以下に、最も一般的な例を示します。

一般的なアクション:



フロー制御アクション:



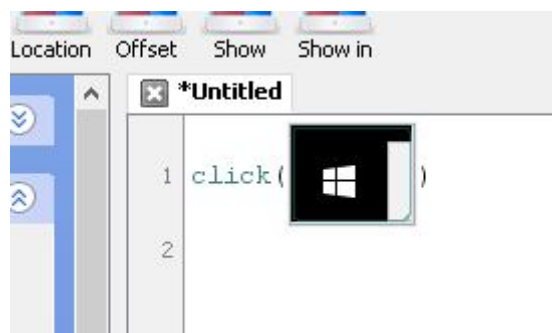
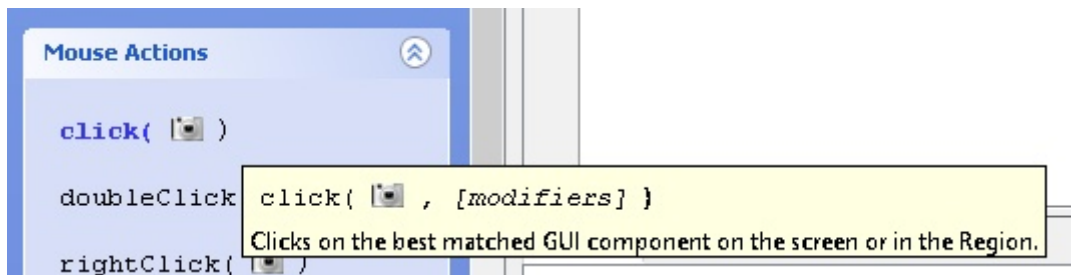
このスクリプト言語の詳細は、<http://sikulix-2014.readthedocs.io/en/latest/index.html> を参照してください。

プロジェクトが記録されると、次の要素を持つファイルが作成されます。

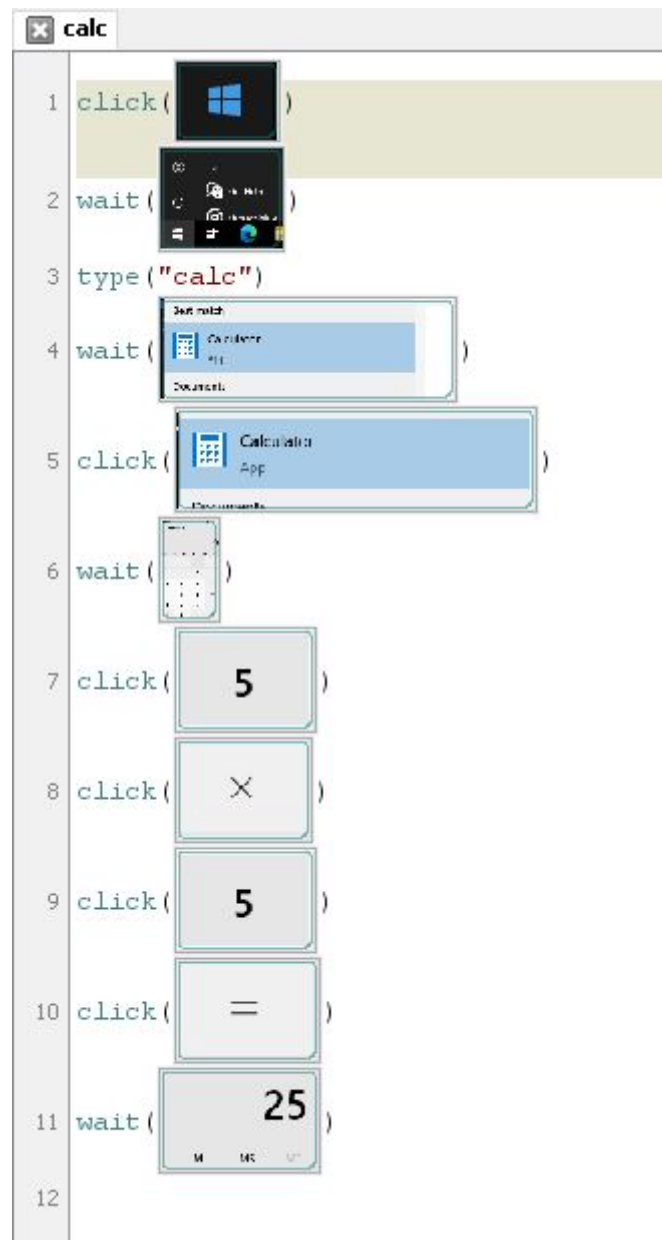
- 自動実行コードを含んだ .py ファイル
- ウェブブラウザ制御イメージ

実行の簡単な例としては、Windows の電卓が正しく動作するかどうかを確認します。一度に1つずつスライドさせる方法を見てみましょう。

1. “click” アクションを選択し、アクションをしたいエリアを選択します。画面が “ エリア選択 ” モードに変わります。



2. type アクションとテキスト “calc” を入力し、電卓が表示されるのを待ってクリックします。電卓が操作できるようになったら、次のようにアクションを入力します。



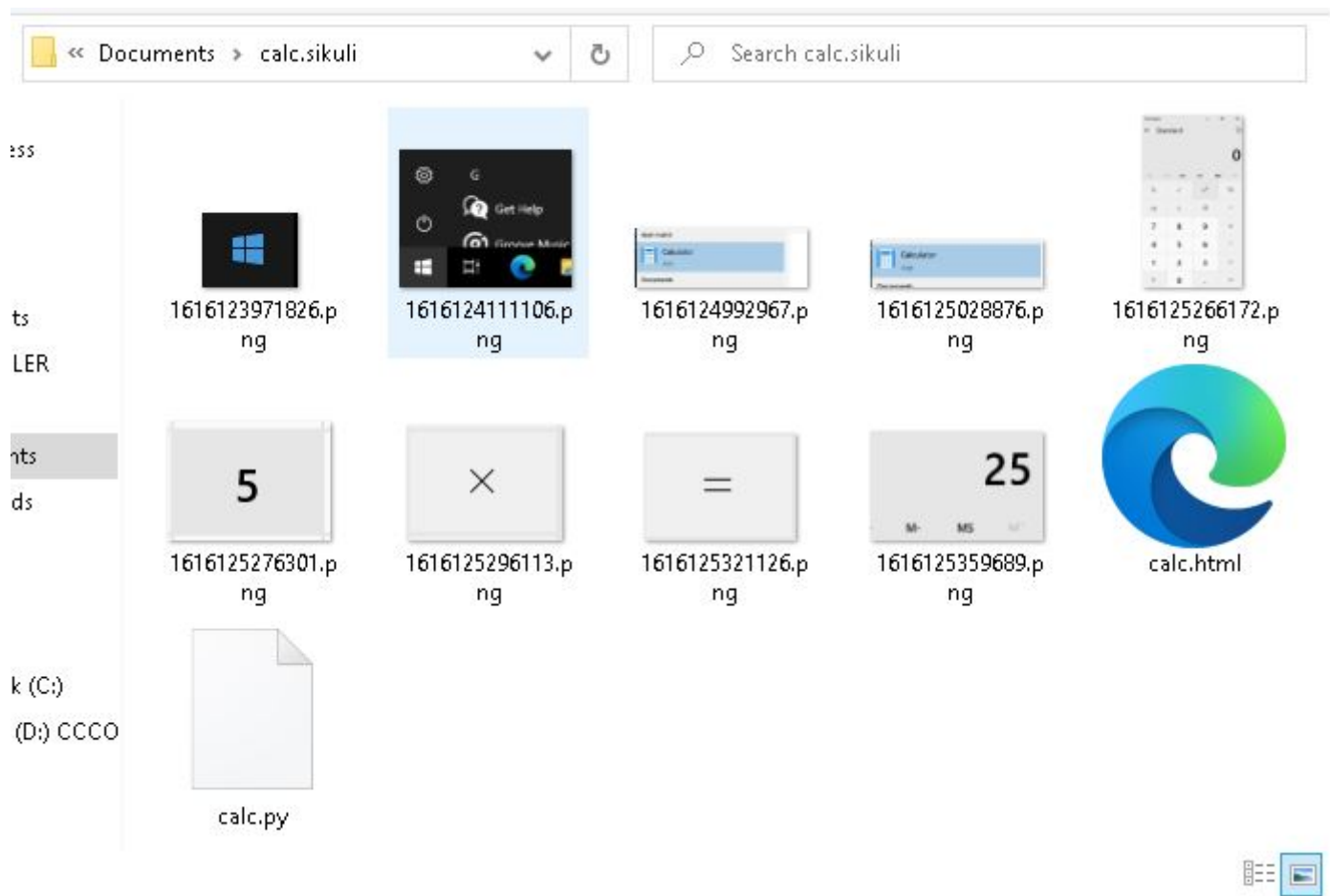
3. 関連する部分をクリックし、以前と同様の領域を選択します。

4. 最後に、保存して Run をクリックして実行します。

重要な注意点:

- 記録環境内の任意の画像をダブルクリックすると、クリックする場所を正確に選択するなど、コントロールキャプチャの詳細を調整できます。
- オペレーティングシステムの遅延によって実行が停止しないように、各クリックの間に「待機」アクションを入力することを強くお勧めします。
- レコーダーはスクリーンショットのような領域を探しますので、ボタンがその上に置かれているときに強調表示されているものがあれば特に注意する必要があります(たとえば、カーソルがそれらの上にあるときに電卓のボタンがオレンジ色に変わります)。

記録を保存したフォルダに、python ファイルと画像ファイルがあります。



注意: 次のコードを使用して PDR レコーディングエンジンからによるキャプチャ内容をカスタマイズできます。

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

これは、指定したファイルに、指定した特定の時間のアクティブウィンドウのスクリーンショットを作成します。座標を使用してスクリーンショットをカスタマイズすることができます。次の構文を用います：

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

ここで、変数は以下の通りです。

X: キャプチャしたい領域の水平方向の位置

Y: キャプチャしたい領域の垂直方向の位置

W: キャプチャしたい幅

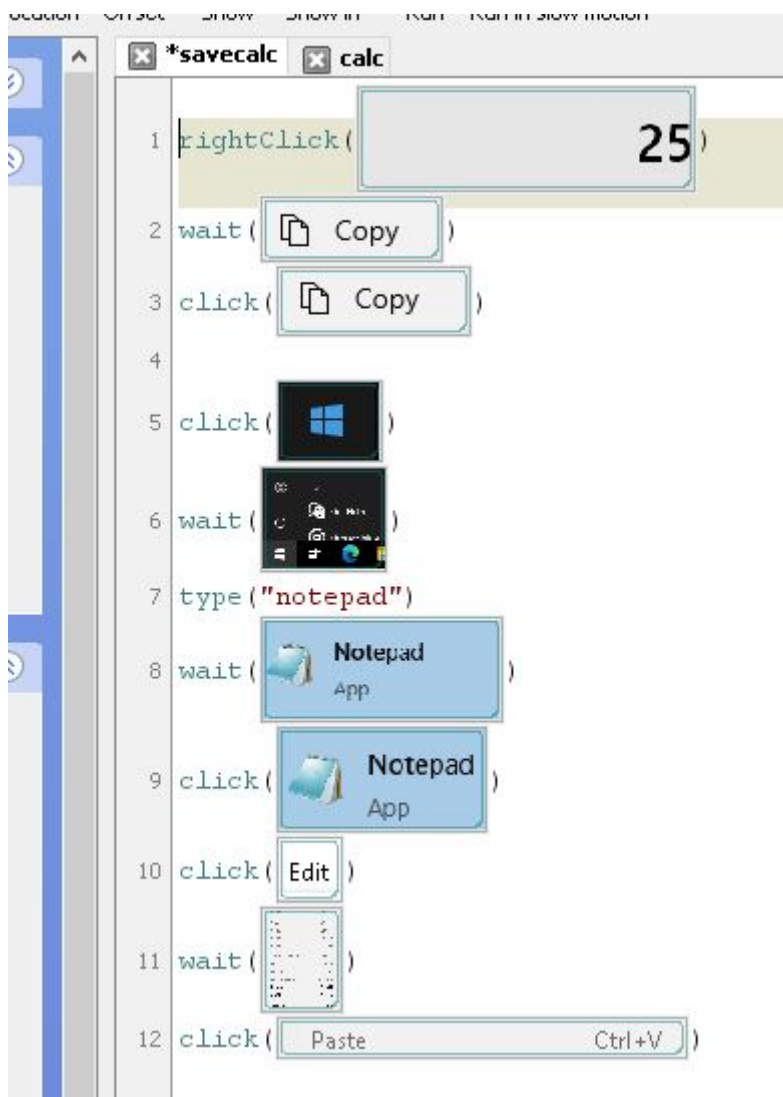
H: キャプチャしたい高さ

Pandora UX PDR でトランザクションセッションの記録

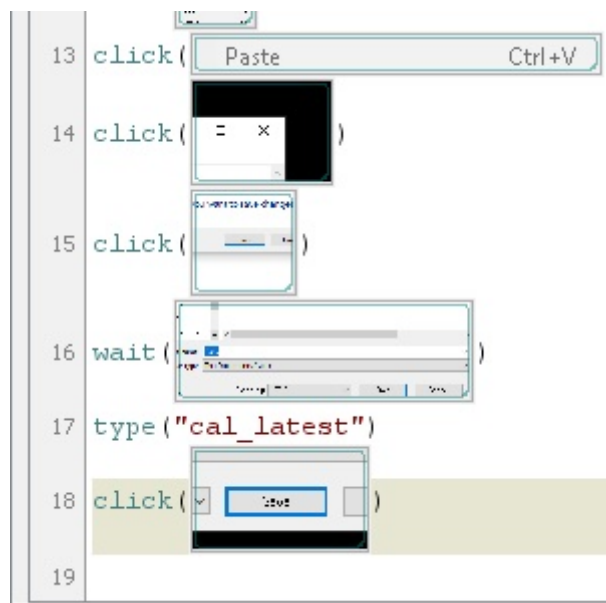
特別なセッションの記録は必要ありません。異なるスクリプトでセッションを記録するだけです。Pandora UX は、複雑なトランザクションの結果を返し整理します。次の段階では、正しく動作させるための実行の呼び出し方を見ていきます。

トランザクション処理を作成するために複数のレコーディングを行う場合は、記録時に探している要素が存在することに注意する必要があります。単一のトランザクション処理ごとに、レコーディング内容を手動で実行することをお勧めします。手動で実行してみたあとに、アクションが期待どおりに行われるようにします。

次の例では、上記の例のプロセス次の段階を構成する記録プロセスを確認できます。電卓の結果をメモ帳にペーストします。以下に例を示します。



ここでは、テキストファイルを特定の場所に保存し、前のファイルを上書きするという別の手順を示します。これにより、柔軟性が大幅に向上し、並行してファイルの監視が可能になり、ヘビー級のデスクトップアプリケーションに対するさまざまな情報を扱えます。この3つ目のシーケンス例は、以下のステップからなります。



PDR セッションの実行

標準実行

あらかじめ記録した PDR セッションを起動するには `pdr.cmd` ファイルを置いた場所とともに動作モードを指定します。このファイルの引数は “-r” セッションガイドラインを含むファイル(-script)を指定し、スクリーンショットが保存されるディレクトリ(-folder)の指定は “/” で終了します。実行失敗時にスクリプトを再実行するための `-retries` オプションも利用できます。

次の実行では、スクリーンショットはアクティブウインドウのみが対象です。

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

次のモジュールが返されます。

- UX_Time_project_name
- UX_Status_project_name
- UX_Control_Snapshot_project_name (初回の実行時のみ)

フェーズでエラーが発生した場合は、次のモジュールが作成されます。

- UX_Snapshot_project_name

エラー時には、アクティブウインドウの画像が表示されます(-ss_config が有効な場合)。

正常時の実行結果例:

```
<module>
  <name><![CDATA[UX_Status_calculadora.sikuli]]></name>
  <type>generic_proc</type>
  <![CDATA[1]]>
```

```

<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>
  <![CDATA[20.204]]>
  <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>
  <![CDATA[data:image/png;base64,
IBCAIAAAA0CnfhAAAAAXNSR.../4x79e/7757f8H2C00s1C73yMAAAAASUVORK5CYII=]]>
  <description><![CDATA[Control image rebuilt]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_calculadora.sikuli</module_parent>
</module>

```

エラー時の実行結果例:

```

<module>
  <name><![CDATA[UX_Status_std.html]]></name>
  <type>generic_proc</type>
  <![CDATA[0]]>
  <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>
  <![CDATA[15.463]]>
  <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>
  <![CDATA[data:image/png;base64, iVBORw0KGgoAAAANSUHEUgAA...JRU5ErkJggg==]]>
  <description><![CDATA[Image (last error)]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>

```



```
<module_parent>UX_Status_std.html</module_parent>
</module>
```

すべて問題なければ Windows マシンの Pandora FMS エージェントのプラグインとしてコマンド設定を行います。以下に例を示します。(1行です)

```
module_plugin C:\Users\artica\Documents\Product\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"
```

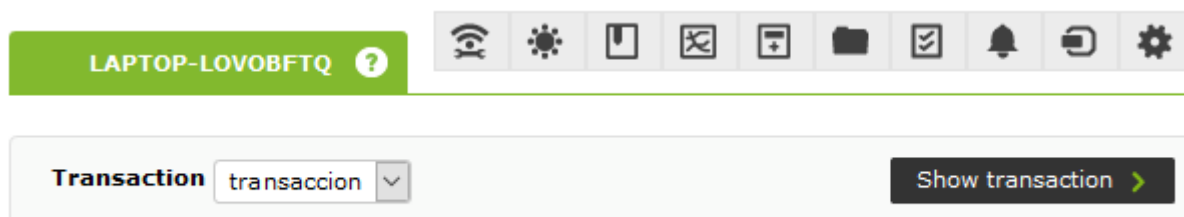
ご覧の通り、実行にはオプションが追加されています。パラメータ `-checkpoint` は、エラーが無かったとしても最終結果のスクリーンショットを表示します。`-post` は、セッションの実行が完了した後にアクションを実行します。この場合、ウインドウを多重に開くなどを防ぐために、電卓のプロセスを終了します。

エージェントに `module_plugin` の設定ができれば、プロセスモードでの起動準備完了です。管理者権限のコマンドラインから以下を実行します。

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

“PandoraAgent.exe” の場所はインストールに依存します。

起動すると、どのようにアクションが記録に従って自動的に実行されるかを見ることができます。ここからは、エージェントのプロセスは終了してはいけません。また、リモートデスクトップで開始したセッションや実行は中断してはいけません。マシンには触れない状態にしておきます。これが、仮想マシンが必要である理由です。



他の自動監視のテストがある場合、次のようにして Pandora FMS に読み込むことができます。

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

- `<exe of the automation system>`: 自動化システムの exe (実行可能ファイル)。
- `<system arguments>`: システムの引数。
- `<test file path>`: テストファイルのパス。

トランザクションベースの実行

いろいろな処理が記録され、正しく機能することが確認されたら PDR で以下を実行します。

```
C:\Users\artica\Documents\Product\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
```

見ての通り、`-script` のパラメータで新たなスクリプトの場所を指定する必要があるだけです。前のスクリプトとはカンマで区切ります。また、`-t` パラメータですべての異なるフェーズをカバーするトランザクションの名前を指定します。より多くのフェーズがあったとしても設定の考え方は同じです。以下に例を示します。

```
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
```

この例では、次の行をエージェントの設定ファイルに追加します。

```
module_plugin C:\Users\artica\Documents\Product\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"
```

`-checkpoint` パラメータで、それぞれのフェーズの最終スクリーンショットを Pandora コンソールに表示することができます。

フェーズごとに、以下のモジュールが返ります。

- UX_Time_project_name.phase_order
- UX_Status_project_name.phase_order

フェーズでエラーになった場合は、次のモジュールが作成されます。

- UX_Snapshot_project_name.phase_order

エラーの場合は、エラーが発生した時点の画像が表示されます。

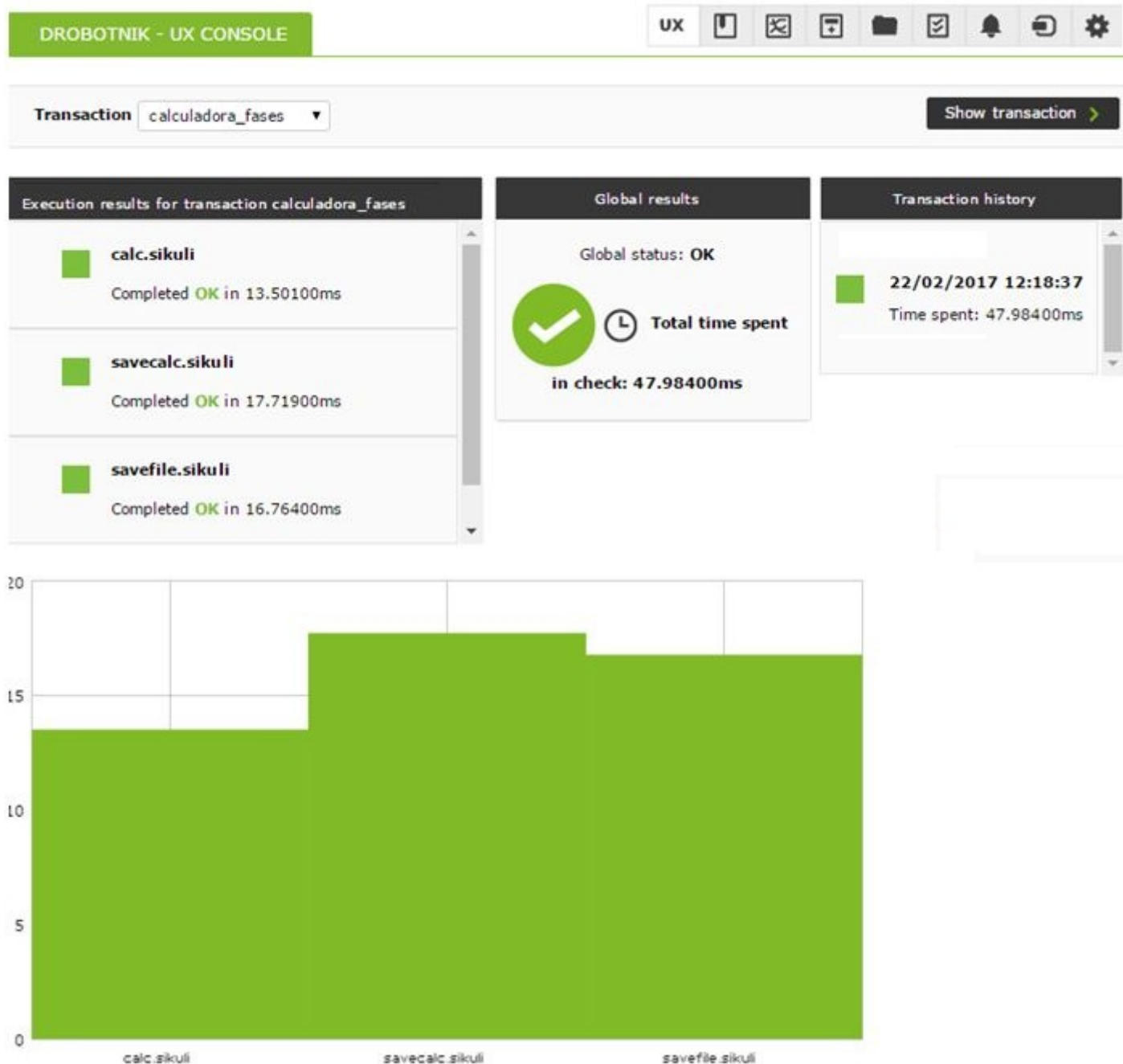
次の名前で全体のまとめのモジュールも返されます。

- UX_Global_Time_project_name
- UX_Global_Status_project_name
- UX_Global_Snapshot_project_name

エージェントがプロセスモードで動き、モジュールを Pandora FMS が受け取ると、コンソールで見ることができます。繰り返しになりますが、**階層モード**で、モジュール間の関係とフェーズ間の区切りがわかりやすく表示されます。

	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

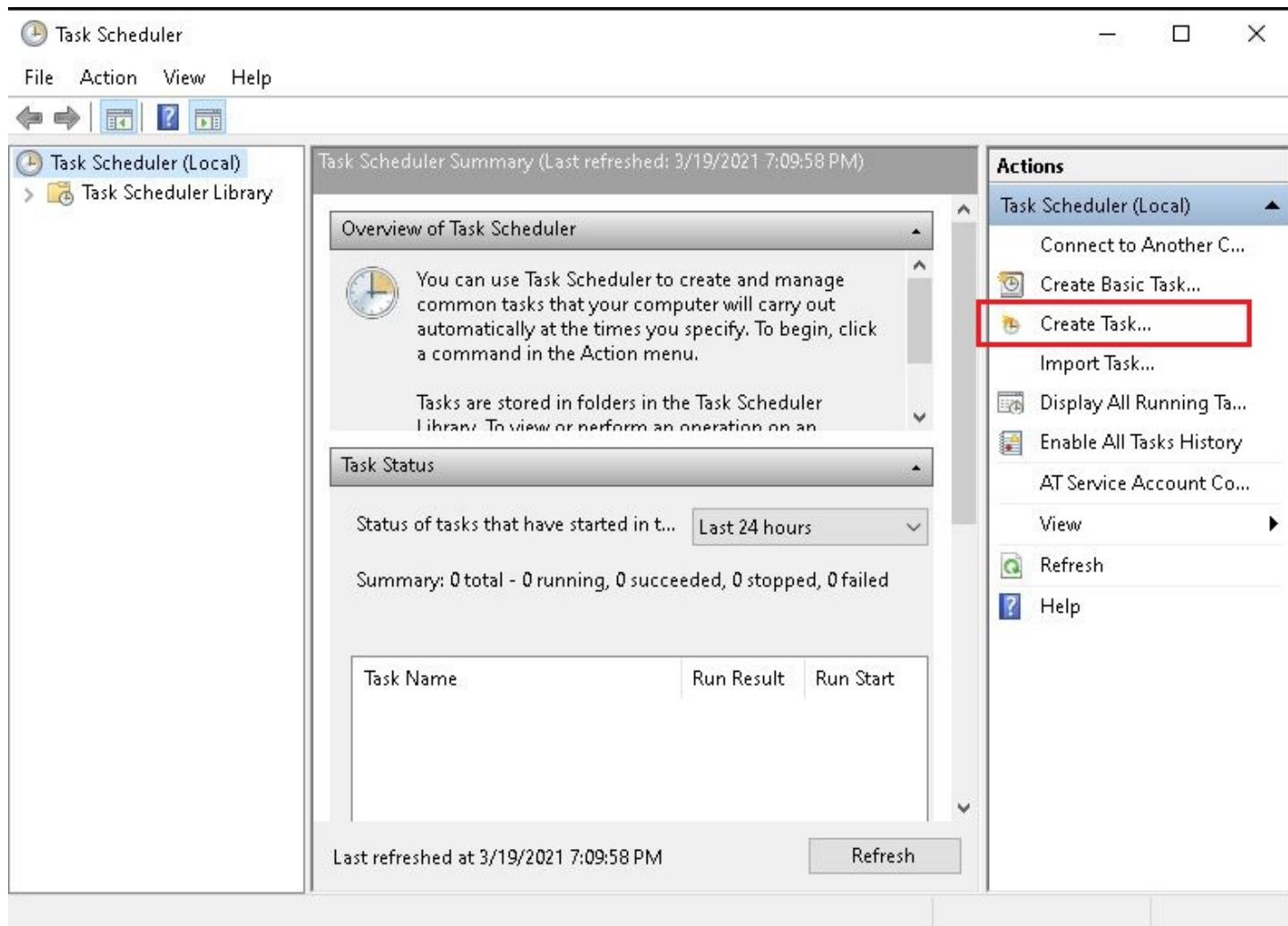
フェーズとタイムチャートの詳細は、トランザクションの画面で見ることができます。



エラーのスクリーンショットは、UX クライアント(PWR)と PWR サーバが共に同じマシンで動作しているときにのみ表示されます。そうでない場合は、Pandora で画像を表示するために PWR サーバの画像配布ディレクトリがクライアントからアクセスできる必要があります。

同じモジュールを使用して、アラートを作成し、システムの時間経過に伴う変化を表示するために、履歴表示やグラフ生成ができます。

システムを再起動したときなどに実行の中断を避けるために、システムが起動したときに Pandora エージェントをプロセスモードで起動させるプログラムタスクを作成することと、パスワード無しでマシンへ自動ログインし、エージェントが確実に実行されるようにすることをお勧めします。



The screenshot shows the Windows Task Scheduler application window. The title bar reads "Task Scheduler". The menu bar includes "File", "Action", "View", and "Help". The left sidebar shows "Task Scheduler (Local)" and "Task Scheduler Library". The main pane displays the "Task Scheduler Summary" for "Task Scheduler (Local)", last refreshed on 3/19/2021 at 7:09:58 PM. The summary includes an overview of the tool and a "Task Status" section showing 0 total, 0 running, 0 succeeded, 0 stopped, and 0 failed tasks. A table with columns "Task Name", "Run Result", and "Run Start" is visible but empty. The "Actions" menu is open on the right, with "Create Task..." highlighted by a red rectangle. Other menu items include "Connect to Another C...", "Create Basic Task...", "Import Task...", "Display All Running Ta...", "Enable All Tasks History", "AT Service Account Co...", "View", "Refresh", and "Help".

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

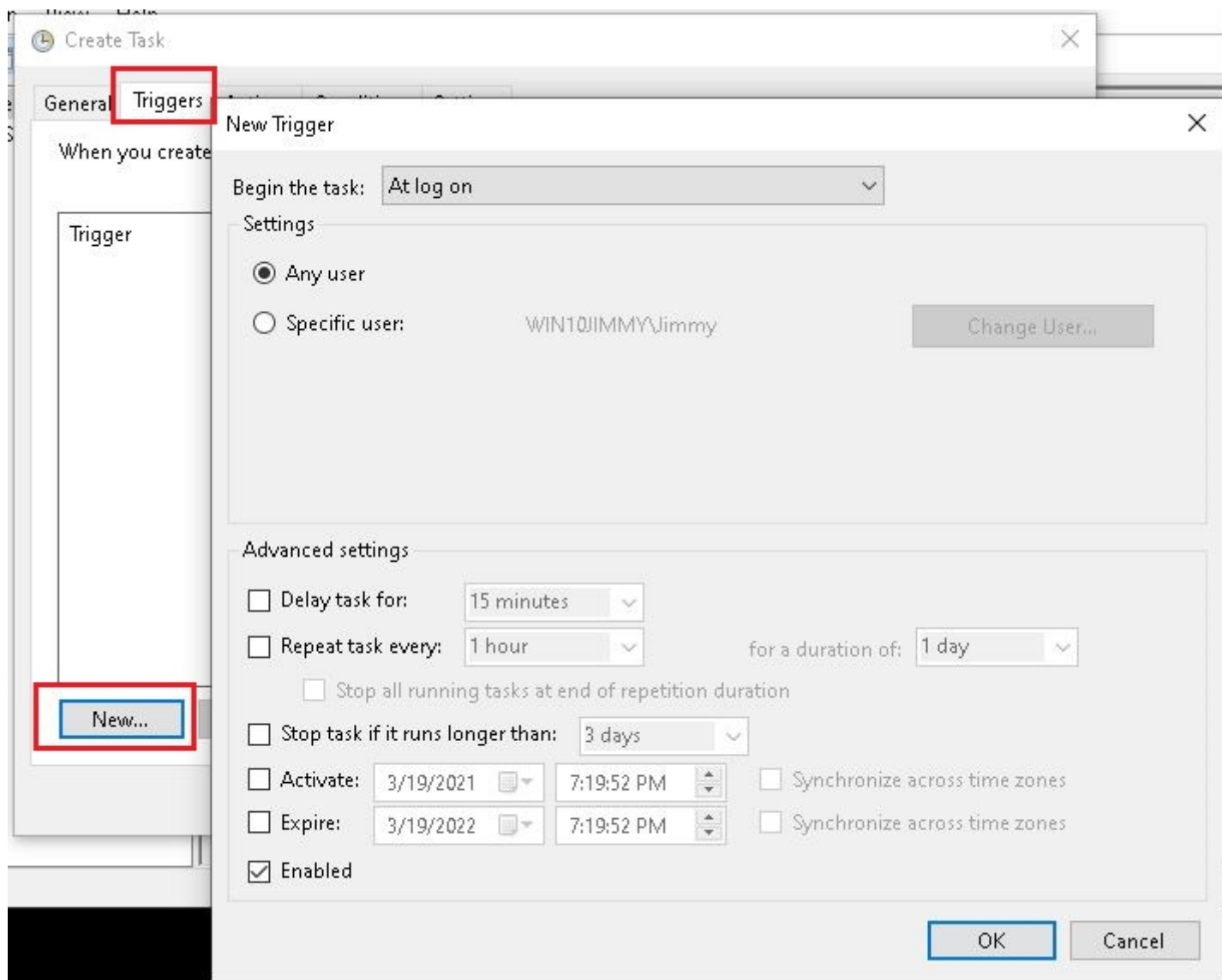
Do not store password. The task will only have access to local computer resources.

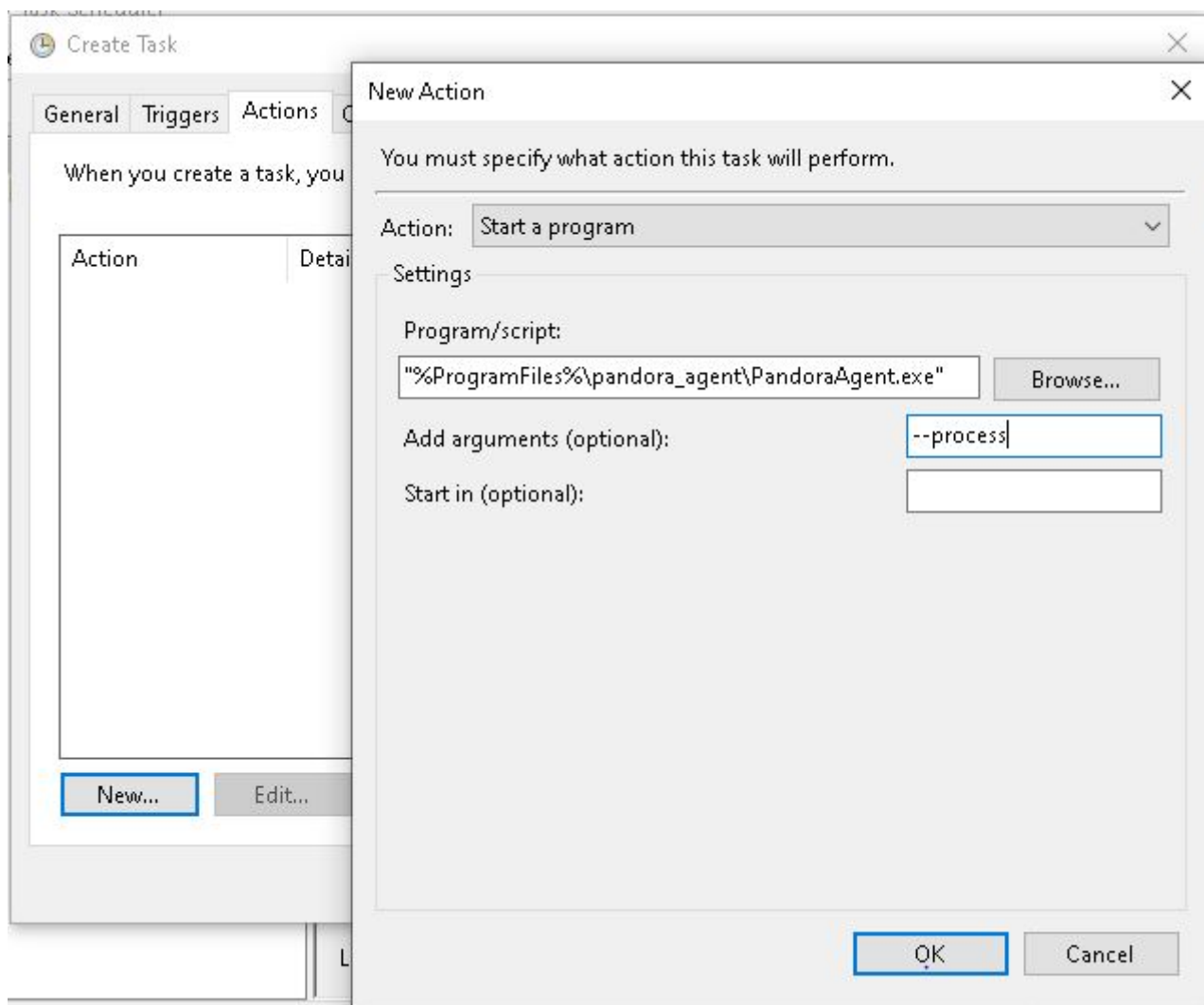
Run with highest privileges

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

これにより、対象の Windows インスタンスでは、マシンが再起動しても Pandora FMS エージェントが常にプロセスモードで実行され、常に PDR プロブによって収集されたデータを送信できることが保証されます。

さまざまなデスクトップシステムでは問題が発生する可能性があります。そのため、前述の設定を使い、デスクトップモードで自動起動を単一のデスクトップを使用することをお勧めします。

[Pandora FMS ドキュメント一覧に戻る](#)