Deadlines and Escalations

Introduction

Deadlines act as a timer which triggers an exception transition to another activity when a specified duration has elapsed. Deadlines can be set for each activity.

We set time limits on each activity, which will trigger an exception (in this case, a specific string variable). When a matching transition that has the condition set to 'exception' matches this string value, the transition will be invoked.

Caution

- Remember to turn on **Deadline Checker Interval** in General Settings so that Deadlines will be triggered.

- Do take into account of deadline tasks processing times. All deadline tasks **must** be completed before the deadline checker interval starts calculating again.
  This effect is more prevalent if there are many deadline tasks in queue and/or those task(s) requires heavy processing.
  Example:
  Assuming the deadline checker interval is set to 10 minutes & all current deadline tasks in queue would take 5 minutes to finish processing, it would take 5 minutes to complete processing and then only the deadline checking of 10 minutes will begin again after. From the perspective of the Joget app designer, it would seem that the deadline checker interval is irregular but this is actually the expected behavior.

- When an activity assignment is created, any associated deadline(s) is also created at that instance and ready to be executed upon its set deadline duration. Because of this, deadlines will also store the values of workflow variables at the time of creation of such activity.
  Hence, any modification to the workflow variables after the creation of the deadline(s) will **not** take into effect (e.g: changing workflow variable values in the Process Monitor), while the old values will.

- You can set multiple deadlines in an activity.
  As an example, you can include a first-level reminder (asynchronous), which is then followed by an escalation (synchronous), if the activity has not been completed within the time limit.

- Remember to match the exception names. If they do not match, nothing will happen.

Types of Deadlines

- Deadline execution can be **synchronous** or **asynchronous**.

- For synchronous execution, the current activity will no longer be active when the deadline is triggered. This is used in cases such as an approval escalation.

- For asynchronous execution, the next activity will be executed while the current activity is still waiting. This is used in cases such as sending reminders.

- Multiple deadlines are supported for each activity.
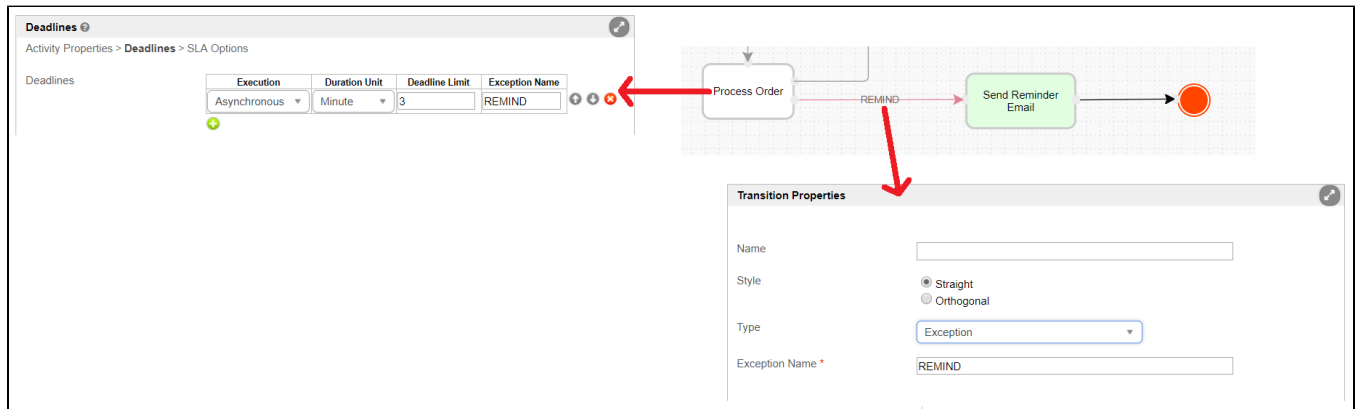
Setting Deadlines - A Step-by-Step Guideline

Activating Deadline Checker

- The activity deadline checker is disabled by default. To allow activity deadline checking by Joget, you will have to set the **Process Deadline Checker Interval** to a non-zero value. Set it to a suitable value, depending on your need.

- To enable deadline checking, go to the Admin Bar > General Settings > Timer Settings.

- Change the value for 'Process Deadline Checker Interval' to a non-zero value, i.e. 30 (seconds). The unit used is seconds.

Setting Deadlines in Activities

- Edit the activity properties you wish to set a deadline for, then click on the **Deadlines** tab. Add a new row and enter appropriate values for 'Execution', 'Duration Unit' and 'Deadline Limit'.

- Type the exception name in the **Exception name** field (e.g., TIMEOUT). Exception names are case sensitive. Make sure this matches the exception name given to the transition.

- Click on '**OK**' to save your deadline configuration. You can as many deadlines as you require.

Setting Transition as Exception Flow

- When a deadline is triggered, and where the exception name matches the one set for a transition, the flow will execute along that transition to the next element along the workflow.

- Add an Exception-type transition from the activity to the next (i.e., Send Reminder).

- Set the type of condition to 'Exception'.

- Enter an Exception name (e.g., TIMEOUT) in the Expression text area.

- Confirm and close the properties window for the transition.

Deadline In A Joget Cluster

The deadline checker is controlled by the database. When the deadline checker interval time is reached. Joget will pick up a batch of 10 from the running activity instances having deadlines, and run it until it goes through all the running activity instances. Then it will start the next interval. Joget will skip the activity if the deadline is already triggered. In a clustered environment, every cluster node will have its deadline checker and will perform the same deadline checking process. The first checker that reaches the activity deadline first will flag (update) the deadline in the database, then proceed with the deadline workflow. The second checkers will skip that activity and continue the checking process.

Related Documentation

Office Working Hour Deadline Plugin is an optional addition to your app where SLAs and deadlines are implemented. This plugin will adjust the SLA and deadlines due date calculations based on working hours, weekends and company holidays.

Download Demo App

Download the tutorial app on Process Deadline from Joget Marketplace to learn more.