Importing IEEE XES files to PM4PY

IEEE XES is a standard format describing how event logs are stored. For more information about the format, please study the IEEE XES Website. A simple synthetic event log can be downloaded from here. Note that several real event logs have been made available, over the past few years. You can find them here.

Function 1 – Importing IEEE XES files

```
from pm4py.objects.log.importer.xes import importer as xes_importer
log = xes_importer.apply('<path_to_xes_file.xes>')
```

Figure 1

The example code on the above [Figure 1]shows how to import an event log, stored in the IEEE XES format, given a file path to the log file.

Print trace of the log

```
print(log[0]) #prints the first trace of the log
print(log[0][0]) #prints the first event of the first trace
```

Figure 2

Event logs are stored as an extension of the Python list data structure. To access a trace in the log, it is enough to provide its index in the event log. Consider the example above [Figure 2] on how to access the different objects stored in the imported log.

Function – apply() method , variant and parameter.
(see the table below for more description on variant and parameter)

```
from pm4py.objects.log.importer.xes import importer as xes_importer
variant = xes_importer.Variants.ITERPARSE
parameters = {variant.value.Parameters.TIMESTAMP_SORT: True}
log = xes_importer.apply('<path_to_xes_file>',
                         variant=variant, parameters=parameters)
```

Figure 3

The apply() method of the XES Importer, i.e. located in pm4py.objects.log.importer.xes.importer.xes.importer.py, contains two optional parameters: variant and parameters. The variant parameter indicates which variant of the importer to use. As

shown in Figure 3 above. The parameters parameter is a Python dictionary, specifying specific parameters of choice. This method invocation style is used throughout PM4Py in the various different algorithms implemented, i.e., by wrapping around the different implementations, new variants of algorithms are easily called, using previously written PM4Py code. With respect to XES importers, two variants are provided. One implementation is based on the iterparse() function of xml.etree. The other variant is a line-by-line, custom parser (for improved performance). It does not follow the standard and is able to import traces, simple trace attributes, events, and simple event attributes. To specify a variant, we add the following argument to the call to the importer: variant= xes importer.Varients.ITERPARSE (note that, in the example code, this is encapsulated in local varaible variant). The xes_importer.Varients.ITERPARSE-value, actually maps on the underlying Python module, implementing the iterparse-based importer. We are able to access that reference, by accessing the value property, e.g., xes_importer.Varients. ITERPARSE.value.xes That module, contains a parameter definition, i.e., Parameter, containing all possible parameters for the iterparse-variant. As an example, parameter TimeStamp sort is one of those, accessed by xes_importer.Varients.ITERPARSE.value.Parameter.TIMESTAMP_SORT. Click the button below, to reveal all variants and corresponding parameters defined for importing IEEE XES files.

more description on variant Iterparse and Line-by-line and its parameter keyswith description

Variant	**Parameter Key**	**Type**	**Default (if we don not specify value)**	**Description**
ITERPARSE	TIMESTAMP_SORT	boolean	False	If True, the log is sorted by timestamp
	TIMESTAMP_KEY	string	Time:timestamp	If timestamp_sort is true, then using this event attribute key to read timestamp.
	REVERSE_SORT	Boolean	False	If true trace indices are added as an event attribute for each event
	MAX_TRACES	integer	10000000 (default size it will take if youdo not provide)	Maximum number of traces to import from the log

Line-By-Line	TIMESTAMP_SORT	BOOLEAN	FALSE	If True, the log is sorted by timestamp
	TIMESTAMP_KEY	String	Time:timestamp	If timestamp_sort is true, then using this event attribute key to read timestamp.
	REVERSE_SORT	Boolean	False	If true trace indices are added as an event attribute for each event
	INSERT_TRACE_INDICES	Boolean	False	(same as iterparse)
	MAX_TRACES	Integer	1000000000	(same as iterparse)
	MAX_BYTES	Integer	1000000000	(same as iterparse)

Source - PM4PY documentation