

# Configure Case Automation Rules

Last Modified on 09/11/2025 9:58 am EDT

V10.8

## Overview

To enhance the end user experience, Cora Orchestration supports case automation such that the system automatically performs case processing actions with minimal manual intervention.

For example:

- The system automatically updates the case data and case status after the system receives a response from the ERP.
- The system automatically sends a reminder email to a collaborator to get response for case resolution.

As a workflow developer, you can configure rules to automate case processing. These rules automatically trigger workflows according to the implementation needs and cover multiple use cases.

## Steps to configure rules

Step	Location	Description
Create a Config Set	Administration site>Global Settings> ConfigSets.	Create a config set with YAML files that define rules.
Configure Rule-Based Workflow Starter Job	Administration site>Global Settings>Jobs Management.	Add a job that triggers workflow and automates several required actions.

## Create a Config Set

1. In the Administration site, go to **Administration>Global Settings>ConfigSets**.
2. Copy the *Extraction\_Configuration*, to create a new config set for a workspace.
3. Edit the new config set to make changes.  
The window to edit configuration properties opens in a new tab in your browser window.  
Each YAML file is a configuration file for different steps.
4. Click a YAML file to open and update the parameters per requirement.

## Configure Rule-Based Workflow Starter Job

In the Administration site, go to **Administration>Global Settings>Jobs Management**, and click **Add Job**.

1. Select Rule-Based Workflow Starter on Create New Job screen, and click **Create**.
2. On the **Job** tab of job definition screen, define the following:
  - **Name:** Enter a meaningful name for the data collector job.
  - **Job is enabled:** Select this option only after you complete the workflow, or if you want to run the job for testing purposes.
3. Click **Next**.
4. On the **Command** tab, set the following:

- Solution name: The JSON name of the solution for which the job will process rule set.
  - Document Meta data name: The JSON name of the document metadata as appears in the YAML header.
5. Click **Next**.
  6. On the **Advanced Options** tab, set the following:
    - **Job Execution Schedule**: Set the job to execute once a day.
    - Select the "If a job is currently running, do not start the next job..." check box, to avoid congestion.
    - If not defined, job will run with default options.
  7. Click **Finish**.

## Example

```
kind: ruleset
metadata:
  name: config-set/v1/my-ruleset-name1
spec:
  nodes:

  - if: wf.variables["caseState"], = "awaiting response" And dateDiff("hour", wf.variables["statetime"], now()) > 2
    then:
      workflowSpaceId: '0BA83A46-548F-4DAB-BA5A-00037C7019E8' # Trigger the active workflow in this space
      startAsStandalone: false # true starts the workflow as a standalone instance, false starts the instance in scope of
the case Id
      ruleId: 123 # unique and unchangeable integer
      ruleName: 'Reminder 2H' # Rule name (max 100 chars)
      ruleSuppressString: ': toString(wf.variables["statetime"])' # nvarchar(max)

  parameters:
    - typeCode: Int32
      name: 'hoursPending'
      value: ': ToInt32(dateDiff("hour", wf.variables["statetime"], now()))'
    - typeCode: string
      name: 'myComment'
      value: 'Reminder 1'
```