

# SignalByMailReply

## Description

This timer checks mail boxes for e-mails and forwards instances if there are any e-mails which have to be answered in order for further instances to be generated. The timer has to be used in combination with a [VariableDecisionHandler](#) at a node if a decision has to be made after receiving a signal. The sender will be assigned as an actor to all tasks of the current node once the signal has been transmitted successfully. Additionally, all incoming e-mails will be attached to activities as a note, so long as they are able to be correlated to a given activity. [RegExp](#) will be used to select a part of a successful e-mail, which will be brought into the process as a variable (with the name given *Nodename-answer\_successful*).

The corresponding process instance has to be referred to as

```
----- (${SYS.PROCESSINSTANCE_ID}) -----
```

in the text of the e-mail. The answer can either be written in the first line or in the following form

```
----- [Reply: ]-----
```

.

---

## Web service name

```
ProcessInstanceManager
```

## Web service method

```
signalByMailReply
```

## Parameter

email, password, port, host, limit, moveFolder, pildRegExp, pDName, NodeNameRegExp, firstline, box, decisionRegExp, signalSuccessMail, processVariable, transitions new version: username, password, port, host, moveFolder, pDName, NodeNameRegExp, signalSuccessMail, processVariable, transitions, decisionRegExp

### 1. e-mail

This entails the complete e-mail address which will be used to generate further e-mails.

## 2. password

This will show the password used to access the desired e-mail account.

## 3. Port

This provides information on the port corresponding to the e-mail address and the host. Frequently used ports are: 110 *Pop* 995 *Pop including encryption* 143 *IMAP* 993 *IMAP with encryption*

## 4. Host

Host, e.g., pop3.gmail.com or imap.gmail.com (very often this will entail either pop or pop3 or IMAP. SMTP is far less common!). An IMAP mail box will be necessary if moving all e-mails to other folders. The kind of protocol to use is dependent on the mail box being employed. If both common protocols are supported by the web service, then IMAP should be selected as preferred protocol.

Important: If you want to move mails to other folders, an IMAP mailbox is required. Basically, the use of the protocol depends on the mailbox. If both protocols are supported, then IMAP is to be preferred. It is highly recommended to use a separate mail box for each timer and process definition.

## 5. moveFolder

This is not a mandatory field if activated. This is the name of the folder into which successfully processed e-mails for signalling should be moved.

## 6. pDName

Name of the process definition.

## 7. nodeNameRegExp

nodeNameRegExp is a regular expression which provides clues on how to find the name of a node. The textual content of e-mails as well as corresponding subject headings will be scanned for this expression. e.g., ++(.+)+ The element which has to be searched would be the nodeName included in brackets. For example:

```
++Wait++
```

For further elaboration on this topic, please refer to the entry pildRegExp.

## 8. signalSuccessMail

If the value is "true" then an e-mail will also be sent to the sender as a reply confirming the successful receipt of a signal. The reply can be configured within the process by using variables. Please refer to the section provided [below](#) for further information on this topic.

## 9. processVariable

The value of a process variable will be set to the selected transition which then enables the [VariableDecisionHandler](#) at the XOR-Gateway to make a decision.

## 10. leavingTransitions

How will the correct pathway for the process be found if multiple options exist after the current activity? Transitions will be included in the timer as parameters by mapping which exact terms are meant to be detected by this transition. At least one transition has to work properly, which in turn means that at least one of the correct terms has to be included in the text of an e-mail. For example: Approval:ok-O.K.-okay-yes-y-to-approve-approval-accepting-accept-adopting-all-right-alright-agreed-agreeing;rejected-rejecting:-not-okay-not-ok-not-O.K.-no-never-unacceptable

The transitions according to this node are: approval and rejection To signal an approval a response to the e-mail might entail ok, okay, O.K. etc. Alternatively, to signal a rejection words would be used like "not okay", no, never etc.

## 11. reasonRegExp

The decision contained in the e-mail which signals that the process should be saved for access at a later stage of the process. At this point the terms should be used can be determined by searching the text of the e-mail. If the decision is meant to provide a result extending over several rows the RegExp has to use the ending '?s', for example `(.*?)Replies:?s`

## Variables

The replies to invalid messages can be setup either by using a variable or the file `loom.properties`:

*no-tim-user-found-for-signal-by-mail-reply*

*user-not-actor-for-signal-by-mail-reply*

*found-no-transition-for-signal-by-mail-reply*

*found-no-node-for-signal-by-mail-reply*

*signal-done-for-signal-by-mail-reply*

*found-no-process-instance-for-signal-by-mail-reply* Only the configuration using `loom.properties` would be possible in this case.

The correct answer has to be configured in the following way:

*signal-successful-signal-by-mail-reply*

The name of the folder which has to be accessed can also be configured by using `loom.properties` via *inbox-name-signal-by-mail-reply*.

## Attention!

Depending on the exact stage of the procedure the following variables can be implemented:

*emailFrom*, This indicates the sender

*userNameLast*, This entry shows the surname of the TIM user

*userNameFirst*, This variable represents the given name of the current TIM user

*nodeName*, The name of the node that should be signaled is provided in this section

*nodeEndDate*, This displays the date when this node was terminated. This variable is meant to be used only after termination of a node when trying to signal

*nodeEndTime*, This indicates the point in time when this node was terminated. This variable is intended to be used only after termination of a node when trying to signal

*lastUserNameLast*, This shows the surname of the user who terminated the node. This is supposed to be used only when trying to signal after termination of a node

*lastUserNameFirst*, This displays the given name of the user who terminated the node. This is this meant to be used for signalling only after the termination of a node

After sending the signal, the following variables are also available: *nodename-lastActor* *nodename-signalDateTime* (Format: "yyyy-MM-dd HH:mm:ss")

## Procedure

First, the mailbox specified by the timer will be opened to receive incoming e-mails. As a default setting the folder *Inbox* will be retrieved. This setting can be changed by configuring *loom.properties* with the parameter *inbox-name-signal-by-mail-reply*.

In the second step the first five e-mails to be found will be processed. This number of e-mails can be adapted by configuring *loom.properties* with the parameter *inbox-name-signal-by-mail-reply*. With each e-mail, a search will be conducted to find a text according to the form of `---(1234)---` at first in the textual content and then in the subject heading. This text represents the identification number (ID) of the TIM process instance.

The message will be ignored if neither an ID nor a corresponding process instance can be found within TIM.

In the next step a TIM user will be sought while sending an e-mail. The sender will receive an e-mail if no user can be identified.

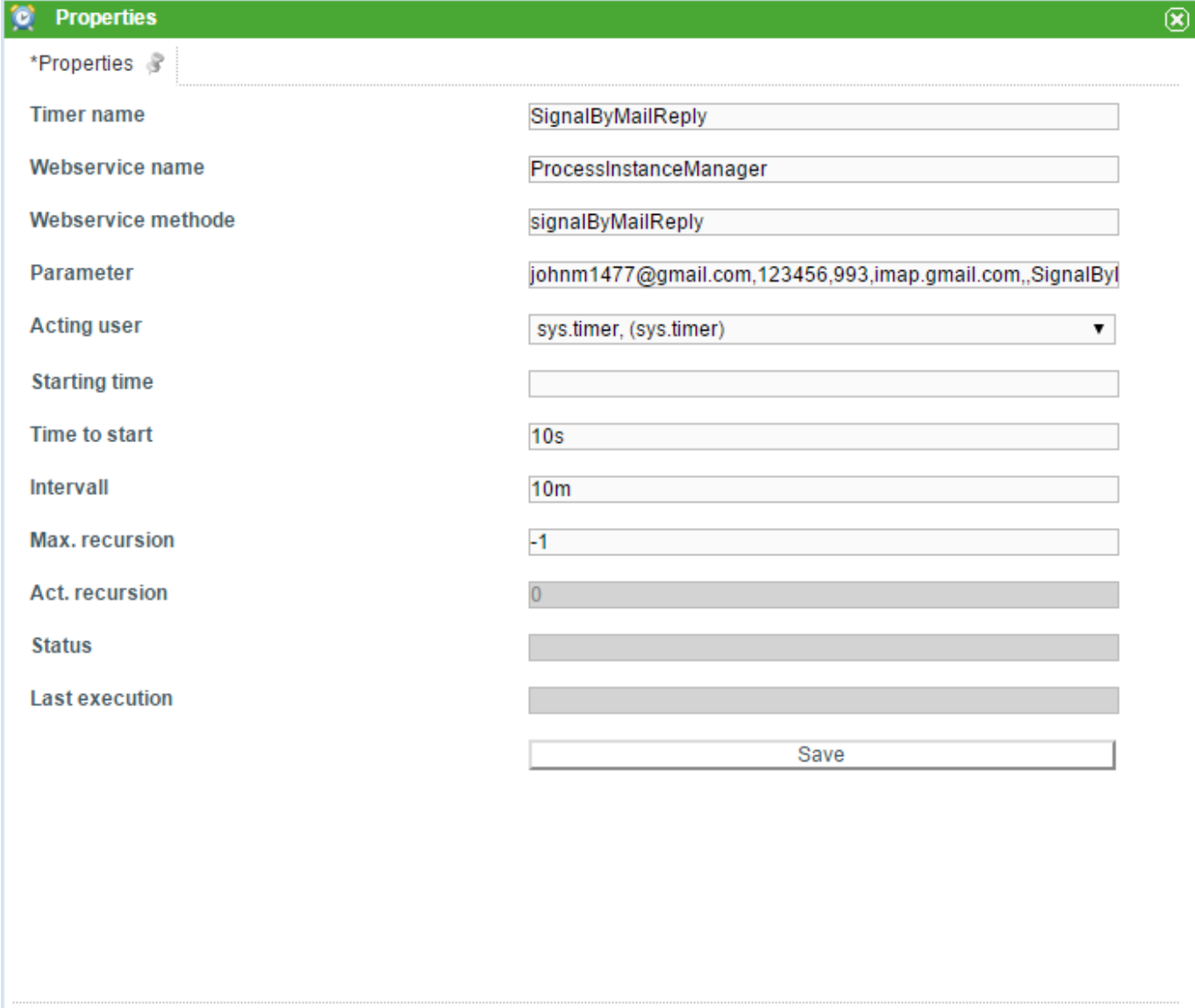
In the following stage the name of an activity will be searched for in the content as well as in the subject heading of the e-mail by using [NodeNameRegExp](#). In case a name is found, an examination will be conducted to check if this process instance is pertaining to the provided [process definition](#). If this is not the case, then this e-mail is not supposed to be assigned to this timer. This e-mail would then also be ignored and no e-mail would be sent as a reply.

If this timer is the proper one pertaining to this particular e-mail, an examination will be started to ensure that the process is also focused on the correct activity. If this is the case the e-mail will be attached as a notice to an activity. In addition, the sender will receive a corresponding message. Now the TIM user who sent the e-mail will be examined to ensure if he/she is responsible at least for one of the tasks related to the current activity. This user will then in turn be informed if this is not the case.

Next, a check for follow-up activities takes place. If there is only one existing activity the user will be assigned the status of an actor who is responsible for all tasks. There are several possibilities to continue with the process and to check if the mail entails a corresponding [decision](#).

Afterwards, the user will receive the status of an actor assigned to the task. Process variables will then be generated and the process forwarded. An e-mail will also be sent if [desired](#). All e-mails that had been answered will be moved to the folder errorMessages if this folder already exists. The initial e-mail which forwarded the whole process will be moved to the [configured](#) folder.

## Examples



The screenshot shows a 'Properties' dialog box with a green header. The dialog contains the following fields and values:

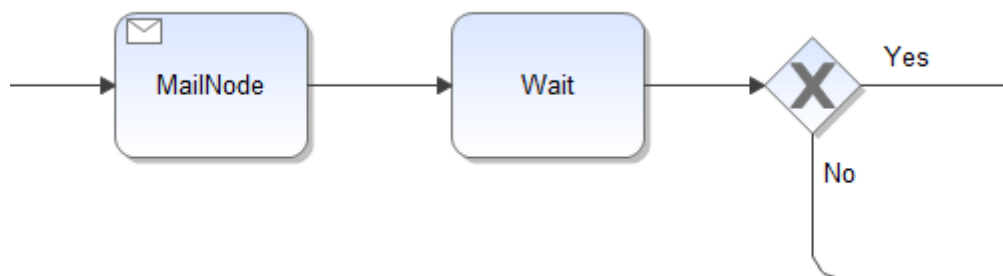
Property	Value
Timer name	SignalByMailReply
Webservice name	ProcessInstanceManager
Webservice methode	signalByMailReply
Parameter	johnm1477@gmail.com,123456,993,imap.gmail.com,,SignalBy
Acting user	sys.timer, (sys.timer)
Starting time	
Time to start	10s
Intervall	10m
Max. recursion	-1
Act. recursion	0
Status	
Last execution	

At the bottom of the dialog, there is a 'Save' button.

Parameter:

```
johnm1477@gmail.com,123456,993,imap.gmail.com,,SignalByMailReply mk4, -  
+{(Wait.*)}-+,true,decision,no:rejected-no-nok;jo:approved-ok-yes,(.*?)wrote  
?s
```

This process is provided as an example:



The following is a simple example for the text of an e-mail:

Hello,

Please answer with "rejected", "no", or "nok" for the decision "No".

Please answer with "approved", "yes" or "ok" for the decision "Yes".

Best regards,  
Your TIM-System

-----({SYS.PROCESSINSTANCE\_ID})-----{Wait4}-----

## Dependencies

TIM version : this is available for configuration starting with [version 3.6.2](#)

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:timer:signalbymailreply>

Last update: **2021/07/01 09:52**

