

 **Fix Me!** This page is not fully translated, yet. Please help completing the translation.
(remove this paragraph once the translation is finished)

SimpleMilestonesHandler

Recalculates the lead time of activities in order to reach given milestones.
Milestones cannot be scriptnodes and must contain a duration.

The action handler proceeds with each milestone as described here (starting with the first):

1. The action handler goes through all paths, continuously adding the duration each activity, until the milestone is reached.
2. The paths are sorted in descending order according to their total duration.
 - 2.1. If an assignment is completed at this moment, the duration of the now-completed activity is added to the actual required lead time.
 - 2.2 Begin with the longest remaining path.
 - 2.3 Thereby, the total duration will be recalculated (if necessary with already-shortened durations), and a determination is made about whether the milestone date can be upheld.
 - 2.3.1 If the date is maintained, the process advances to the next milestone, which is then assessed starting with step 1.
 - 2.3.2 If the submission date cannot be upheld with the current durations, a calculation will be made to determine what percent of the duration of all activities in the path must be subtracted in order to reach the milestone. The percentage is then subtracted from the durations of all unfinished activities in the path. Lastly, the critical path is recalculated.

Caution! `calculateTimesForAll` must be set up! If the calculation should be after every activity, it is advisable to use the `AddActionHandlerToNodes`. The function “calculate critical path” must be activated in iGrafx.

Action Class

```
com.dooris.bpm.actionhandler.SimpleMilestonesHandler
```

Event Type

node-leave IMPORTANT!

Action Name

any name

Mandatory Fields

none

Parameter

path

Path to txt-file, in which the milestones are stored.
All possible paths up to the milestone are stored in the txt-file, .
Example

```
path=/jboss_tim/server/default/deploy/loom.ear/loom-portal.war/custom/super/milestones.txt;
```

Explanation (see [example TXT File](#))

dates

IDs of the fields in which the data can be written.
If there are multiple milestones, these are separated by commas.

Example

```
dates=${milestone_1},${milestone_2},${milestone_3};
```

Explanation (see [example](#))

dateformat

The format in which the dates of the milestones are formatted. Here, the SimpleDateFormat is used.

```
dateformat=dd.MM.yyyy HH:mm;
```

Explanation (see [example](#)). Possible options include:

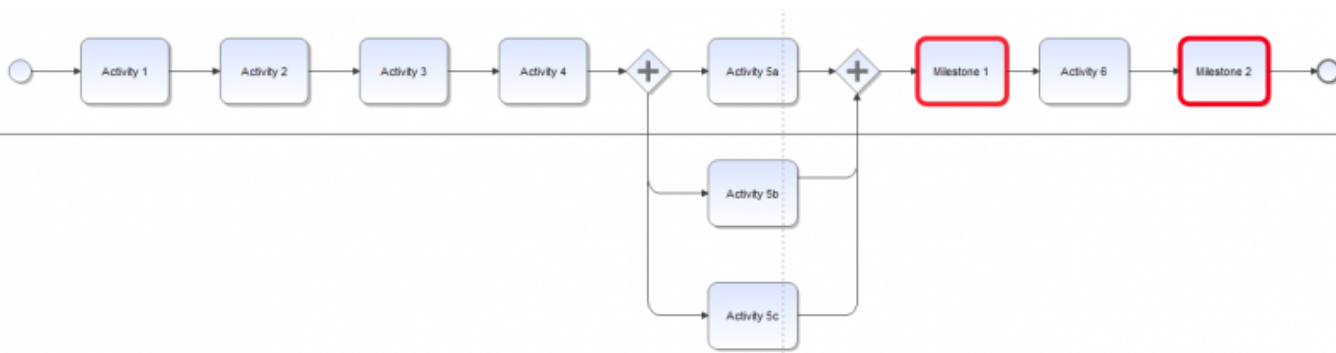
Example TXT File

Explanation: Rows can be commented out with a "#". The activities are separated by a comma. A new row is used for each path.

Milestones are masked with a "\$". The first parameter denotes which date is relevant for the milestone. (0-n)

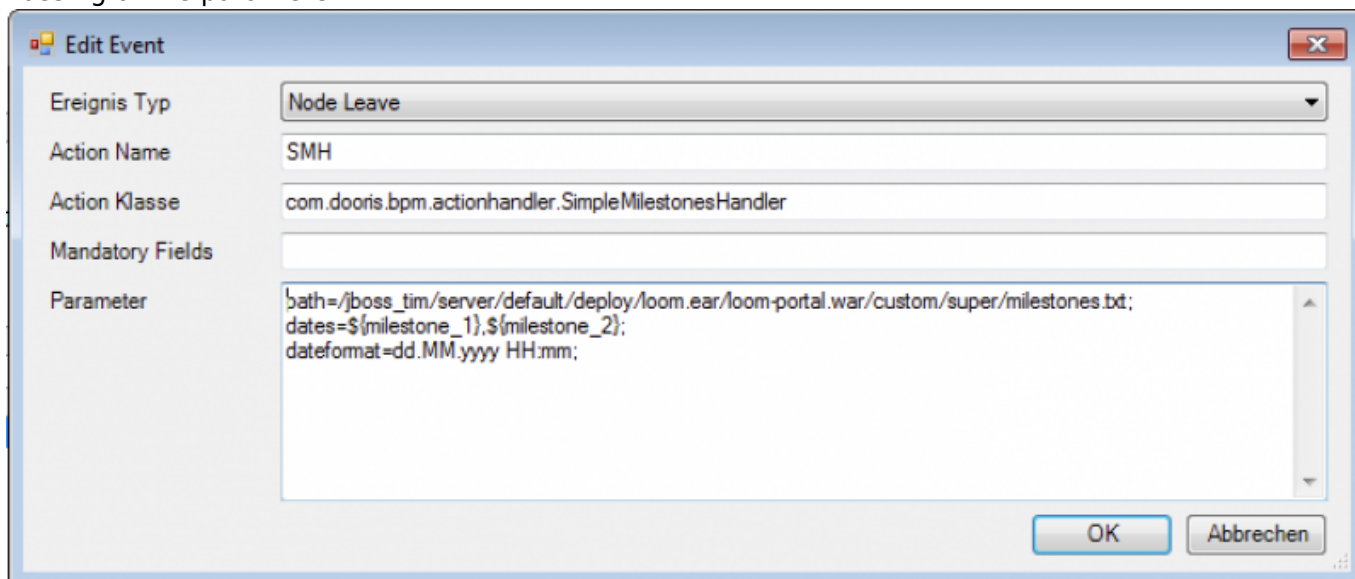
```
# Use # to mark a comment row
#Possible paths to Milestone 0 - Milestone 1
0,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5a,$Milestone 1
0,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5b,$Milestone 1
0,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5c,$Milestone 1
#Possible paths to Milestone 1 - Milestone 2
1,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5a,Milestone 1,Activity 6,$Milestone 2
1,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5b,Milestone 1,Activity 6,$Milestone 2
1,Activity 1,Activity 2,Activity 3,Activity 4,Activity 5c,Milestone 1,Activity 6,$Milestone 2
```

The example applies to the following process.



Example

Passing of the parameter



Dependencies

Available starting with [Version 4.3](#)

From: <https://wiki.tim-solutions.de/> - [TIM Wiki](#) / [NEW TIM 6 Documentation](#)

Permanent link: <https://wiki.tim-solutions.de/doku.php?id=en:software:tim:actionhandler:simplemilestoneshandler&rev=1452784817>

Last update: **2021/07/01 09:54**

