RAD Service

RADS

SLA & Deadlines

https://github.com/radservice/rad-community/fork

Last Revised on Oct 2021

Prerequisites

1. Good understanding on how to design a Workflow Process with SLA & Deadlines

Content

RAD Service

Chapter 1

Service Level Agreement (SLA)

Service Level Agreement (SLA) Limit

- Why set limits?
  - By setting limits to workflow activities, you are able to define appropriate service levels for your processes.
  - Participants in the workflow can be made aware of adherence to these service levels.
  - You can generate reports to determine the efficiency of your processes e.g. identify bottlenecks, etc.

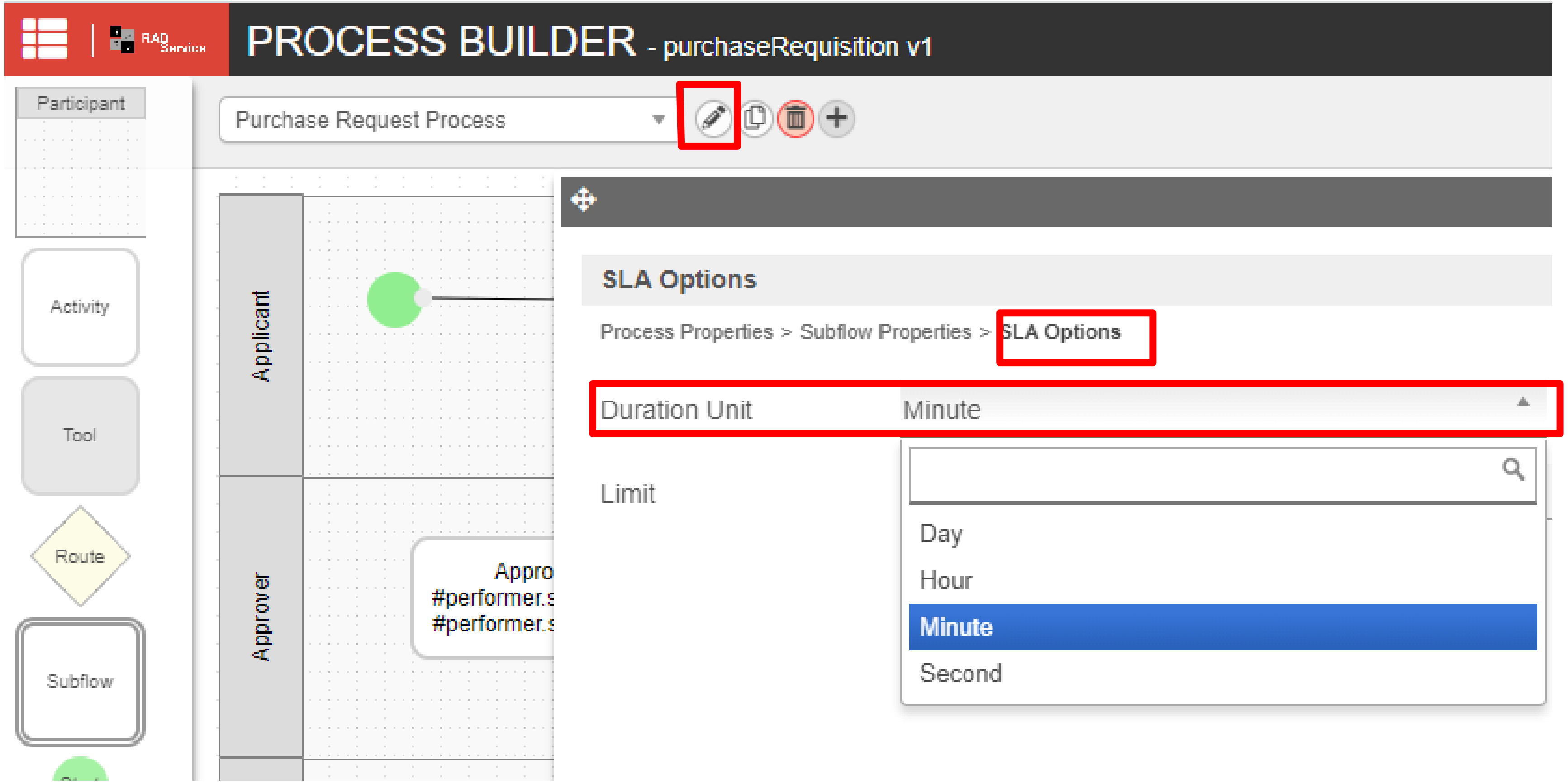
RAD Service

How to Set SLA Limit?

- Define process-level **duration unit**
  (Duration unit will be **shared** among activities in the process)
- Set **SLA Limit** on targeted activity or process
- **SLA** can be affected by the use of Deadline plugins.

Define Duration Unit

- Edit process's properties to set the duration unit.
- Set to **minute.**



SLA Options

Process Properties > Subflow Properties > SLA Options

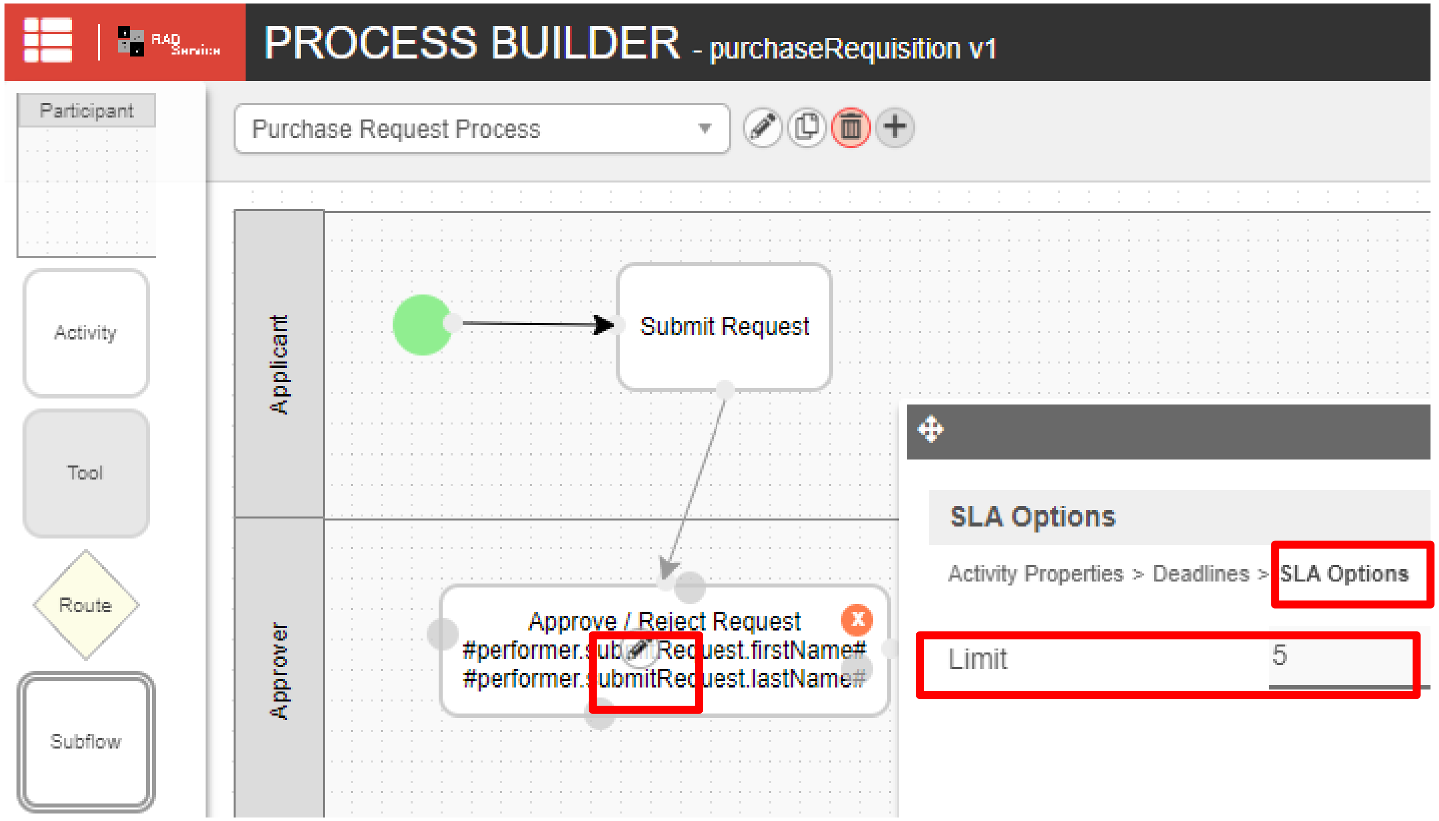
Duration Unit: Minute

Limit

Day
Hour
Minute
Second

Set SLA Limit on Activity

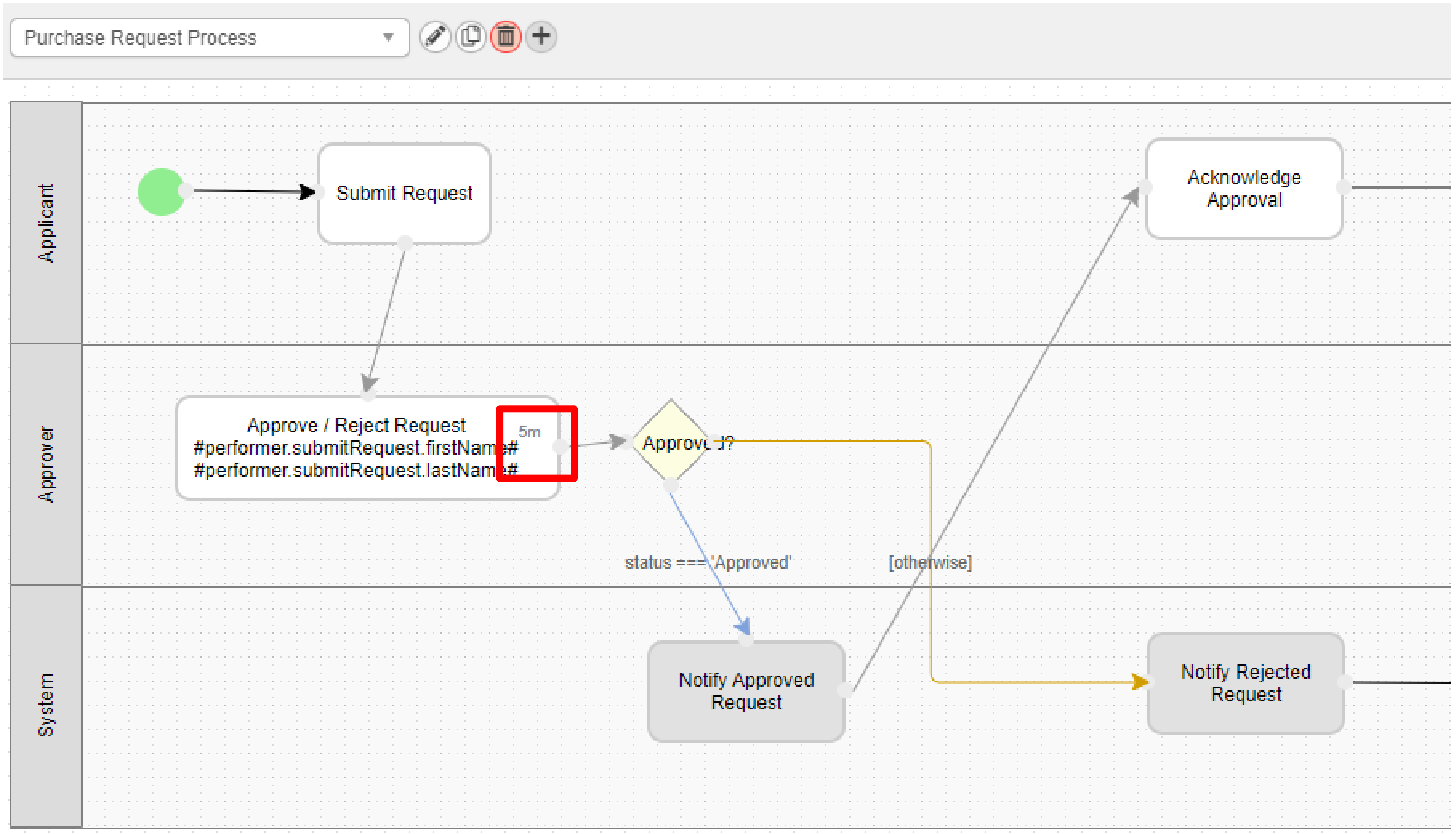
- Set “**5**” in **Approve / Reject Request** activity. (Duration unit was set earlier to **minute**)


PROCESS BUILDER - purchaseRequisition v1
Participant
Purchase Request Process
Activity
Tool
Route
Subflow
Applicant
Approver
Submit Request
Approve / Reject Request
#performer.submitRequest.firstName#
#performer.submitRequest.lastName#
SLA Options
Activity Properties > Deadlines > SLA Options

Limit 5

SLA Limit Indicator

- SLA will be reflected in process diagram itself.



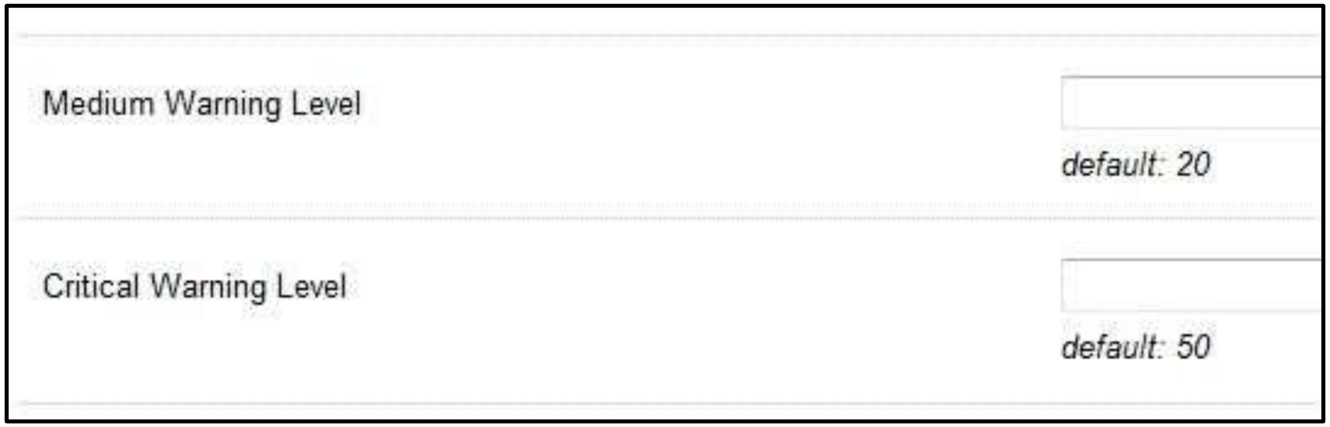
Service Level Monitor

- For activities that have defined **limits**, task assignments will have **due dates**.
- When processes that have limits are deployed, you will notice that the **Service Level Monitor** column in the Inbox will display a **colour coded square** 🟥 🟨 🟩 .
- The colour of the Service Level Monitor will change from **GREEN** to **YELLOW** as the due date approaches. Once the due date is reached, the colour will be **RED**.

RAD Service

Defining SLA Indicator

- Medium Warning Level
  - Color: Yellow
  - Default Value: 20% of elapsed time.
- Critical Warning Level
  - Color: Red
  - Default Value: 50% of elapsed time.
- Can be changed in the System Settings. (System wide effect)

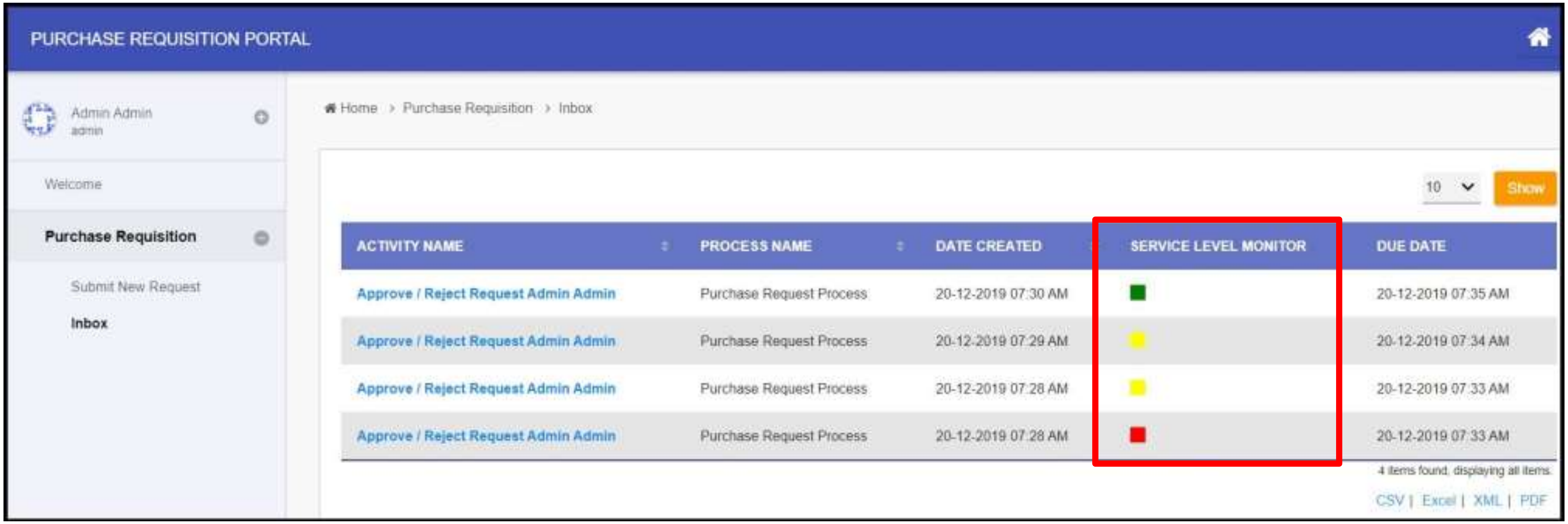


Medium Warning Level	default: 20
Critical Warning Level	default: 50

RAD Service

Service Level Monitor

- SLA Indicator will be seen in the task Inbox.



PURCHASE REQUISITION PORTAL

Admin Admin
admin

Welcome

Purchase Requisition

Submit New Request

Inbox

Home › Purchase Requisition › Inbox

10 Show

ACTIVITY NAME	PROCESS NAME	DATE CREATED	SERVICE LEVEL MONITOR	DUE DATE
Approve / Reject Request Admin Admin	Purchase Request Process	20-12-2019 07:30 AM	🟩	20-12-2019 07:35 AM
Approve / Reject Request Admin Admin	Purchase Request Process	20-12-2019 07:29 AM	🟨	20-12-2019 07:34 AM
Approve / Reject Request Admin Admin	Purchase Request Process	20-12-2019 07:28 AM	🟨	20-12-2019 07:33 AM
Approve / Reject Request Admin Admin	Purchase Request Process	20-12-2019 07:28 AM	🟥	20-12-2019 07:33 AM

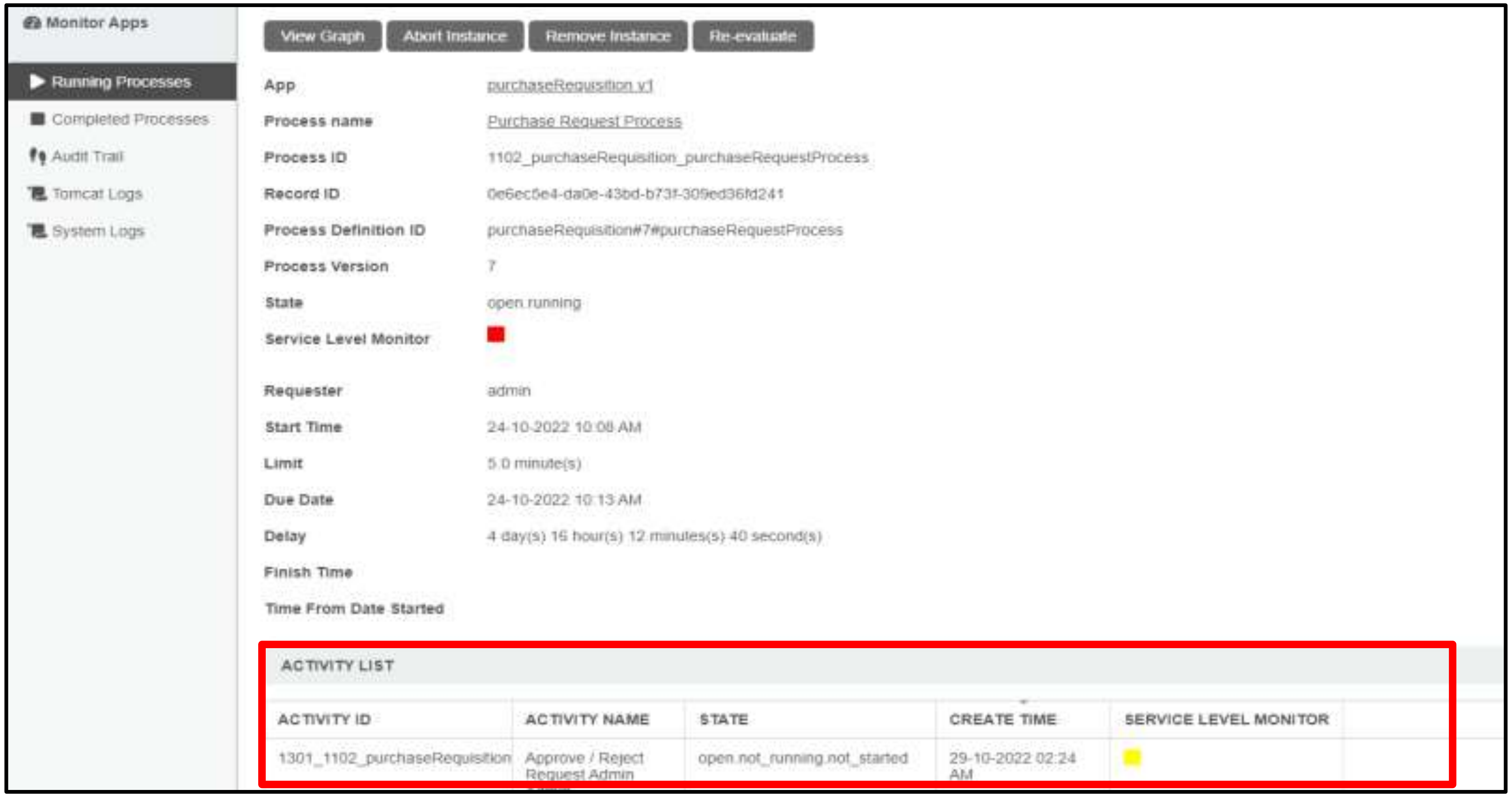
4 items found, displaying all items.

CSV | Excel | XML | PDF

RAD Service

Process Monitoring

- SLA Indicator is also available in the Process Monitoring module for the administrator.



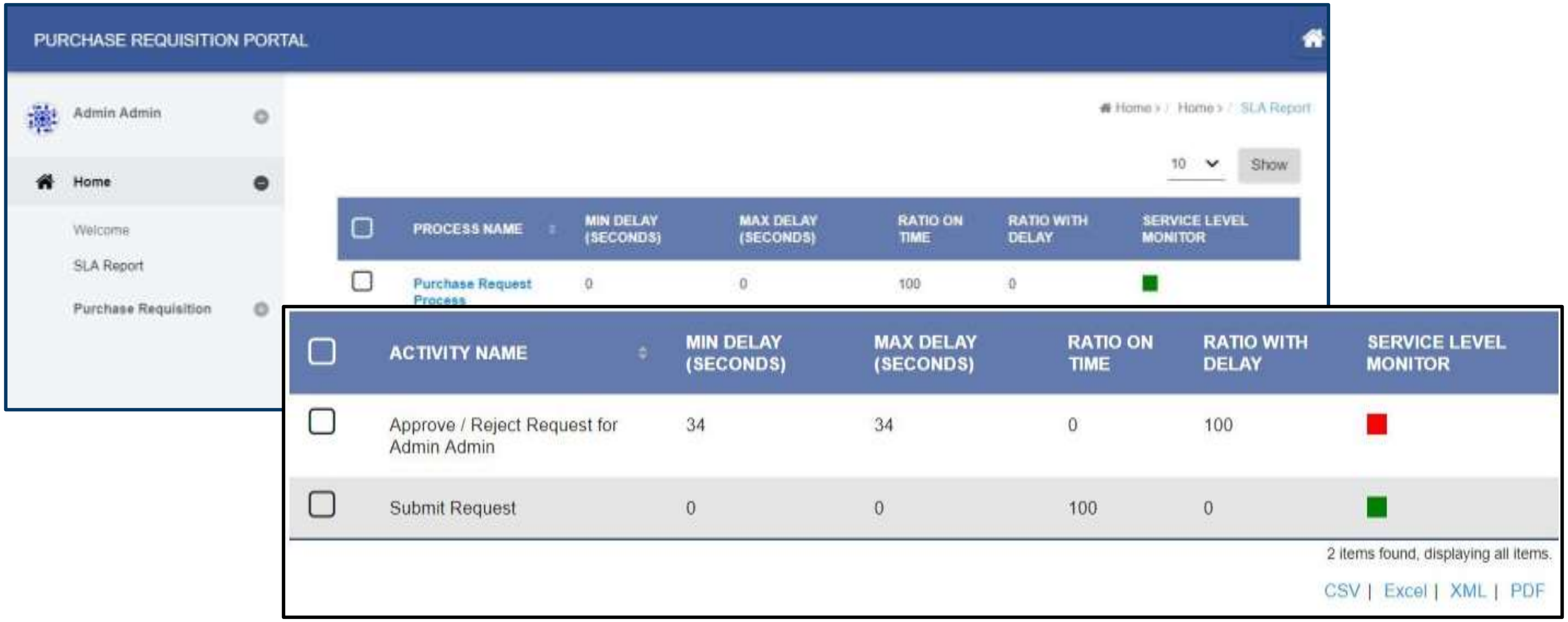
App	purchaseRequisition v1
Process name	Purchase Request Process
Process ID	1102_purchaseRequisition_purchaseRequestProcess
Record ID	0e6ec5e4-da0e-43bd-b73f-309ed36fd241
Process Definition ID	purchaseRequisition#7#purchaseRequestProcess
Process Version	7
State	open.running
Service Level Monitor	🟥
Requester	admin
Start Time	24-10-2022 10:08 AM
Limit	5.0 minute(s)
Due Date	24-10-2022 10:13 AM
Delay	4 day(s) 16 hour(s) 12 minutes(s) 40 second(s)
Finish Time	
Time From Date Started	

ACTIVITY LIST

ACTIVITY ID	ACTIVITY NAME	STATE	CREATE TIME	SERVICE LEVEL MONITOR
1301_1102_purchaseRequisition	Approve / Reject Request Admin	open.not_running.not_started	29-10-2022 02:24 AM	🟨

SLA Report on Userview

- SLA Report can also be made available for end user (e.g. Manager) to inspect.



PURCHASE REQUISITION PORTAL

☐	PROCESS NAME	MIN DELAY (SECONDS)	MAX DELAY (SECONDS)	RATIO ON TIME	RATIO WITH DELAY	SERVICE LEVEL MONITOR
☐	Purchase Request Process	0	0	100	0	🟩

☐	ACTIVITY NAME	MIN DELAY (SECONDS)	MAX DELAY (SECONDS)	RATIO ON TIME	RATIO WITH DELAY	SERVICE LEVEL MONITOR
☐	Approve / Reject Request for Admin Admin	34	34	0	100	🟥
☐	Submit Request	0	0	100	0	🟩

2 items found, displaying all items.

CSV | Excel | XML | PDF

- Reference: https://docs.rads.purwana.net/SLA+Report

Chapter Review

- Set SLA limit to workflow activity, which enables the implementation of service level monitoring.

Exercise

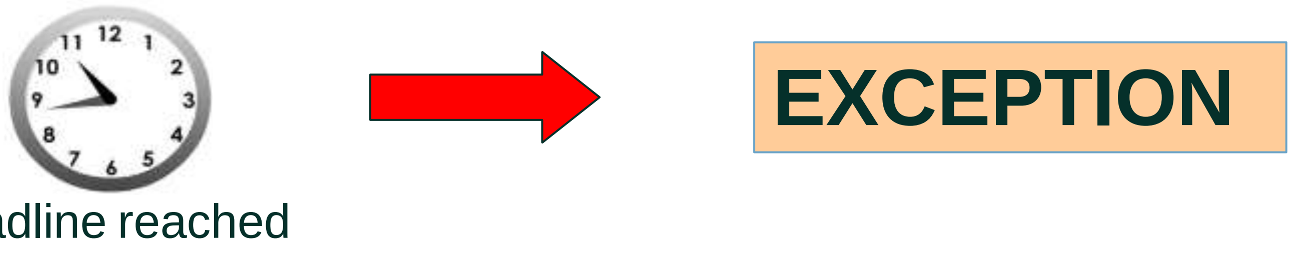
- Tweak the SLA settings to only turn to YELLOW when half of the remaining time elapsed and…
- Change to RED when all the remaining time runs out.

Chapter 2

Deadlines

Deadlines and Exceptions

- For each activity, Deadline(s) can be set.
- Deadlines act as a timer which triggers an Exception transition to another activity when a specified duration has elapsed
- Deadlines that has reached its elapsed duration will be queued and then be picked up by the **Deadline Checker**.



Deadlines

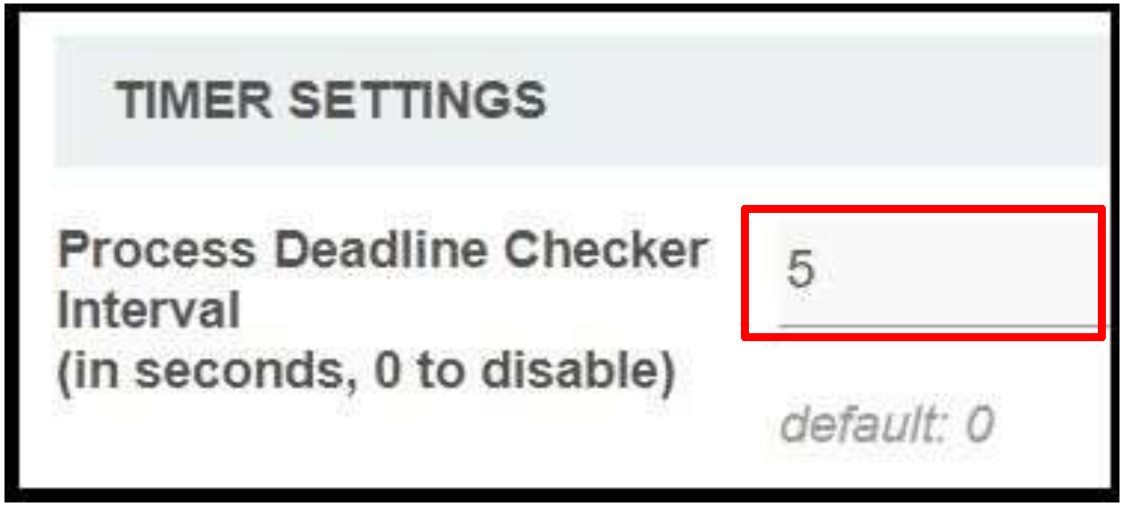
- Deadline execution can be **synchronous** or **asynchronous**.
- For **synchronous** execution, the current activity will no longer be active when the deadline is triggered.
  - Used in cases such as approval escalation.
- For **asynchronous** execution, the next activity will be executed while the current activity is still waiting.
  - Used in cases such as sending reminders.
- Multiple deadlines are supported for each activity.

Deadline Checker

- The Deadline Checker kicks in at **specified intervals** when enabled.
- **Deadlines due** at the time will be **picked up** and processed in batches of 10 by the Deadline Checker until finish.
- The deadline checker will resume counting only when it completes processing all pending deadline tasks.
- Set an appropriate interval that suits your environment.

Activating Deadline Checker

- The **Process Deadline Checker** MUST be enabled under **System Settings** > **General Settings** for deadlines to work.
- This will enable periodic checks on activities' pending deadline tasks as defined by the **checker interval**.


TIMER SETTINGS

Process Deadline Checker Interval
(in seconds, 0 to disable): 5

default: 0

Deadline Checker Discussion

- What will the timeline be like for a Deadline set at 1 minute and Deadline Checker at 1 hour?

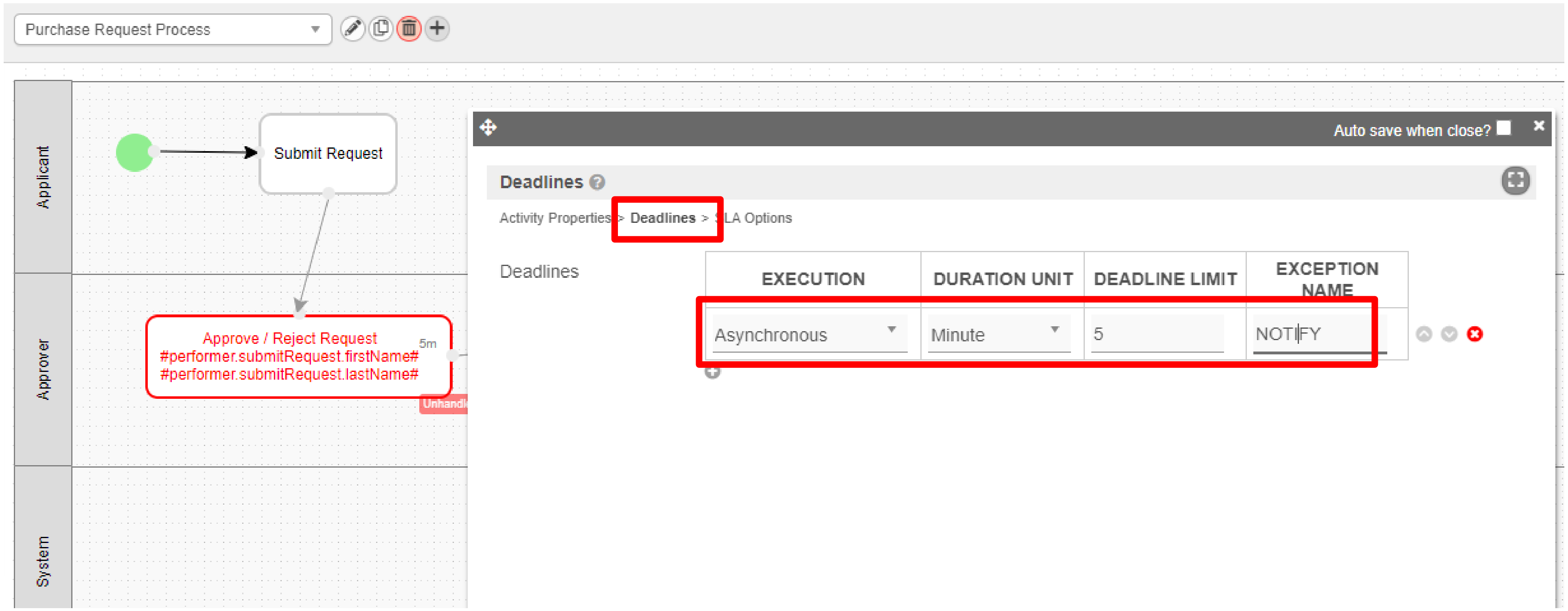
  (Think of the mailman analogy)

Setting Deadlines and Exceptions

- **Deadlines** are set at **activities**.
- **Exceptions** are set at **transitions**.
- Exceptions transition to another **activity** or **tool**.
- These are required for the deadline feature to work.

Add a New Deadline

- Add a new Deadline to the **Approve / Reject Request** activity.
- Add **Asynchronous** deadline, set it to 5 minutes.
- Set an exception name. It **must** be unique.

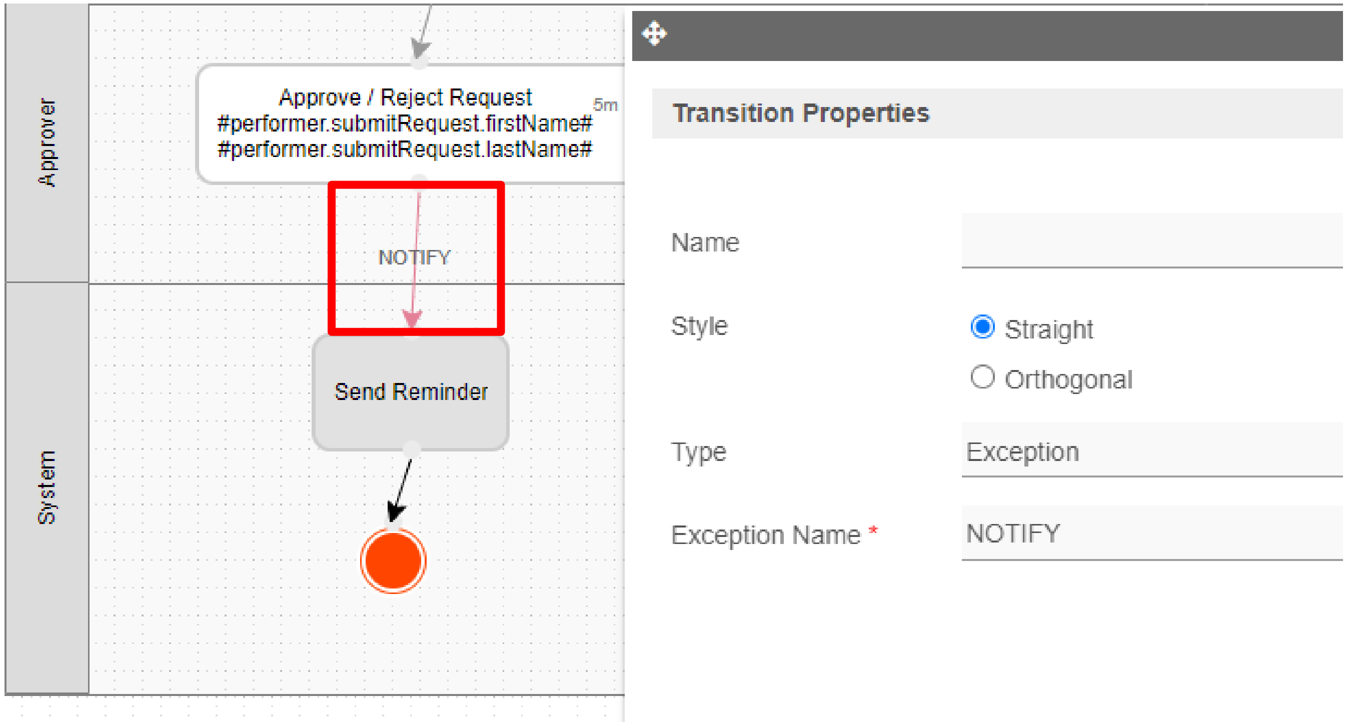


Deadlines

EXECUTION	DURATION UNIT	DEADLINE LIMIT	EXCEPTION NAME
Asynchronous	Minute	5	NOTIFY

Add a New Deadline

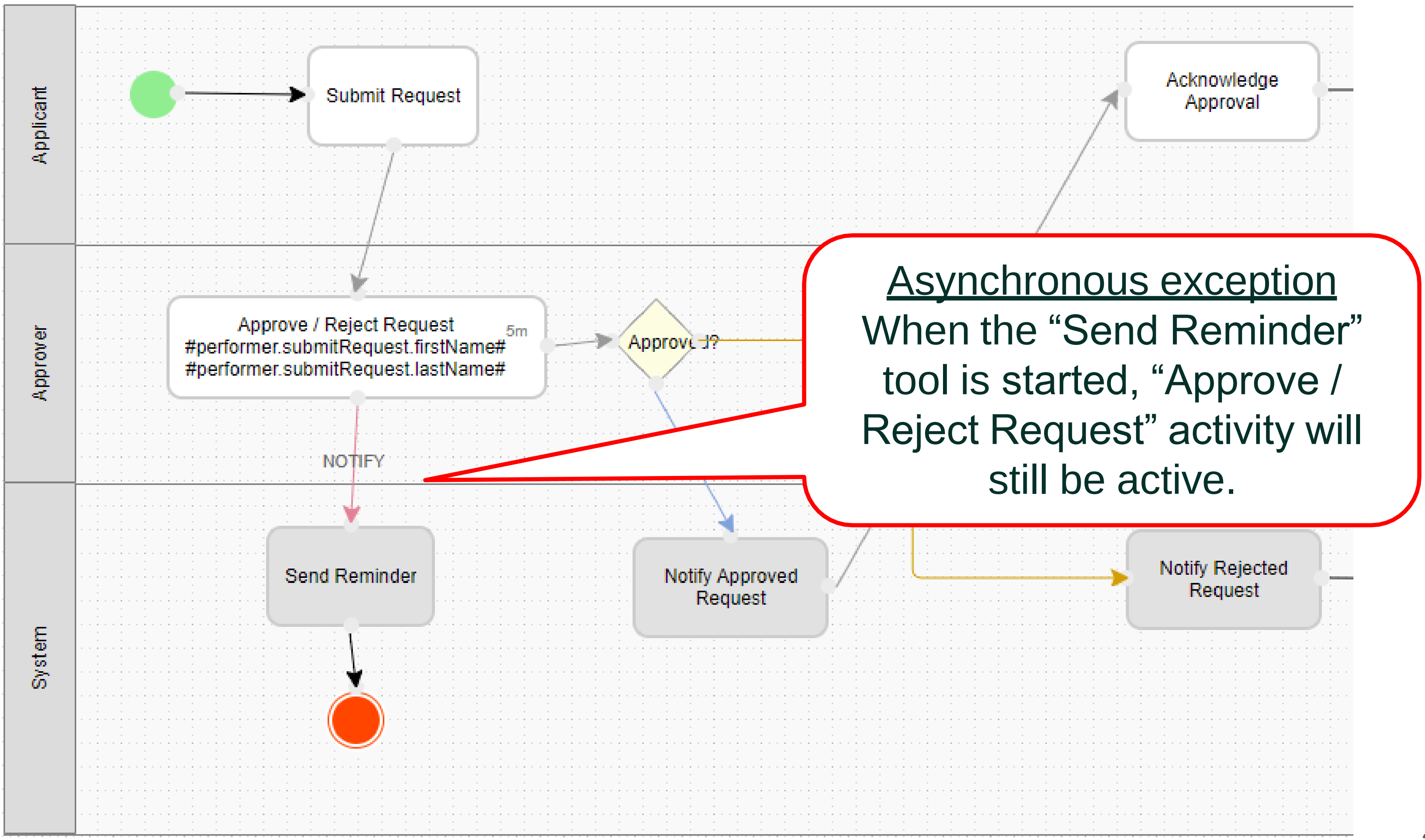
- Add a new Tool, set it to **Send Reminder**.
- Link them up and set transition type to **Exception**. Set the exception name as the one declared earlier.


Approver
System
Approve / Reject Request
#performer.submitRequest.firstName#
#performer.submitRequest.lastName#
5m
NOTIFY
Send Reminder

Transition Properties

Name	
Style	[✓ Straight] Orthogonal
Type	Exception
Exception Name *	NOTIFY

Setting Deadlines and Exceptions



<u>Asynchronous exception</u>
When the “Send Reminder” tool is started, “Approve / Reject Request” activity will still be active.

Process Monitoring

- Study on how deadline would affect your process instance in Process Monitoring.
- Differentiate types of “state” of activity/process.
  – closed.completed
  – closed.terminated
  – close.aborted
  – open.not_running.not_started

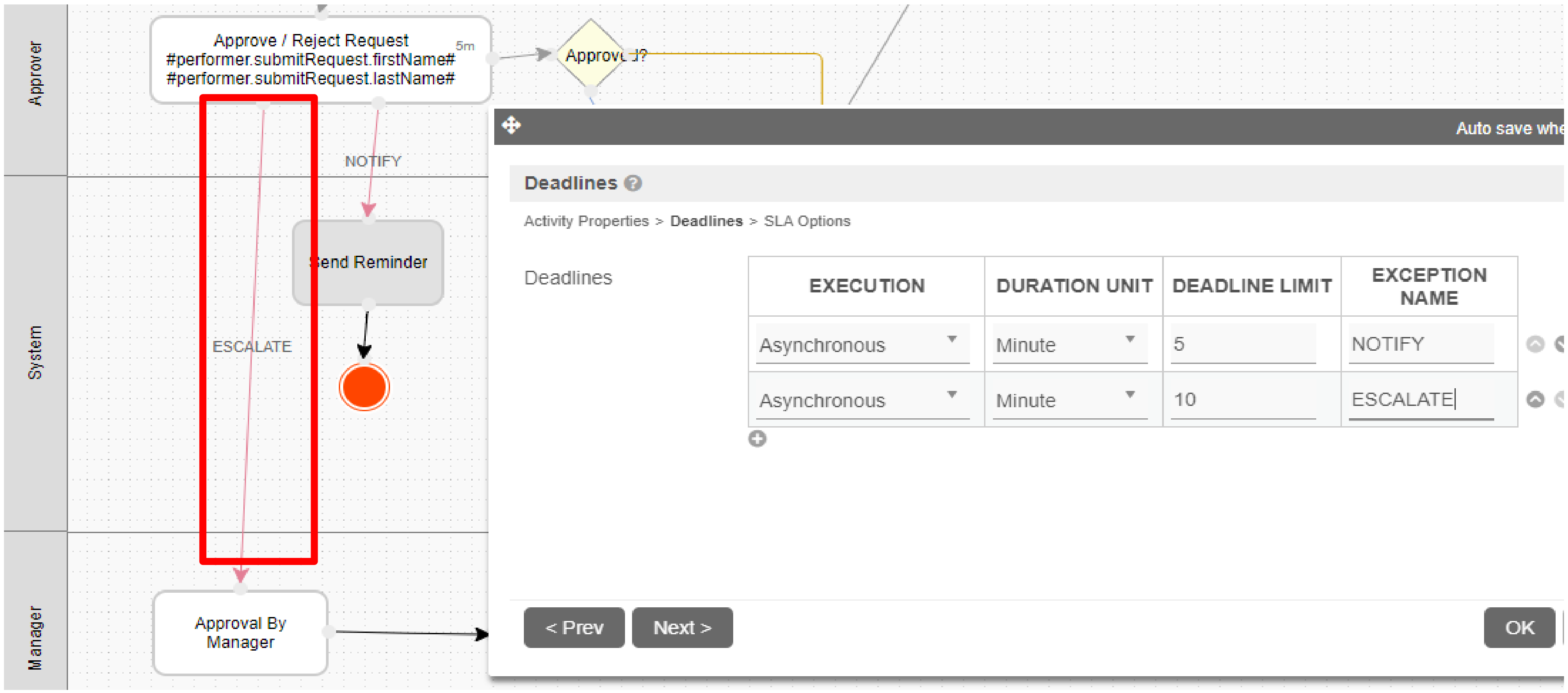
Exercise 1

- Verify that the deadline works as designed by mapping the newly created Tool to a Email Tool plugin.
- Configure the Email Tool accordingly.
- Test if the Email Tool is triggered by the Deadline by starting up a new process instance.
- Optionally, you may use Bean Shell Tool to output to the server log to test this too.

Exercise 2

- Add a second level approval for when the first level approver does not respond in certain period, the second approver will take over.
- Choose the appropriate time frame and type of deadline.
- Configure the new participant's mapping accordingly after deployment of the new process flow.

Exercise 2



Deadlines

Activity Properties > **Deadlines** > SLA Options

Deadlines

EXECUTION	DURATION UNIT	DEADLINE LIMIT	EXCEPTION NAME
Asynchronous	Minute	5	NOTIFY
Asynchronous	Minute	10	ESCALATE

Good To Know

- One activity may contains one or more Deadlines.
- Each Deadline has its own Duration Unit, unlike SLA.

Good To Know

- It is also possible to set use a **Workflow Variable** as the Deadline limit.

  (Remember to declare the Workflow Variable and set the value according to the date format defined)

RAD Service

Important Note

- SLA will **NOT** manipulate the flow of your Workflow Process, Deadline **will**.
- Deadlines will <u>highly unlikely</u> get triggered on the dot as it depends on the Deadline Checker Interval cycle & deadline tasks processing times.
- Deadline interval will begin again only after all deadline tasks in queue has completed processing.
- Do NOT set deadlines on Tool, it's only meant for Activities.

Chapter Review

1. Set deadline and design exception handling.
2. Understand the difference between Synchronous and Asynchronous deadline.
3. Understand the overall deadline behavior

Reference:

https://docs.rads.purwana.net/Deadlines+and+Escalations

RAD Service

Chapter 3

Introduction to Deadline Plugin

Deadline Plugin

- **Deadline plugin** will influence the calculation of **SLA** and **Deadlines** in a process flow.

Office Working Hour Deadline Plugin

- **Office Working Hour Deadline Plugin** is an essential addition to the working environment where **SLAs** and **deadlines** are implemented.
- This plugin will intercept and override how calculations are made by RADS when calculating due dates for SLAs and deadlines imposed on a process.
- The Office Working Hour Deadline plugin takes the following into account during calculations:
  - Holidays
  - Working Hours
  - Working Days

Exercise

Consider a synchronous deadline where:

→ Activity started at	: Friday 5.50pm
→ Deadline trigger set to	: 30 minutes
→ Last deadline checked	: 5.55pm
→ Deadline interval	: every 1 hour
→ Office Hours	: Weekdays 9am - 6pm

Ignoring the deadline task processing time, when does the synchronous deadline trigger?

Exercise - Optional

- Configure the Office Working Hour Deadline plugin into your existing App.
- Observe the changes to the due dates on SLA and Deadline.

Chapter Review

- Able to understand the impact of Deadline plugins to Deadline and SLAs calculations.

Module Review

1. Set SLA limit to workflow activity, which enables the implementation of service level monitoring.
2. Set deadline and design exception handling.
3. Understand the difference between Synchronous and Asynchronous deadline.
4. Understand the overall deadline behavior
5. Understand the Deadline Plugin and its purposes.

RAD Service

Stay Connected With RADS

- rads.purwana.net
- https://github.com/radservice/rad-community