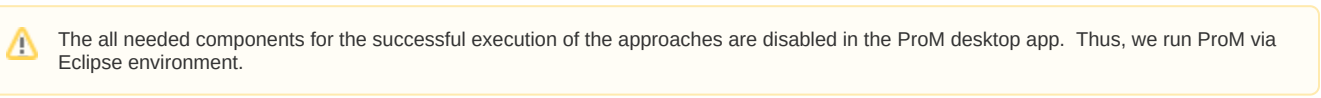
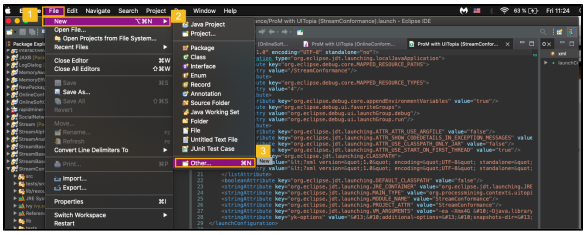
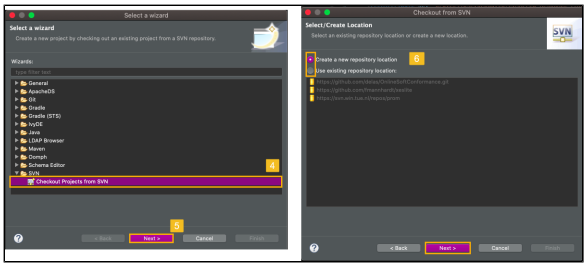
ProM's Setups.

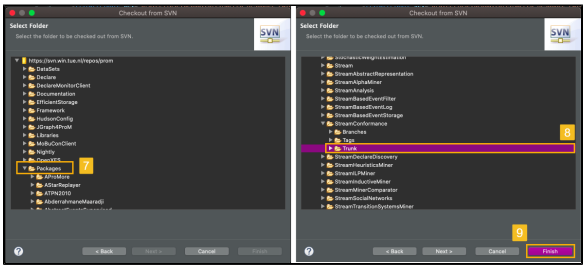


The first step illustrates how to upload an SVN project (in our case StreamConformance package) into the Eclipse environment. The OnlineSoftConformance package is added as project from the local machine. Further, the configurations for both packages have to be applied. Then it describes the even stream generator ProM's plugin. Finally it demonstrates an example of Online Conformance procedure and Offline computations.

1. Upload a single ProM's package

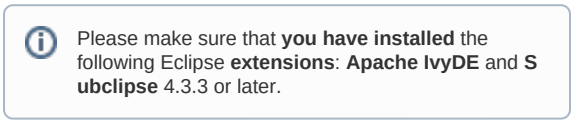






1. Navigate to "File" on the top menu bar.
2. Click on "New".
3. On the drop down menu select "Other"

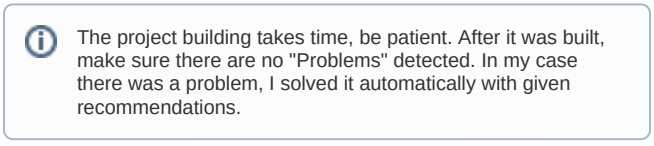
   "Select a wizard" window appears.



4. Navigate to the SVN folder and select "Checkout Projects from SVN"
5. Proceed to "Next".
6. If you have not added any repositories yet:

   Click on "Create a new repository location", add a location (Url) https://svn.win.tue.nl/repos/prom/, and click "Next". Otherwise, select the https://svn.win.tue.nl/repos/prom/ repository and click "Next".
7. Navigate to "Packages".
8. Find and select a package of interest (StreamConformance) and then click on "Trunk".
9. Click on "Finish".

   Project is uploaded and displayed on the right side.



For other details, please refer here .

2. Running configurations

If any packages were not run before, then first run the ProM Package Manager.launch (StreamConformance) file, and install all plugins.

1. Open "Run Configuration" of the **ProM Package Manager.launch** (StreamConformance). On the tab Arguments, insert these VM arguments:

```
-ea -Xmx1G -Djava.library.path=.
```

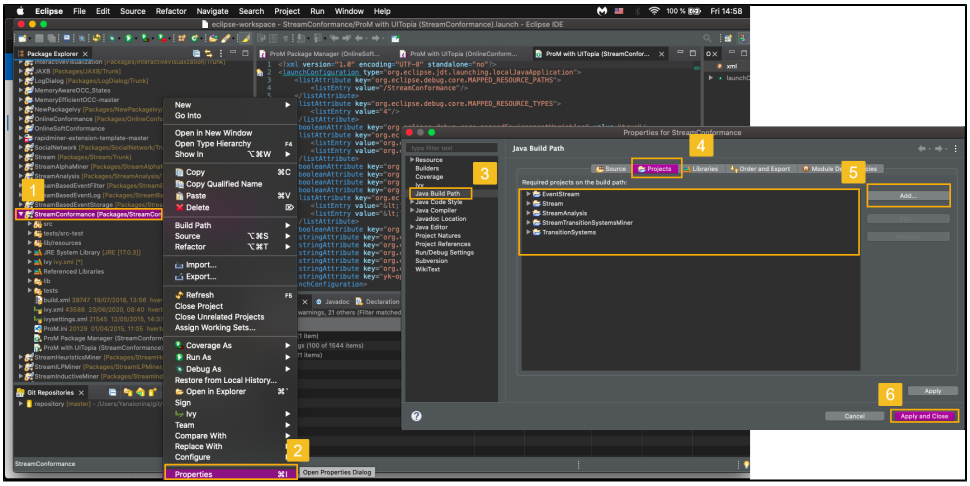
2. Open "Run Configuration" **ProM with UITopia.launch** file of the StreamConformance package, navigate to the Arguments tab, and insert the following arguments to the VM arguments field:

```
-ea -Xmx12G
-Djava.library.path=//lib;
-Dsun.awt.disablegrab=true
-Djava.util.Arrays.useLegacyMergeSort=true
-Djava.system.class.loader=org.processmining.framework.util.ProMClassLoader
```

3. Navigate to the JRE tab and check the current "Runtime JRE".

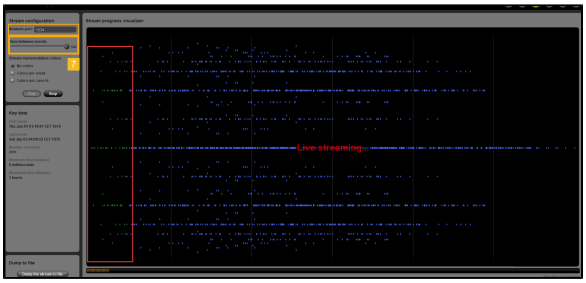
If it is selected, for example, JRE 17.0.3 by default, **it is highly recommended** to choose an Alternative JRE related to Java 8.

4. Click on "Apply".
5. Then add all supplementary packages (listed under step 5) to the Eclipse environment (as described in the section 1) and add them into "Java Build Path". And click apply.



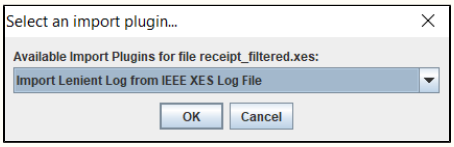
6. And then we run the **ProM with UITopia.launch** file of both packages.

3. Event Stream Generator



1. Input to ProM a static finite event log as an input.

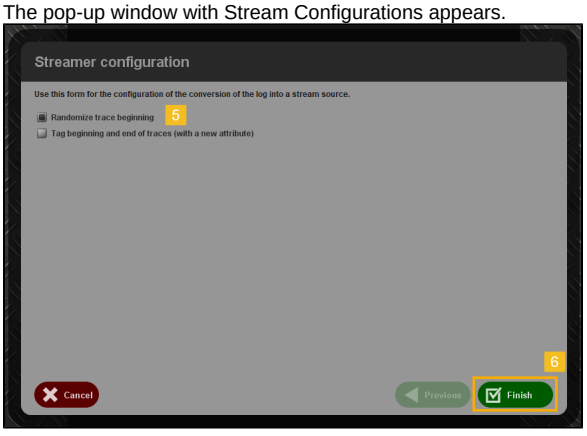
If ProM shows an error while uploading a static log, select this option:



It may happen after manipulations with the event log with PM4Py.

2. Select the Log Streamer plugin.
3. As a result the out will be configured as an "Log Streamer".
4. Click "Start".

The pop-up window with Stream Configurations appears.



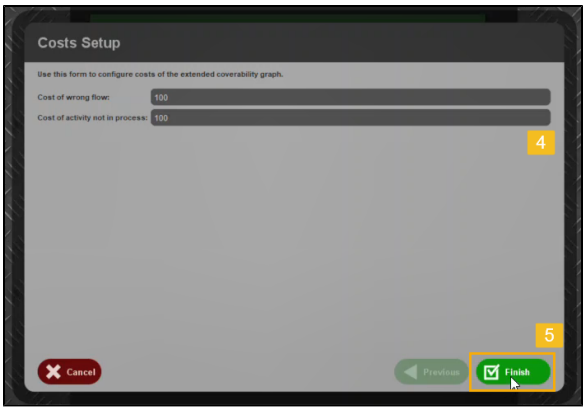
5. Select "Random trace beginning".
6. Click "Finish".

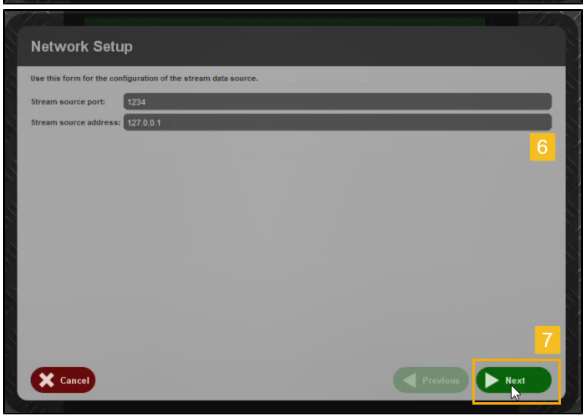
7. Set up the time between events (for all our cases we used 0.5 events/sec).
8. Then click "Play" (on the screenshot it is disabled because it is an example of the running stream)

The "Network port" is set up by default. We connect to this port with algorithms in order to receive events for online conformance checking.

ℹ The red rectangle shows an example that all green small triangles are event streams that have been already released.

4. Online Conformance dashboard example

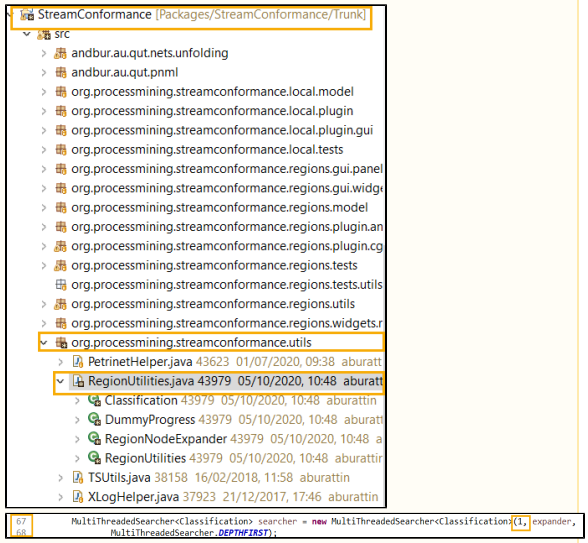




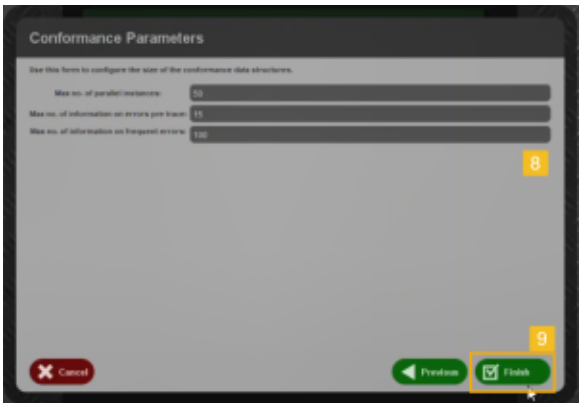
ℹ The particular example relates to Region-based approach (StreamConformance package)

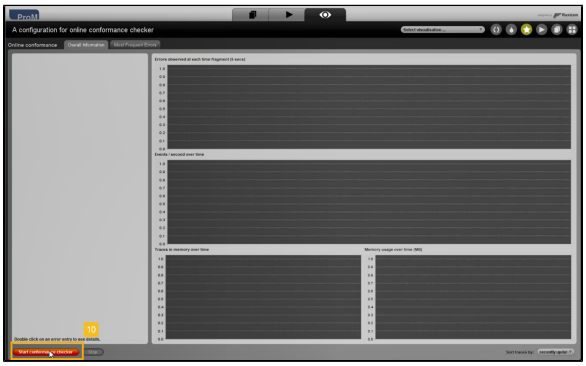
⚠ If a device consists of more than 4 logical processors, navigate to the "RegionUtilities.java" file

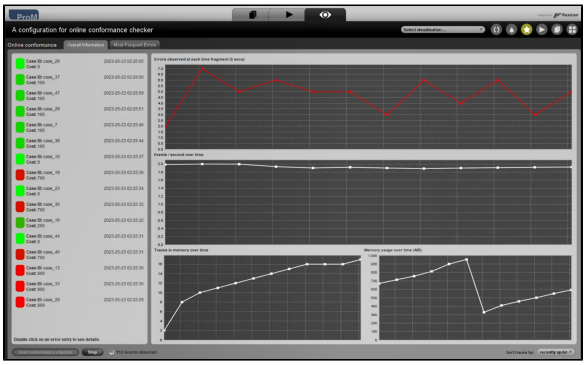
and add "1," on lines 67-68 before "extender".



1. Input a model and corresponding initial marking.
2. Select the algorithm.
3. Click "Start".
4. Set up cost configurations.
5. Click "Finish".
6. Connected to the stream network port as described above.
7. Click "Next".
8. Set up conformance configurations (e.g. how many traces can be observed in parallel, etc.)
9. Click "Finish"
10. The online conformance checking dashboard is shown, click on the "Start conformance checker" button below.
11. Online Conformance Checking algorithm is successfully running by consuming an evet stream.



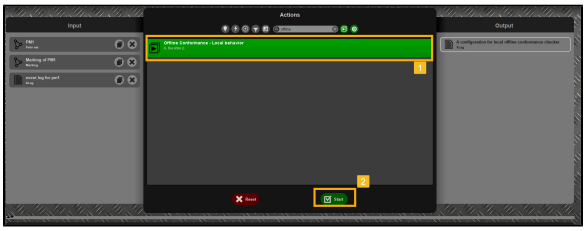




5. Offline results extraction

Offline Behavioural Patterns Approach

1. Input a static event log, a corresponding process model and marking, find in the search bar the "Offline Conformance - Local behavior" plugin.

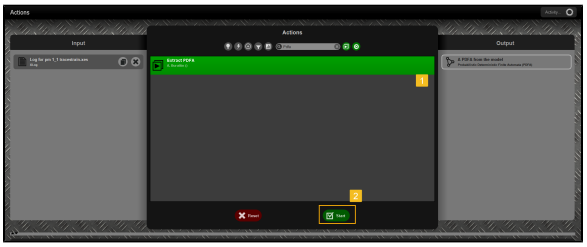


2. Click on "Start" button.

3. It returns the the overview on the event log. Then from drop-down menu select "Log Pattern Explorer".

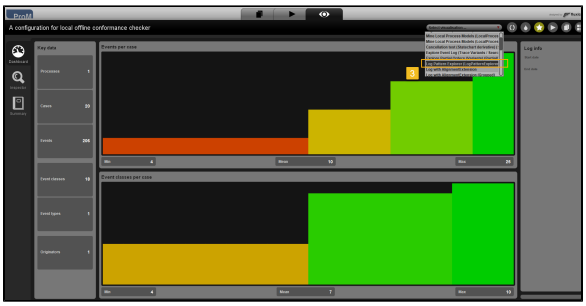
Offline Soft Conformance

1. Input an event log (either train set or filtered on variants) and select the "Extract PDFA" plugin.
2. Click "Start".

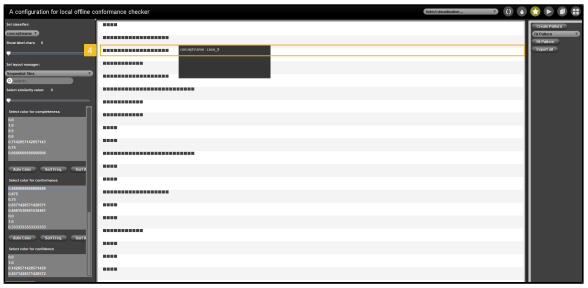


3. Insert the attribute name "concept:name".

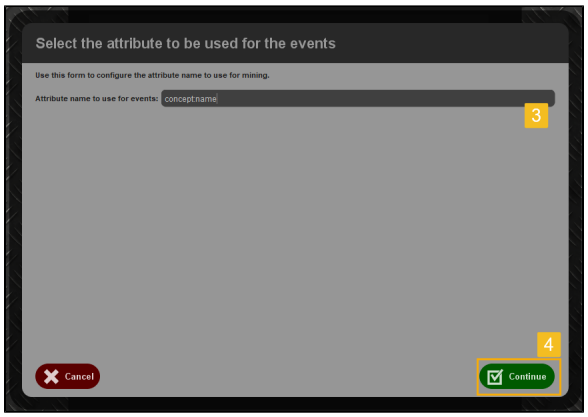
4. Click on "Continue" button. It returns a General Model.



4. Navigate to the line in order to define a trace ID.



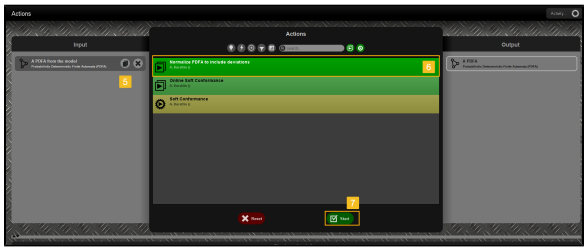
5. The 3 computed dimensions are displayed by navigation to the last event.



5. Input General Model.

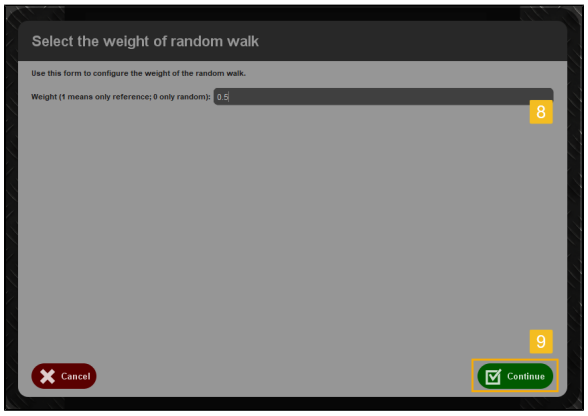
6. Select the plugin "Normalize PDFA to include deviation". (creates a stochastic normalized model)

7. Then click "Start"



8. Choose a weight factor.

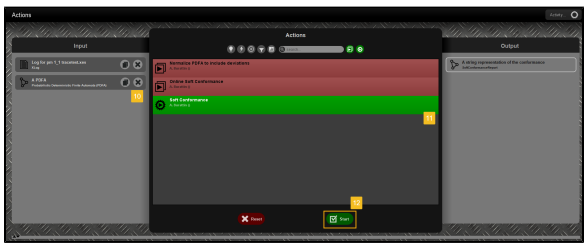
9. Then "Continue".



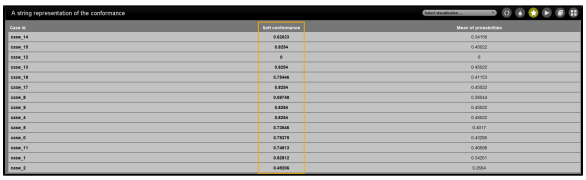
10. Input normalized stochastic model and, for example, test set

11. Select the "Soft Conformance" plugin.

12. Click "Start".



It returns results of Offline Soft Conformance computations.


A string representation of the conformance
Select visualization ...
Case id
Soft conformance
Mean of probabilities
case_14
0.62623
0.34108
case_15
0.9254
0.45822
case_12
0
0
case_13
0.9254
0.45822
case_16
0.75446
0.41153
case_17
0.9254
0.45822
case_8
0.66748
0.38944
case_9
0.9254
0.45822
case_6
0.9254
0.45822
case_5
0.73846
0.4117
case_0
0.79278
0.43266
case_11
0.74612
0.40908
case_1
0.62812
0.34201
case_2
0.49206
0.2664