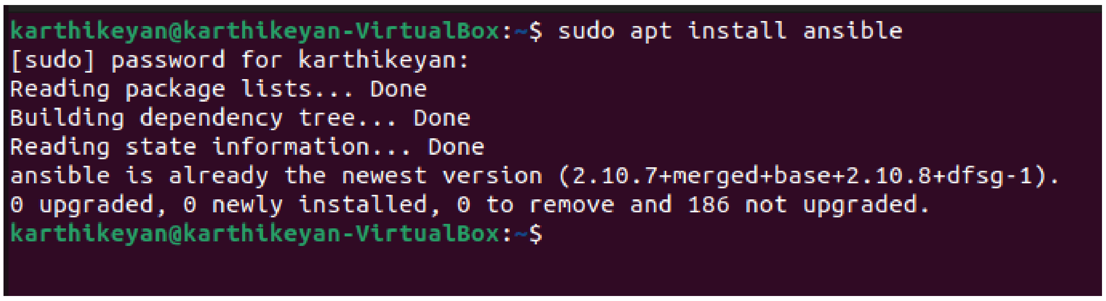
Ex No 8 Date:	**Install Ansible and configure ansible roles and to write playbooks**

Aim:

To Install Ansible and configure ansible roles and to write playbooks.

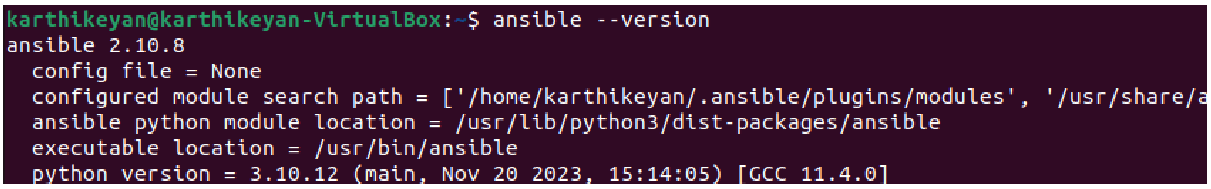
Procedure:

1. Install ansible using sudo apt install ansible command



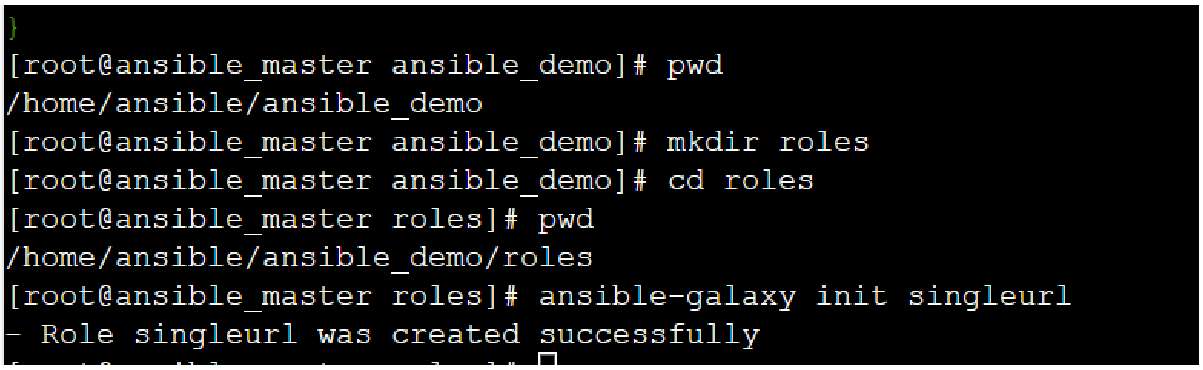
```
karthikeyan@karthikeyan-VirtualBox:~$ sudo apt install ansible
[sudo] password for karthikeyan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (2.10.7+merged+base+2.10.8+dfsg-1).
0 upgraded, 0 newly installed, 0 to remove and 186 not upgraded.
karthikeyan@karthikeyan-VirtualBox:~$
```

2. Check ansible version using ansible –version command



```
karthikeyan@karthikeyan-VirtualBox:~$ ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/home/karthikeyan/.ansible/plugins/modules', '/usr/share/a
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
```

3. Create roles directory
   - Mkdir roles & cd roles
4. Creation of role
   - ansible-galaxy init singleurl



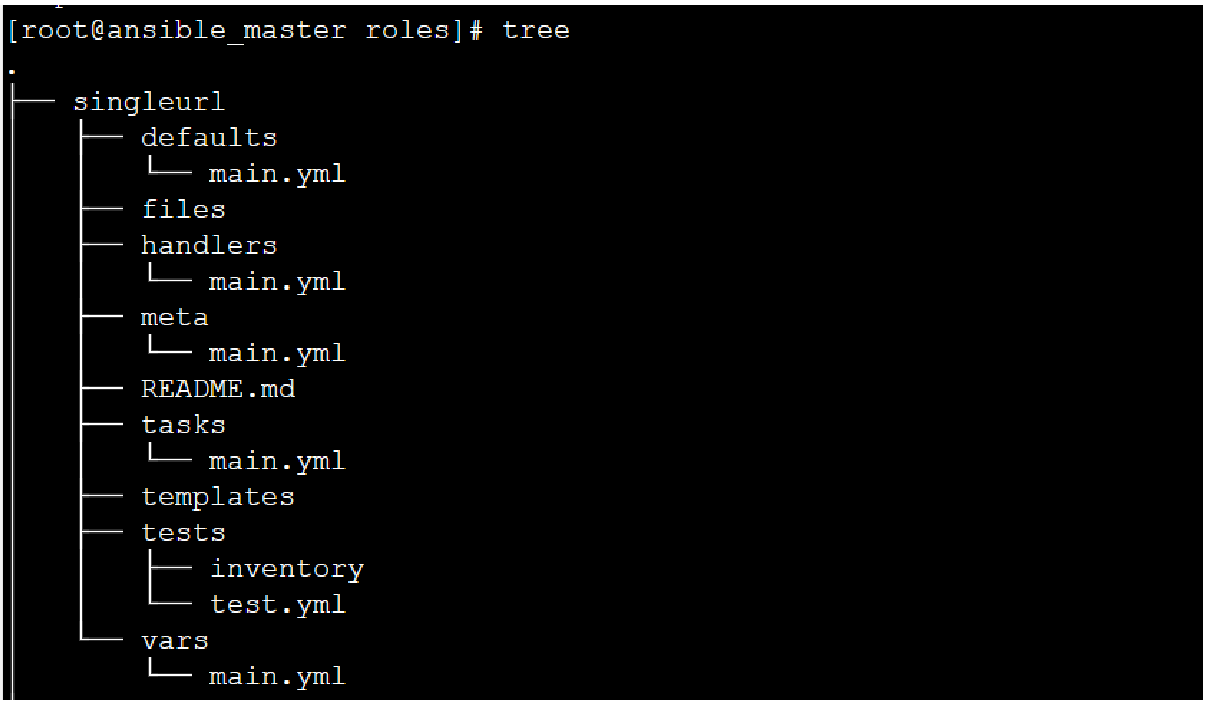
```
}
[root@ansible_master ansible_demo]# pwd
/home/ansible/ansible_demo
[root@ansible_master ansible_demo]# mkdir roles
[root@ansible_master ansible_demo]# cd roles
[root@ansible_master roles]# pwd
/home/ansible/ansible_demo/roles
[root@ansible_master roles]# ansible-galaxy init singleurl
- Role singleurl was created successfully
```

5. Declare your roles in your environment related ansible.clg file.
   - Gedit ansible.clg

```
[defaults]
inventory=hosts
```

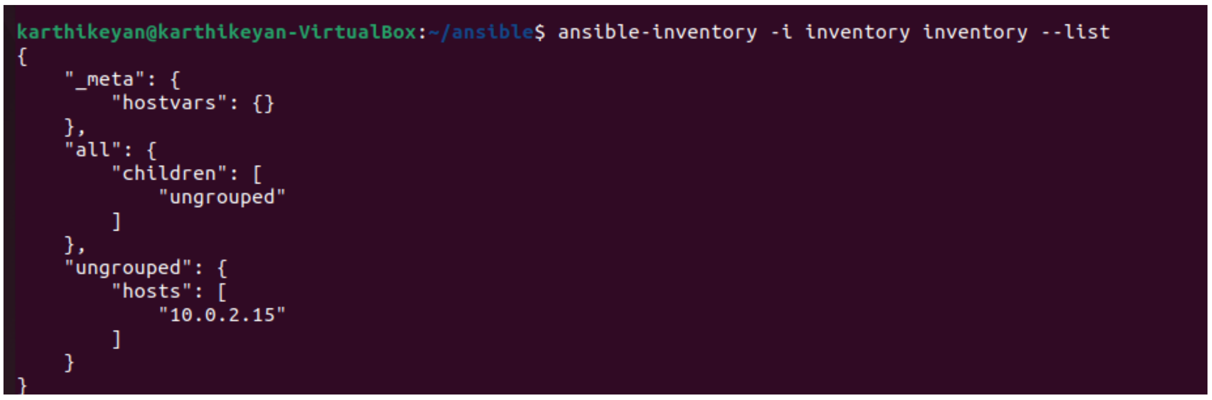
```
remote_user=ansible
timeout=3000
roles_path=/home/ansible/dev/roles
[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False
```

6. We can use the tree command to check the role created.



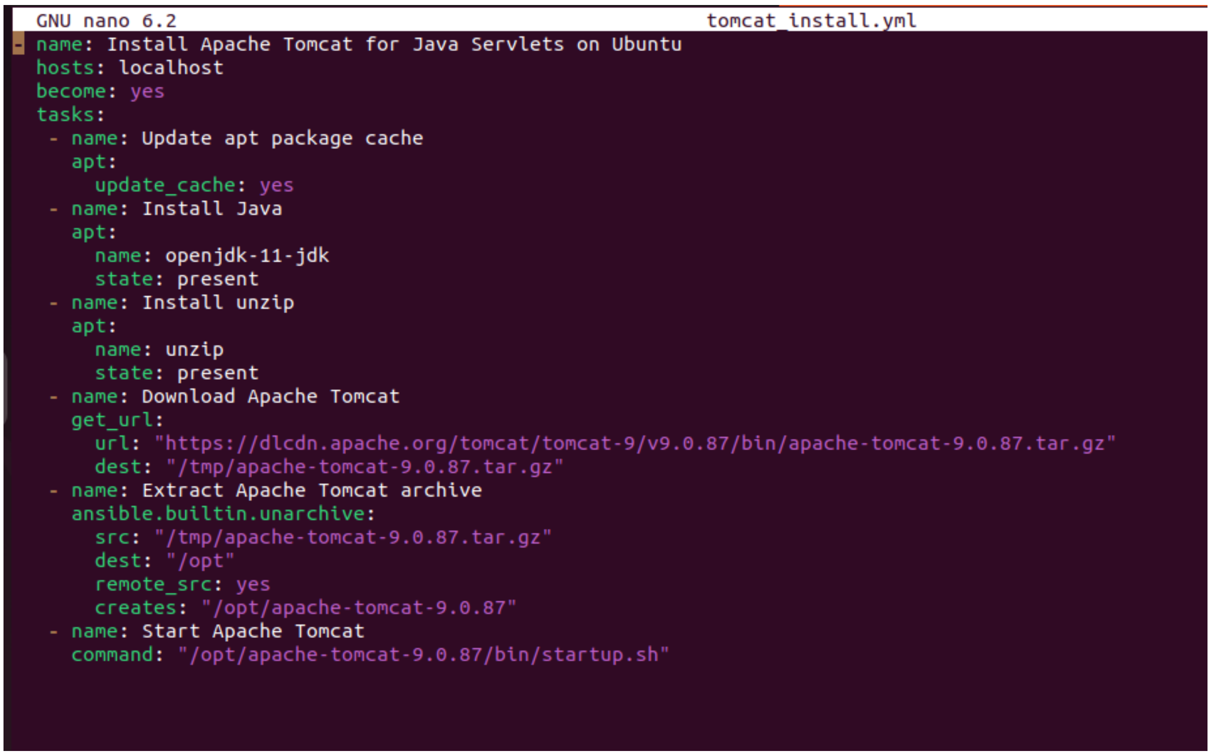
```
[root@ansible_master roles]# tree
.
├── singleurl
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
```

7. Úse the ansible-inventory -i inventory inventory –list command to view the inventory file after adding the IP address.



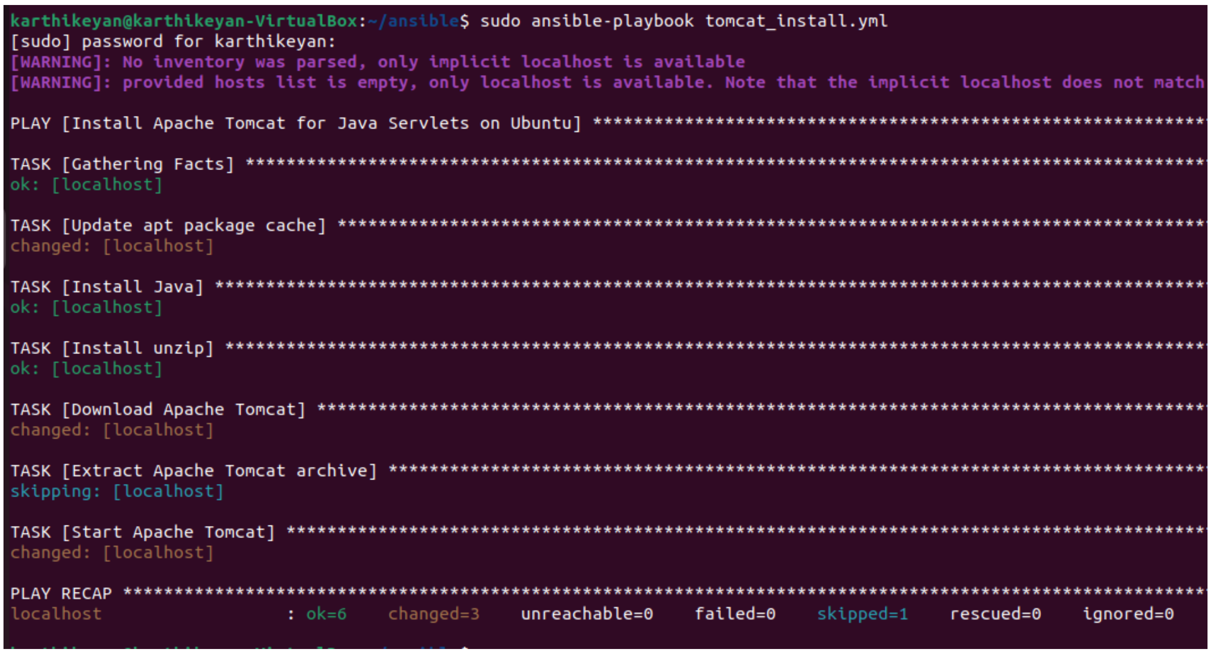
```
karthikeyan@karthikeyan-VirtualBox:~/ansible$ ansible-inventory -i inventory inventory --list
{
    "_meta": {
        "hostvars": {}
    },
    "all": {
        "children": [
            "ungrouped"
        ]
    },
    "ungrouped": {
        "hosts": [
            "10.0.2.15"
        ]
    }
}
```

8. Create a playbook and add the steps to install tomcat server in it.



```
  GNU nano 6.2                                              tomcat_install.yml
- name: Install Apache Tomcat for Java Servlets on Ubuntu
  hosts: localhost
  become: yes
  tasks:
    - name: Update apt package cache
      apt:
        update_cache: yes
    - name: Install Java
      apt:
        name: openjdk-11-jdk
        state: present
    - name: Install unzip
      apt:
        name: unzip
        state: present
    - name: Download Apache Tomcat
      get_url:
        url: "https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.87/bin/apache-tomcat-9.0.87.tar.gz"
        dest: "/tmp/apache-tomcat-9.0.87.tar.gz"
    - name: Extract Apache Tomcat archive
      ansible.builtin.unarchive:
        src: "/tmp/apache-tomcat-9.0.87.tar.gz"
        dest: "/opt"
        remote_src: yes
        creates: "/opt/apache-tomcat-9.0.87"
    - name: Start Apache Tomcat
      command: "/opt/apache-tomcat-9.0.87/bin/startup.sh"
```

9. Run the file using ansible-playbook tomcat_install.yml



```
karthikeyan@karthikeyan-VirtualBox:~/ansible$ sudo ansible-playbook tomcat_install.yml
[sudo] password for karthikeyan:
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match

PLAY [Install Apache Tomcat for Java Servlets on Ubuntu] ***********************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [localhost]

TASK [Update apt package cache] ************************************************************************************
changed: [localhost]

TASK [Install Java] ************************************************************************************************
ok: [localhost]

TASK [Install unzip] ***********************************************************************************************
ok: [localhost]

TASK [Download Apache Tomcat] **************************************************************************************
changed: [localhost]

TASK [Extract Apache Tomcat archive] *******************************************************************************
skipping: [localhost]

TASK [Start Apache Tomcat] *****************************************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************************
localhost                  : ok=6    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Result:

Thus ansible playbook was created and apache tomcat was installed using ansible tool

Ex.no:	Run regression tests using Maven Build pipeline in Azure
Date:	

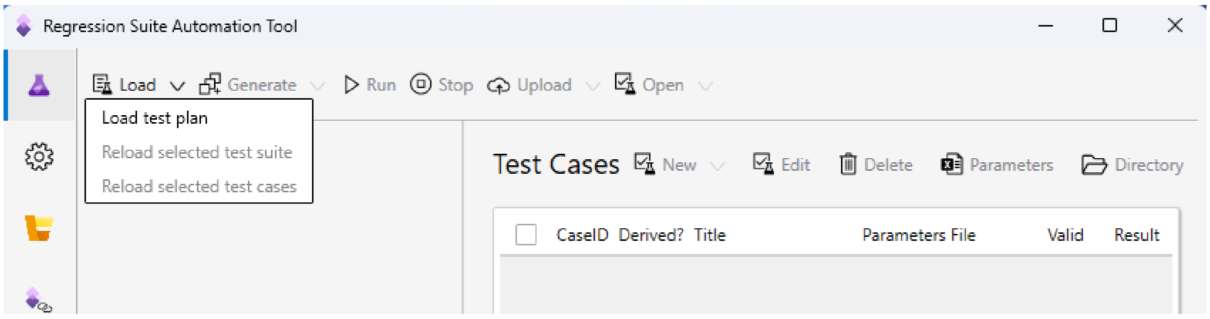
Aim:

To Run regression tests using Maven build pipeline in Azure

Procedure:

Load test cases and create automation files

In RSAT, select the **Test Plans** tab and then select **Load** to download test cases and test case automation files.



Test cases are organized by test suites under a common test plan

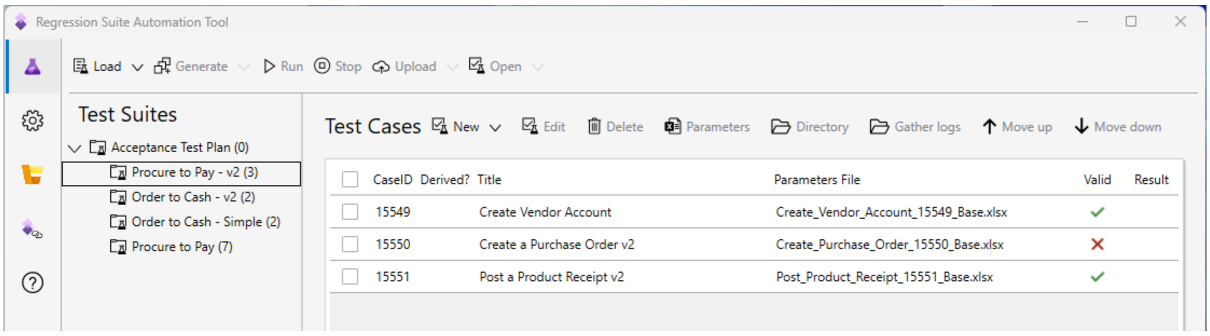
A test case requires the following attachments for successful execution:

- A **recording file**: It defines the steps of your test case.
- **Test automation files** consisting of **a test parameter file**

Select the **Generate** button to generates test automation files in your working directory. If the test case doesn't already have an Excel test parameter file, it's created and appears in the grid under **Parameters File**.

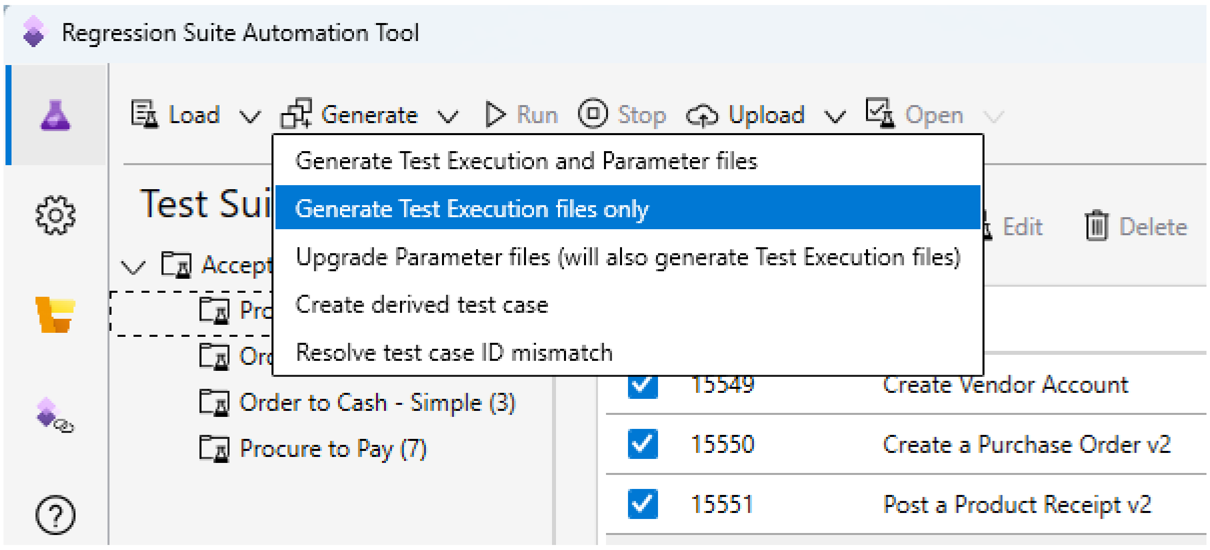
To generate only **test execution files**, without affecting your parameter files, select **Generate > Generate Test Execution files only**.

Selecting **Generate > Generate Test Execution and Parameter files** generates both test automation files and a new Excel parameter file in your working directory.



CaseID	Derived?	Title	Parameters File	Valid	Result
15549		Create Vendor Account	Create_Vendor_Account_15549_Base.xlsx	✓	
15550		Create a Purchase Order v2	Create_Purchase_Order_15550_Base.xlsx	✗	
15551		Post a Product Receipt v2	Post_Product_Receipt_15551_Base.xlsx	✓	

You must generate test execution files when you install a new version of the tool, and when you modify or load a new version of the recording file.

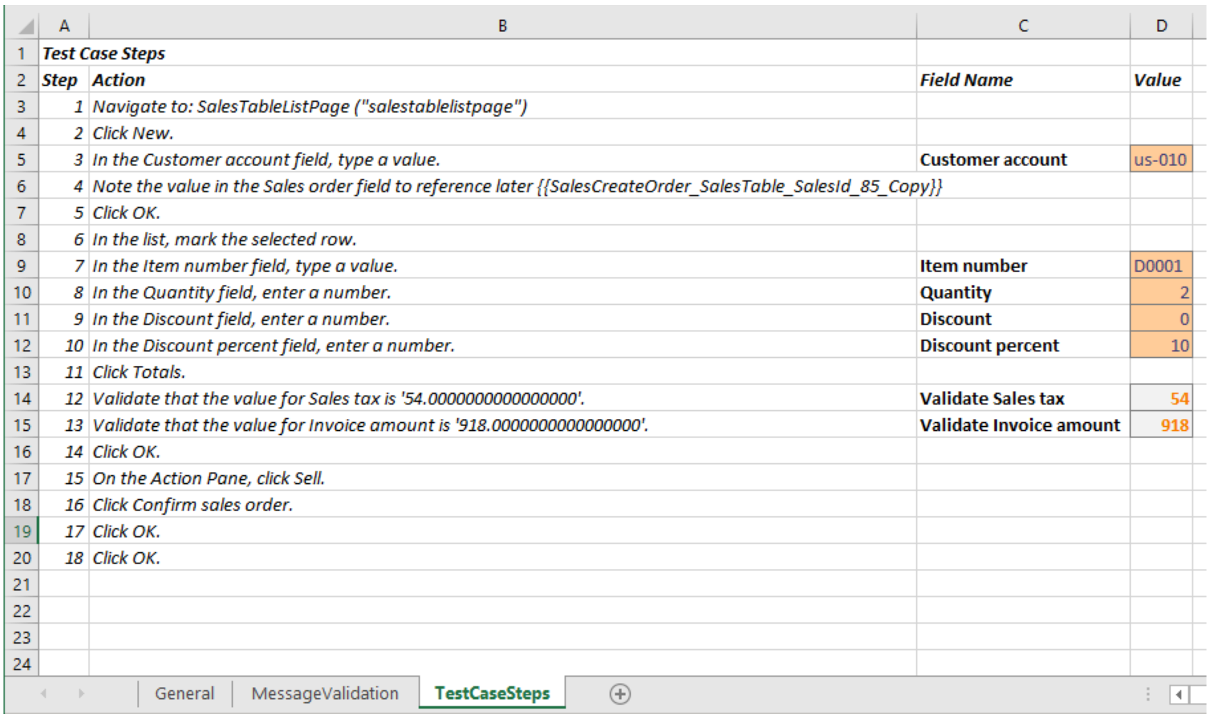

Regression Suite Automation Tool
Load
Generate
Run
Stop
Upload
Open
Generate Test Execution and Parameter files
Generate Test Execution files only
Upgrade Parameter files (will also generate Test Execution files)
Create derived test case
Resolve test case ID mismatch
Test Sui
Accept
Pro
Ord
Order to Cash - Simple (3)
Procure to Pay (7)
Edit
Delete
15549 Create Vendor Account
15550 Create a Purchase Order v2
15551 Post a Product Receipt v2

Modify test parameters

Select one or more test cases to modify, and then select the **Parameters** button on the toolbar. An Excel window opens for each test case that you selected.

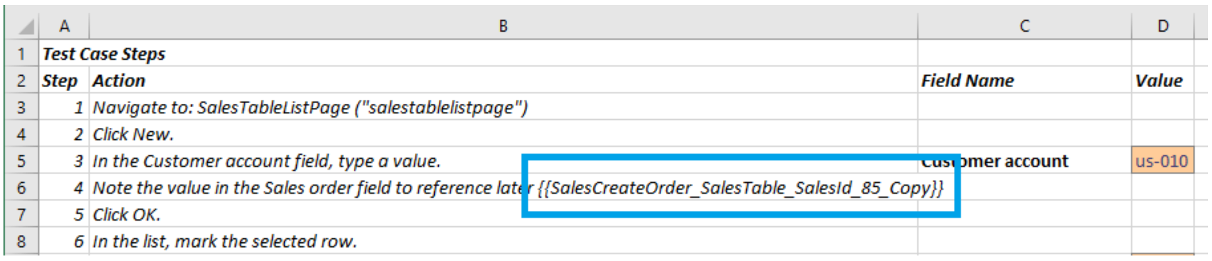
In addition to the **General** tab, the Excel parameter file contains a **MessageValidation** tab and a **TestCaseSteps** tab.

Select the **TestCaseSteps** tab to configure input and validation parameters of your test case.



	A	B	C	D
1	*Test Case Steps*			
2	*Step*	*Action*	*Field Name*	*Value*
3	1	Navigate to: SalesTableListPage ("salestablelistpage")		
4	2	Click New.		
5	3	In the Customer account field, type a value.	Customer account	us-010
6	4	Note the value in the Sales order field to reference later {{SalesCreateOrder_SalesTable_SalesId_85_Copy}}		
7	5	Click OK.		
8	6	In the list, mark the selected row.		
9	7	In the Item number field, type a value.	Item number	D0001
10	8	In the Quantity field, enter a number.	Quantity	2
11	9	In the Discount field, enter a number.	Discount	0
12	10	In the Discount percent field, enter a number.	Discount percent	10
13	11	Click Totals.		
14	12	Validate that the value for Sales tax is '54.0000000000000000'.	Validate Sales tax	54
15	13	Validate that the value for Invoice amount is '918.0000000000000000'.	Validate Invoice amount	918
16	14	Click OK.		
17	15	On the Action Pane, click Sell.		
18	16	Click Confirm sales order.		
19	17	Click OK.		
20	18	Click OK.		
21				
22				
23				
24				

Reusable variables that are copied while recording the test case are also shown in context of the test case step.

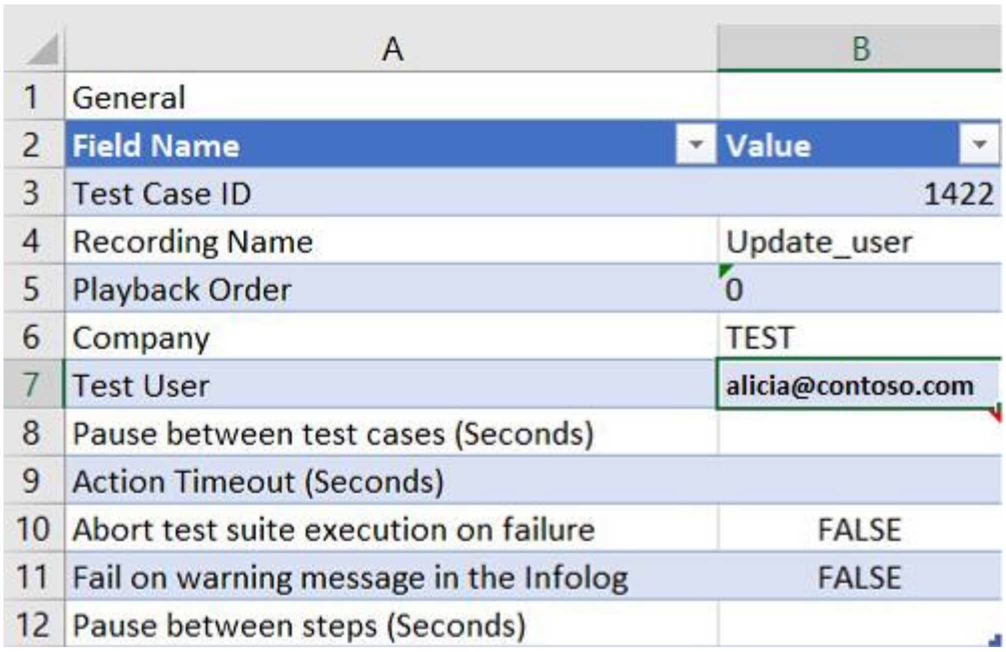


	A	B	C	D
1	*Test Case Steps*			
2	***Step***	***Action***	***Field Name***	***Value***
3	*1*	*Navigate to: SalesTableListPage ("salestablelistpage")*		
4	*2*	*Click New.*		
5	*3*	*In the Customer account field, type a value.*	**Customer account**	us-010
6	*4*	*Note the value in the Sales order field to reference later {{SalesCreateOrder_SalesTable_SalesId_85_Copy}}*		
7	*5*	*Click OK.*		
8	*6*	*In the list, mark the selected row.*		

Save the Excel files and select **Generate** to create new execution files that have the new parameters.

Run a test as a specific user

By default, tests are executed using the admin role. If you want to run the test as a specific security role, specify the email address of a user under the **Test User** parameter in the **General** tab of the Excel parameter file.



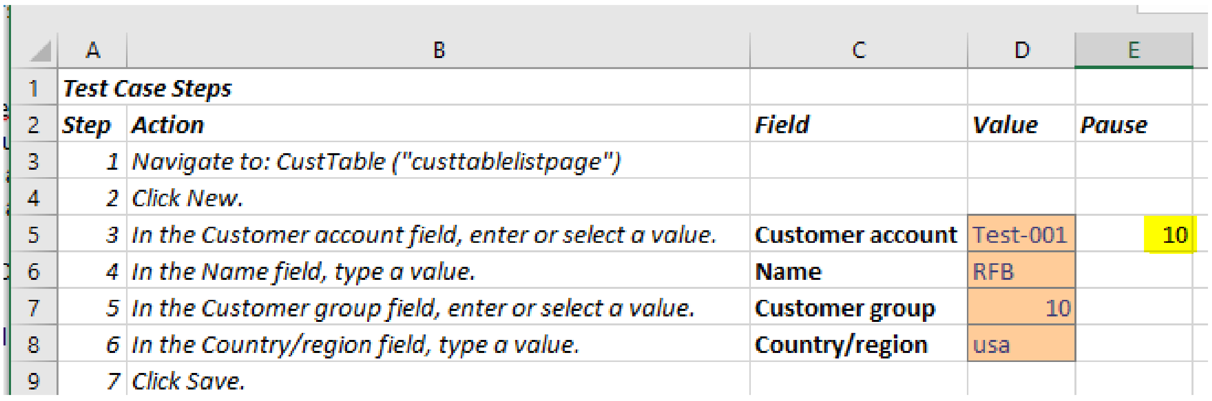
	A	B
1	General	
2	Field Name	Value
3	Test Case ID	1422
4	Recording Name	Update_user
5	Playback Order	0
6	Company	TEST
7	Test User	alicia@contoso.com
8	Pause between test cases (Seconds)	
9	Action Timeout (Seconds)	
10	Abort test suite execution on failure	FALSE
11	Fail on warning message in the Infolog	FALSE
12	Pause between steps (Seconds)	

Run a test in the context of a specific company

You can specify your default company in the **Settings** dialog box of the tool.

Pause after a specific test step

Navigate to the **TestCaseSteps** tab of the Excel parameters file and insert a value (in seconds) in the pause column of a test step.



	A	B	C	D	E
1	*Test Case Steps*				
2	*Step*	*Action*	*Field*	*Value*	*Pause*
3	*1*	*Navigate to: CustTable ("custtablelistpage")*			
4	*2*	*Click New.*			
5	*3*	*In the Customer account field, enter or select a value.*	Customer account	Test-001	10
6	*4*	*In the Name field, type a value.*	Name	RFB	
7	*5*	*In the Customer group field, enter or select a value.*	Customer group	10	
8	*6*	*In the Country/region field, type a value.*	Country/region	usa	
9	*7*	*Click Save.*			

If you don't see the **Pause** column, Select the desired test case, and then select **Generate > Generate Test Execution and Parameter files**.

Other notable test case execution settings

You may find the following settings useful. They are available on the **General** tab of the Excel parameter file.

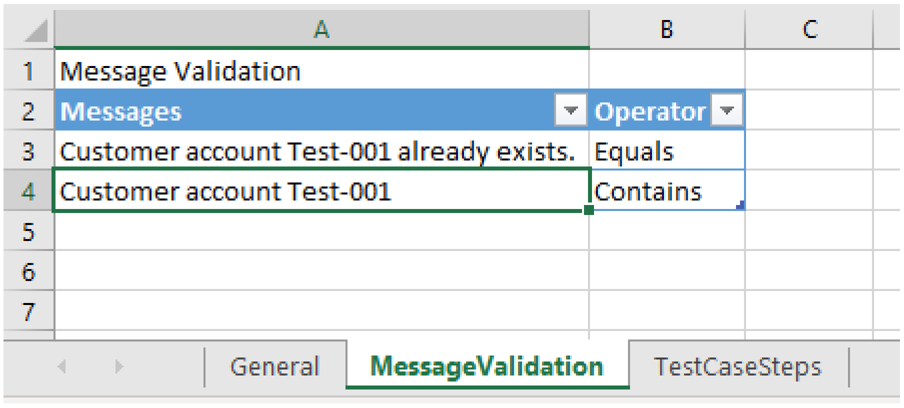
- **Fail on warning message in the Infolog**
- **Abort test suite execution on failure**
- **Pause between steps**

Infolog and message validation

Excel parameter files contain a **MessageValidation** tab.

You can enter messages in this tab under **Message Validation**. After a test case completes execution, it validates that the and displays it in the Infolog. The test case fails if these messages are not found.

You can specify any expected messages including error messages. Two operators are available: **Equals** and **Contains**. If you use **Equals**, then RSAT performs a string comparison. If you use **Contains**, then RSAT validates that at least one message in the Infolog contains the string you specify.



	A	B	C
1	Message Validation		
2	Messages	Operator	
3	Customer account Test-001 already exists.	Equals	
4	Customer account Test-001	Contains	
5			
6			
7			

Run

Select **Run** to execute the selected test cases.

Pause prior to a test case run

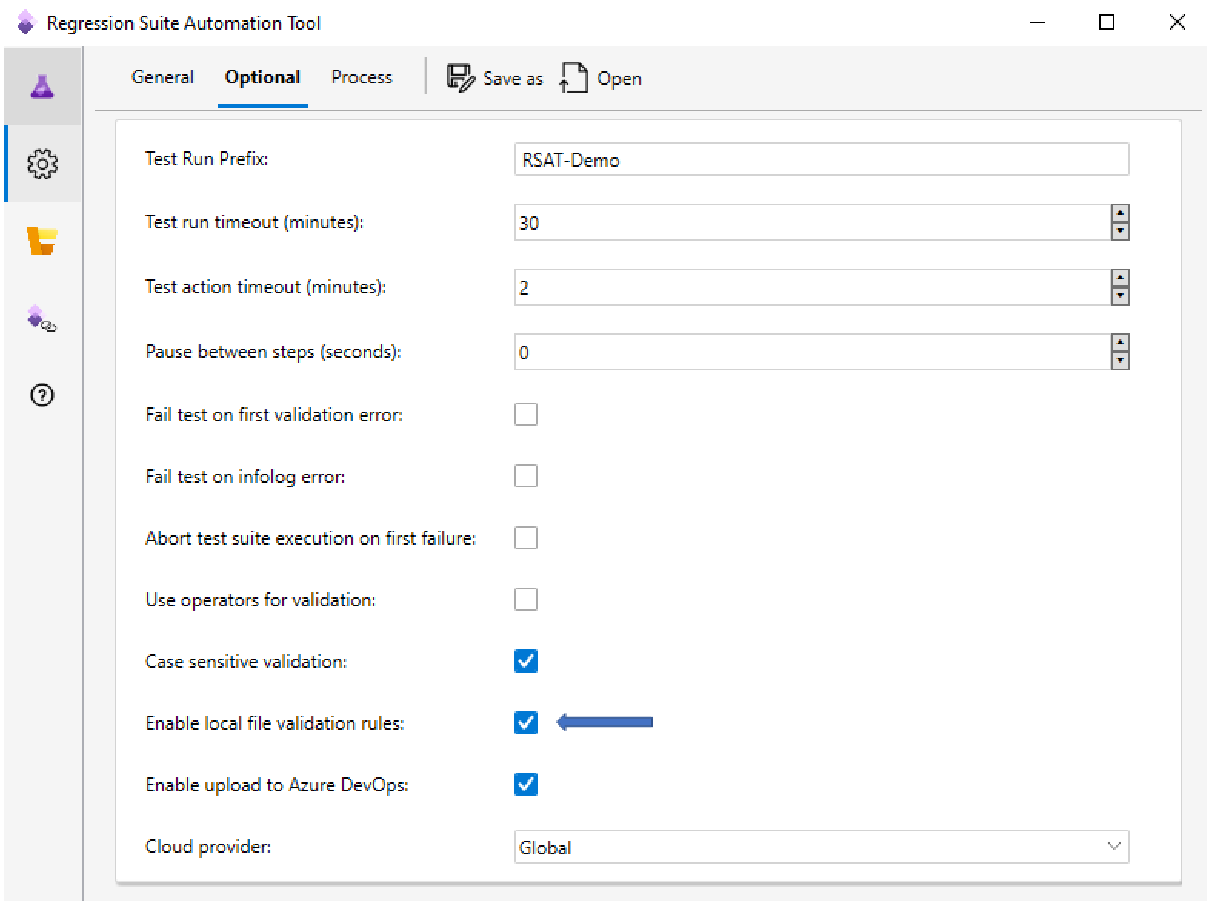
If you want to pause, update the cell **Pause (seconds)** on the **General** tab of the Excel parameters file of the desired test case.

Stop a run

When a test run is in progress, you can select the **Stop** button on the toolbar to cancel the run.

Validate readiness of test automation files

Optionally, you can turn on a setting that validates whether your test cases are ready for execution. You can enable this by selecting the **Settings** tab and then selecting the **Optional** tab.



Test Run Prefix: RSAT-Demo

Test run timeout (minutes): 30

Test action timeout (minutes): 2

Pause between steps (seconds): 0

Fail test on first validation error: ☐

Fail test on infolog error: ☐

Abort test suite execution on first failure: ☐

Use operators for validation: ☐

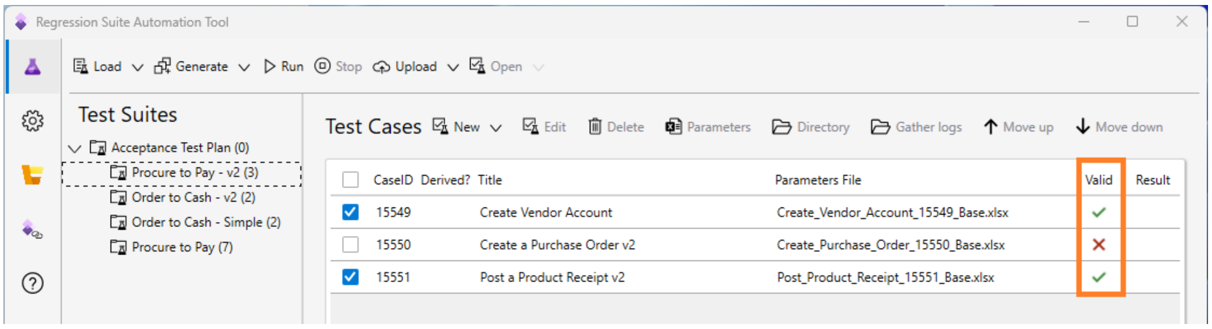
Case sensitive validation: ✓

Enable local file validation rules: ✓

Enable upload to Azure DevOps: ✓

Cloud provider: Global

The Valid column in the grid indicates the result of the validation process.


Regression Suite Automation Tool

Load | Generate | Run | Stop | Upload | Open

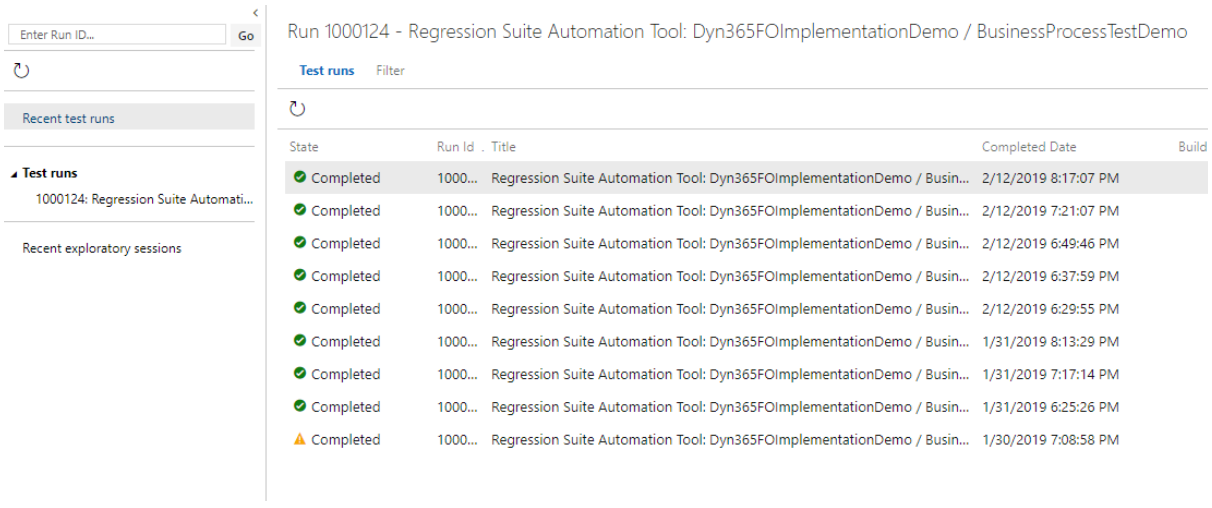
Test Suites
- Acceptance Test Plan (0)
  - Procure to Pay - v2 (3)
  - Order to Cash - v2 (2)
  - Order to Cash - Simple (2)
  - Procure to Pay (7)

Test Cases | New | Edit | Delete | Parameters | Directory | Gather logs | Move up | Move down

	CaseID	Derived?	Title	Parameters File	Valid	Result
✓	15549		Create Vendor Account	Create_Vendor_Account_15549_Base.xlsx	✓	
	15550		Create a Purchase Order v2	Create_Purchase_Order_15550_Base.xlsx	✗	
✓	15551		Post a Product Receipt v2	Post_Product_Receipt_15551_Base.xlsx	✓	

Investigate results

When all test cases complete execution, **Pass** or **Fail** is populated in the **Result** column. To view investigation details information, from your Azure DevOps project page, go to **Test** > **Runs**.


Enter Run ID... | Go

Recent test runs

Test runs
- 1000124: Regression Suite Automati...

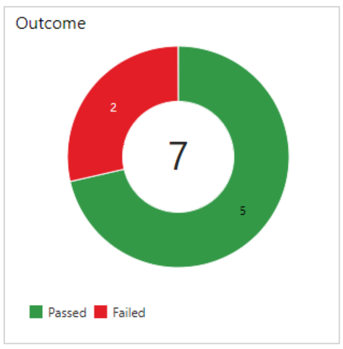
Recent exploratory sessions

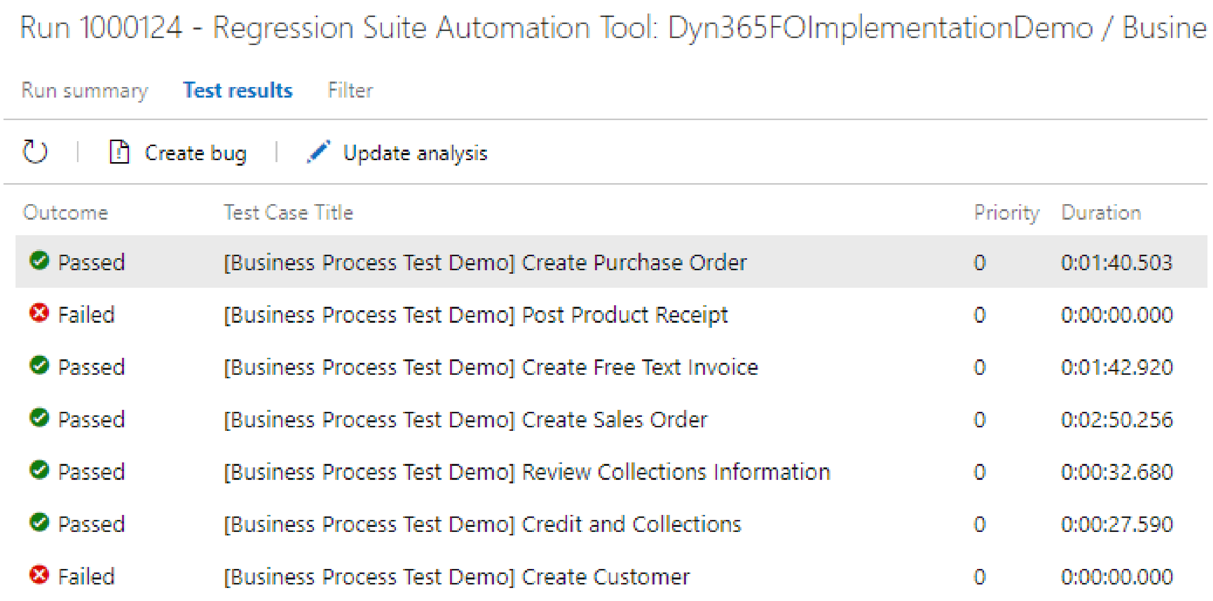
Run 1000124 - Regression Suite Automation Tool: Dyn365FOImplementationDemo / BusinessProcessTestDemo

Test runs | Filter

State	Run Id	Title	Completed Date	Build
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	2/12/2019 8:17:07 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	2/12/2019 7:21:07 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	2/12/2019 6:49:46 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	2/12/2019 6:37:59 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	2/12/2019 6:29:55 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	1/31/2019 8:13:29 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	1/31/2019 7:17:14 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	1/31/2019 6:25:26 PM	
Completed	1000...	Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busin...	1/30/2019 7:08:58 PM	

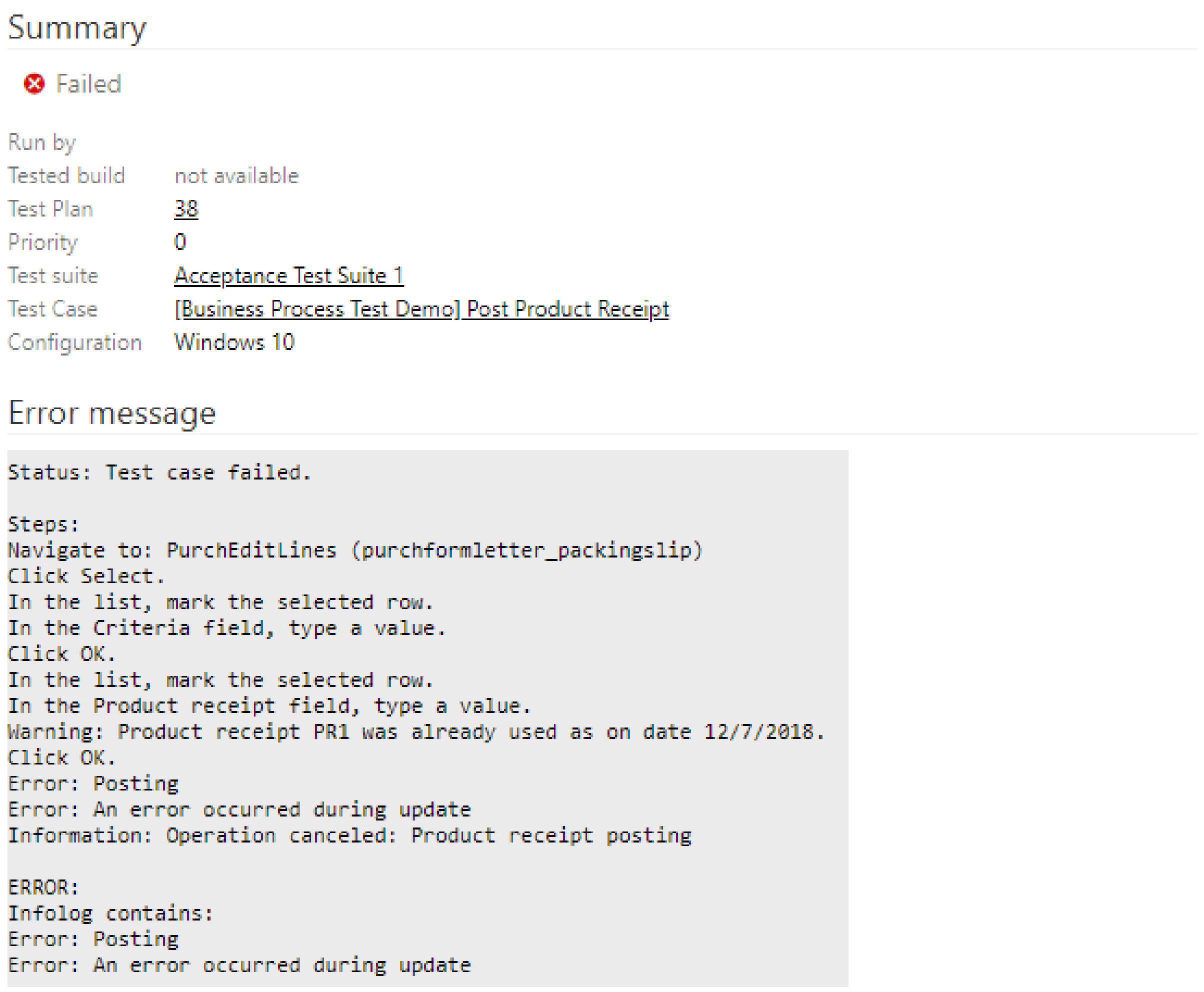
Select the desired test run. It includes the results of all tests that were executed during that run.




Run 1000124 - Regression Suite Automation Tool: Dyn365FOImplementationDemo / Busine

Outcome	Test Case Title	Priority	Duration
✓ Passed	[Business Process Test Demo] Create Purchase Order	0	0:01:40.503
✗ Failed	[Business Process Test Demo] Post Product Receipt	0	0:00:00.000
✓ Passed	[Business Process Test Demo] Create Free Text Invoice	0	0:01:42.920
✓ Passed	[Business Process Test Demo] Create Sales Order	0	0:02:50.256
✓ Passed	[Business Process Test Demo] Review Collections Information	0	0:00:32.680
✓ Passed	[Business Process Test Demo] Credit and Collections	0	0:00:27.590
✗ Failed	[Business Process Test Demo] Create Customer	0	0:00:00.000

You can open a failed test result and review the **ErrorMessage** section for information about the failure.


Summary

✗ Failed

Run by	
Tested build	not available
Test Plan	38
Priority	0
Test suite	Acceptance Test Suite 1
Test Case	[Business Process Test Demo] Post Product Receipt
Configuration	Windows 10

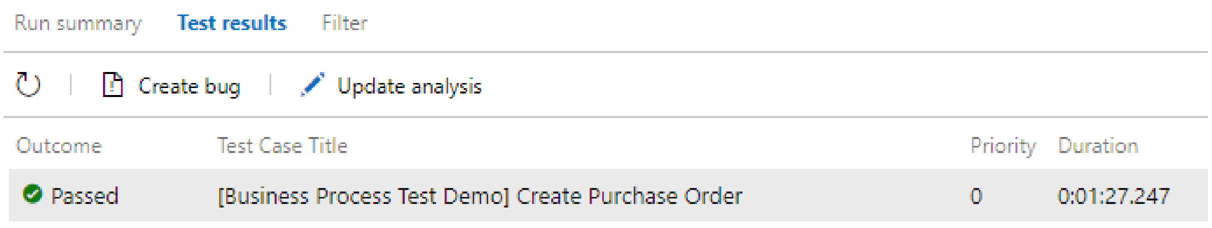
Error message

```
Status: Test case failed.

Steps:
Navigate to: PurchEditLines (purchformletter_packingslip)
Click Select.
In the list, mark the selected row.
In the Criteria field, type a value.
Click OK.
In the list, mark the selected row.
In the Product receipt field, type a value.
Warning: Product receipt PR1 was already used as on date 12/7/2018.
Click OK.
Error: Posting
Error: An error occurred during update
Information: Operation canceled: Product receipt posting

ERROR:
Infolog contains:
Error: Posting
Error: An error occurred during update
```

All error messages are also available locally under C:\Users\$YourUserName\AppData\Roaming\regressionTool\errormsg-<TestCaseId>.txt.

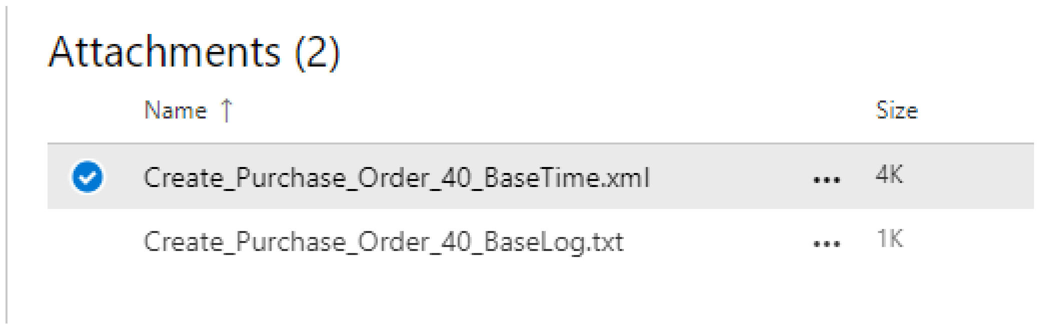
Test response times

The duration of a test case is also available in the test result.



Outcome	Test Case Title	Priority	Duration
✓ Passed	[Business Process Test Demo] Create Purchase Order	0	0:01:27.247

You can also review the response time of each step of the test case by opening the **BaseTime.xml** file attached to the test result.


Attachments (2)

	Name ↑		Size
✓	Create_Purchase_Order_40_BaseTime.xml	•••	4K
	Create_Purchase_Order_40_BaseLog.txt	•••	1K

Upload to Azure DevOps to commit your work

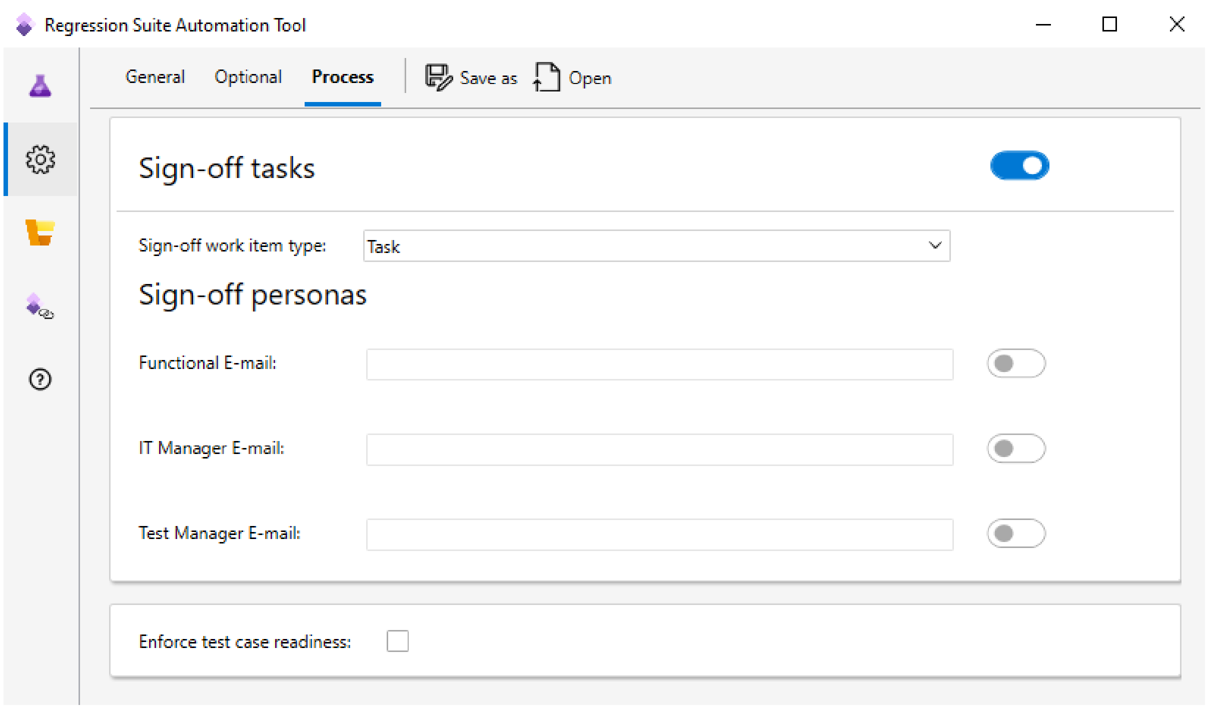
To commit your work to Azure DevOps, select **Upload**. After upload, you use the Regression suite automation tool, you can use **Load** and then **Run**, without generating test execution files or editing Excel parameter files.

In the upload menu, you also have the option to upload recording files.

To commit all changes to Azure DevOps, select **Upload all modified automation files** in the upload menu.

Process compliance

RSAT provides capabilities for managing the readiness of test cases and signoff process for test runs. This is configurable in the **Process** tab under Settings.



Sign-off tasks

Sign-off work item type: Task

Sign-off personas

Functional E-mail:

IT Manager E-mail:

Test Manager E-mail:

Enforce test case readiness: ☐

Enforce test case readiness

Select the **Enforce test case readiness** check box.

Result:

Thus, to Run regression tests using Maven build pipeline in Azure was performed successfully and verified.