Example 1: Creating and running procedures

7 minute read **Developer productivity**

Procedures are the basic automation objects in CloudBees CD/RO. They are used to implement tool integrations and can be used for almost any automation task, such as running scripts and command line actions.

Procedures are made up of *steps*. A *command step* invokes a block of shell or script commands on a CloudBees CD/RO agent.

In this example, you create a simple procedure that:

- Produces standard I/O output from a CloudBees CD/RO agent machine.
- Accepts input from parameters (both static and dynamic).
- Runs steps conditionally.

You also customize the runtime job name and step summaries.

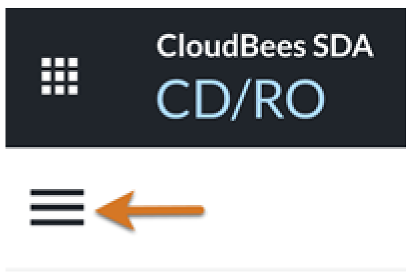
Because procedures run inside of projects, you must create a project first.

Create a project

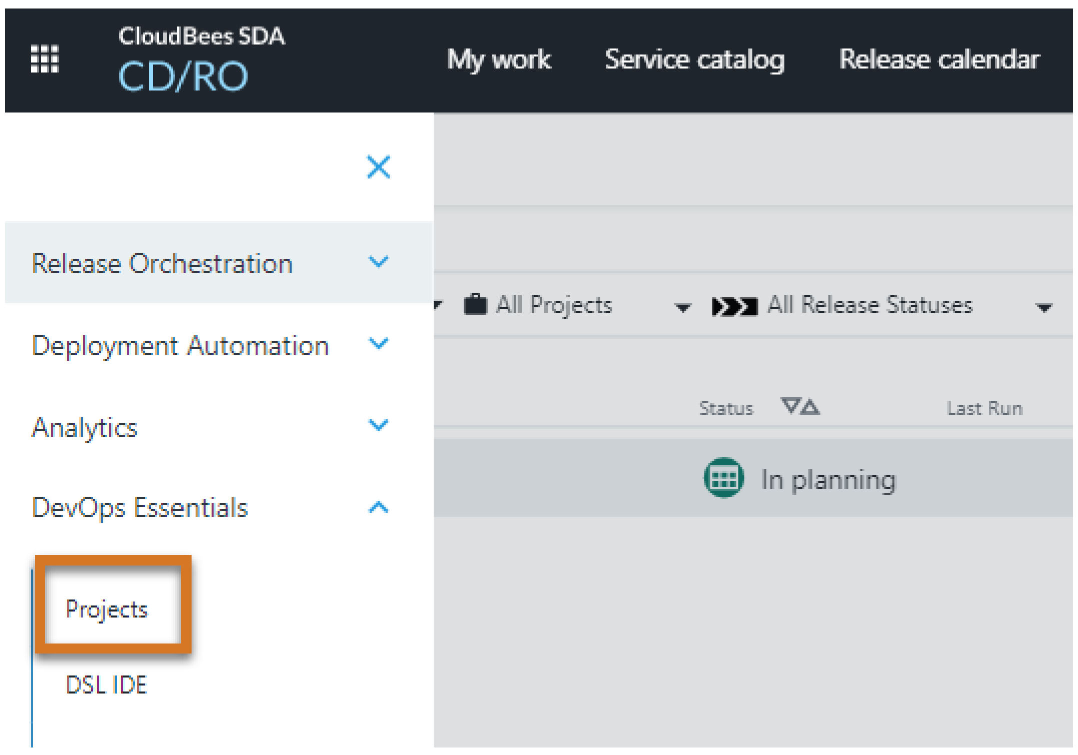
Before you create a procedure, you need to create a container for the procedure, which is called a *project*. Projects are root-level objects that can contain procedures, pipelines, releases, applications, and other modeling objects. For more information about objects, see Understanding CloudBees CD/RO objects.

To create a project:

1. Select the CloudBees CD/RO main menu.



2. Navigate to **DevOps Essentials > Projects**.



The project list appears.

3. Select **New+** in the upper right. The New Project dialog appears.

4. Select **Create New**.

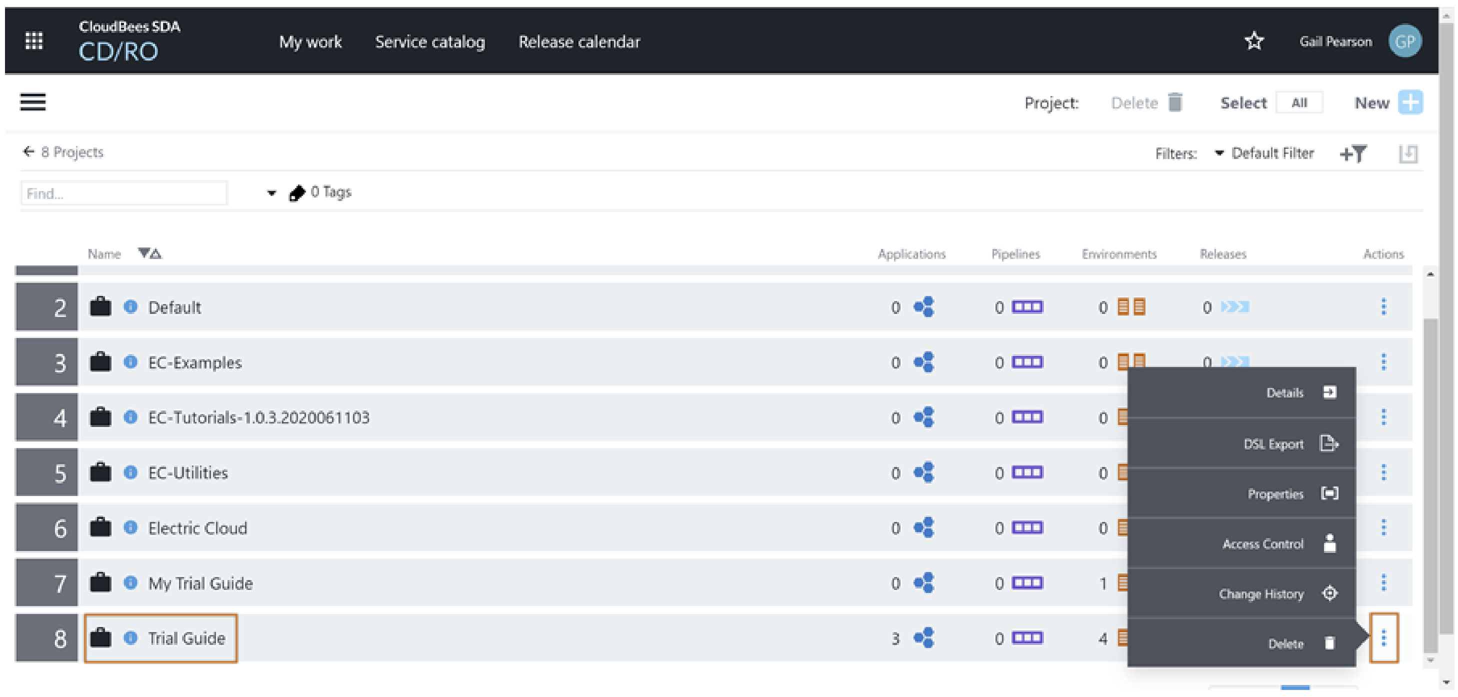
Submit Feedback

5. Enter the project name *Trial Guide* (or similar name) and then select **OK**.

   You will not be using credentials (secrets), so when asked if you want to create or edit credentials, select **No**.

   Tip

   After the project is created, you can search for it by name. When it is displayed, select the menu under **Actions** to access the project's attributes.

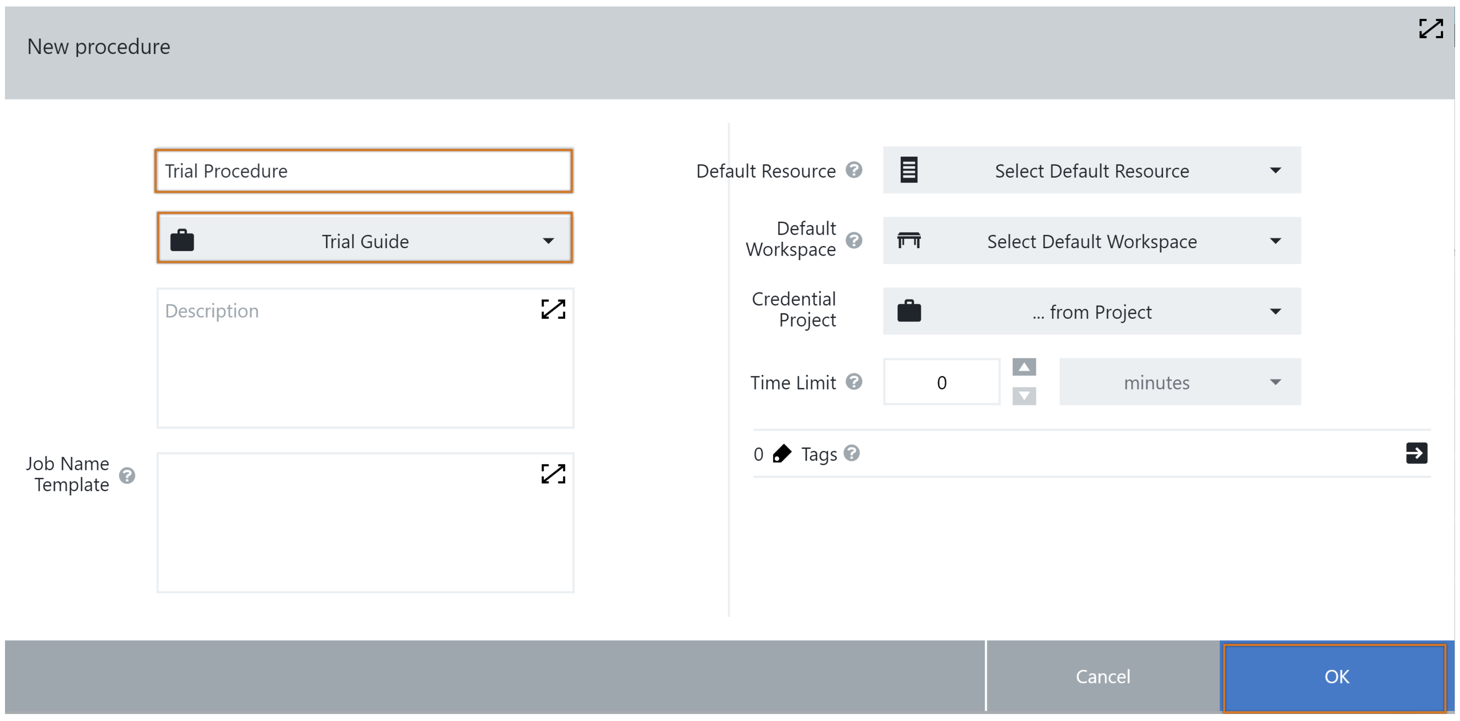


	Name	Applications	Pipelines	Environments	Releases	Actions
2	Default	0	0	0	0	⋮
3	EC-Examples	0	0	0	0	⋮
4	EC-Tutorials-1.0.3.2020061103	0	0	0		⋮
5	EC-Utilities	0	0	0		⋮
6	Electric Cloud	0	0	0		⋮
7	My Trial Guide	0	0	1		⋮
8	Trial Guide	3	0	4		⋮

Create a procedure

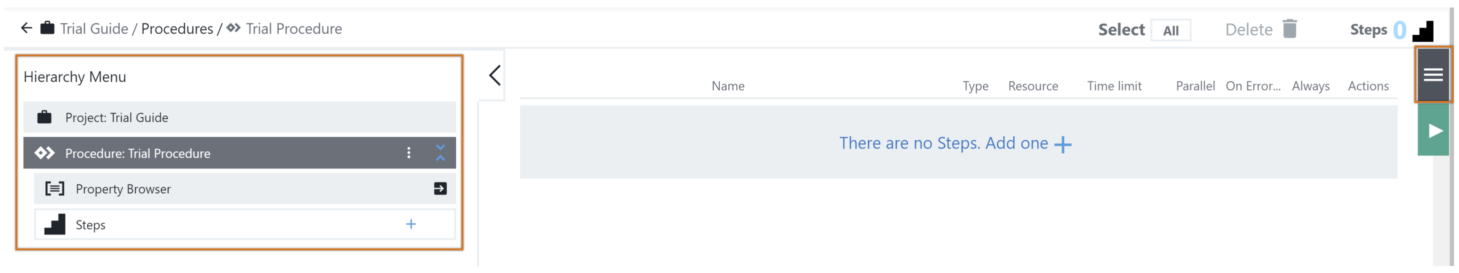
Now that you have a project to contain objects, do the following to create a procedure.

1. From the CloudBees CD/RO main menu, navigate to **DevOps Essentials > Procedures**. The list of procedure objects displays.
2. Select **New+** in the upper right to add a procedure. The New Procedure dialog displays.
3. Select **Create New**.
4. Enter a procedure name such as *Trial Procedure*. Select the target project that you created above (for example, *Trial Guide*) and then select **OK**.



	Trial Procedure
	Trial Guide
	Description
Job Name Template	
Default Resource	Select Default Resource
Default Workspace	Select Default Workspace
Credential Project	... from Project
Time Limit	0 minutes
0 Tags	

You are now in a procedure. The Hierarchy Menu on the left appears, as well as a Procedure menu on the right.



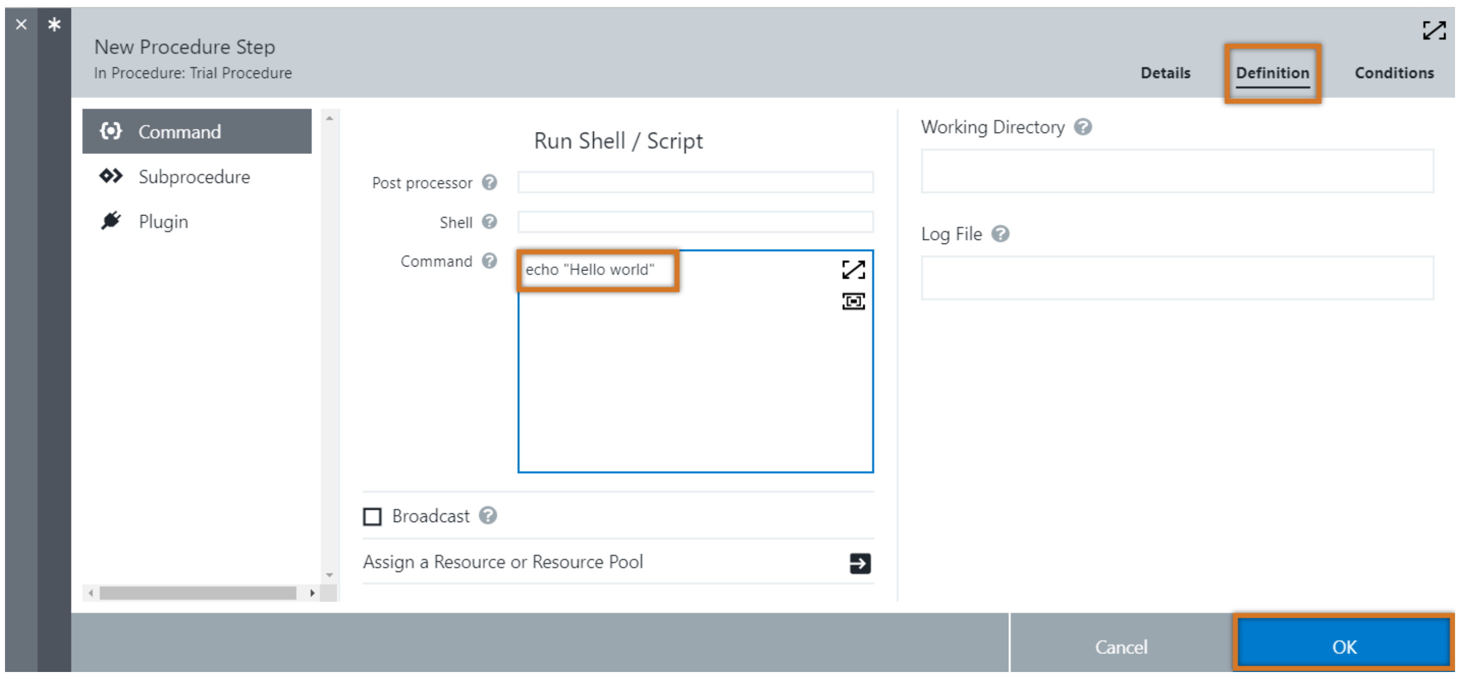
There are no Steps. Add one +

Add a step

When a procedure is first created, it has no steps to execute, so you must add steps.

Select **There are no Steps. Add one +** to begin adding a step. The New Procedure Step dialog appears.

1. Select **Create New**.
2. Provide a name, *Hello World*, and a description for the step — *A step to echo out a simple message*.
3. Select **Definition** in the upper right of the dialog to define the functionality of the step.
4. In the **Command** field, enter this command to be run: `echo "Hello world"` .
5. Select **OK**.



```
echo "Hello world"
```

Your *Hello World* step is displayed in the list of steps.

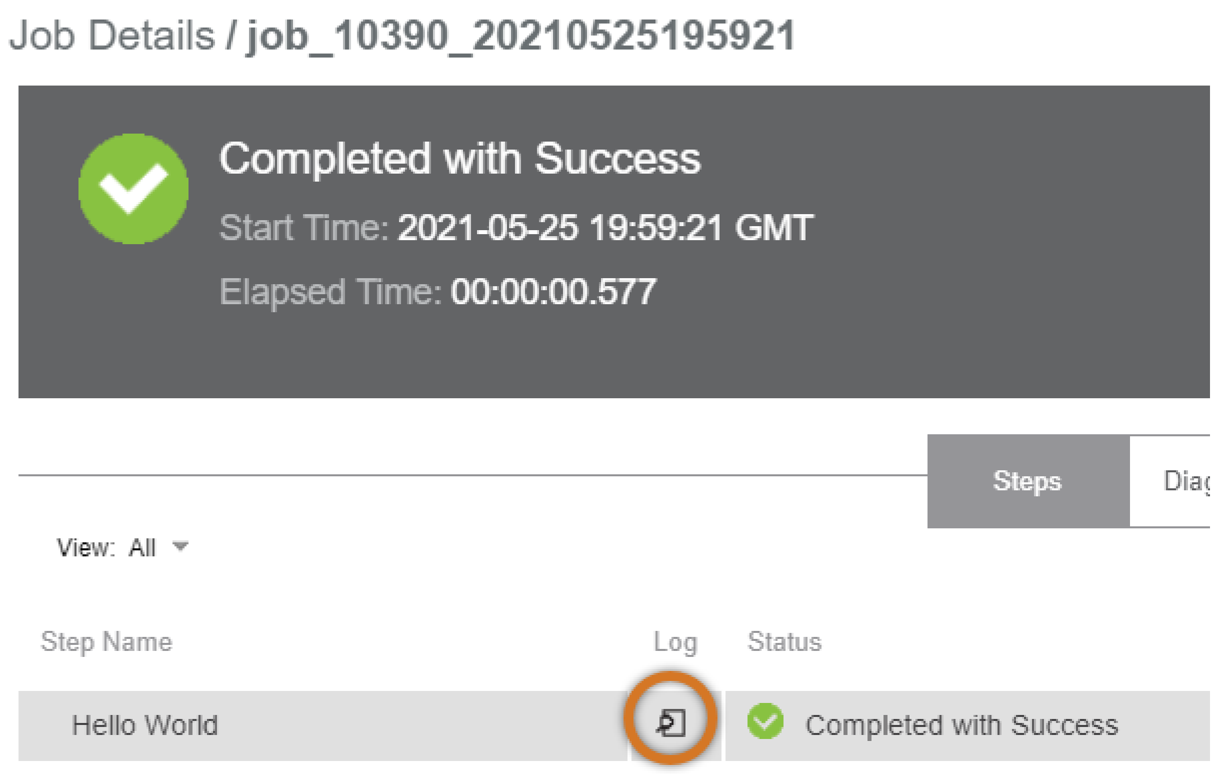
Run the procedure for the first time

1. Select the Run Procedure icon (green arrow) on the right, and then select **New Run**. The Run Procedure dialog box appears. Select **OK**. Note the job's progress.

Tip

The abort button can be used to halt a procedure job.

2. View the job details by selecting **Procedure Runs** at the top of the page, then selecting the job number. Select the job once again to be able to view the log.
3. Select the icon under **Log** to view the job output.


Job Details / job_10390_20210525195921

Completed with Success
Start Time: 2021-05-25 19:59:21 GMT
Elapsed Time: 00:00:00.577

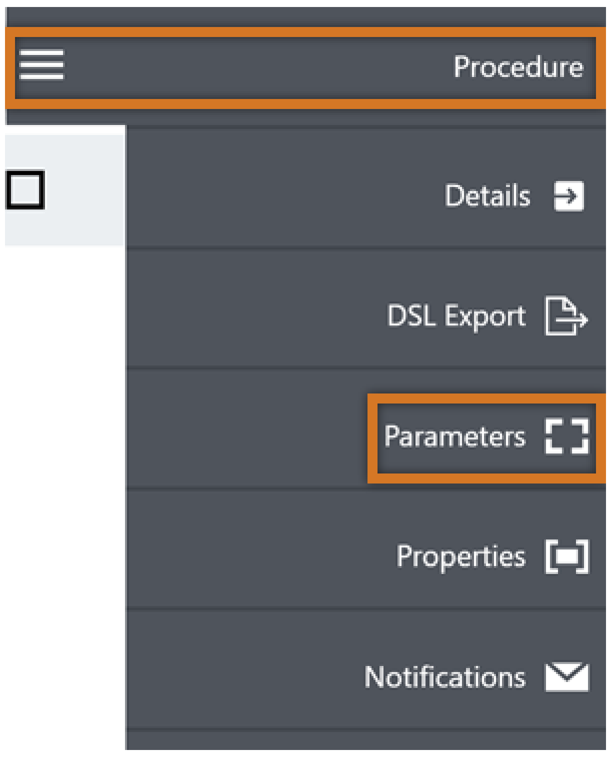
Step Name	Log	Status
Hello World		Completed with Success

You should now see the *Hello world* message.

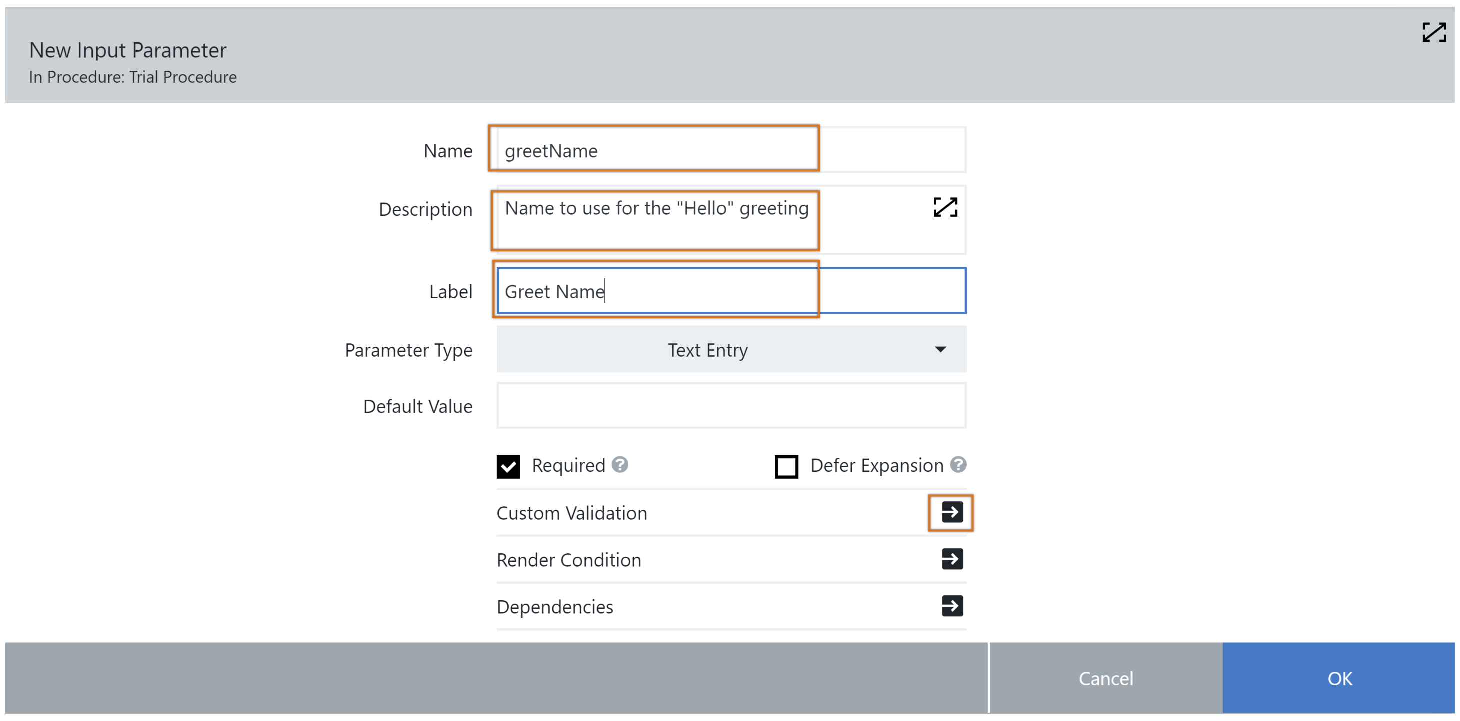
4. Close the job dialog.

Add a parameter

1. From the breadcrumb at the top of the window, return to the procedure by selecting **Trial Procedure**.
2. Open the Procedure menu on the right and then select **Parameters**.



3. Add a parameter by selecting **There are no Parameters. Add one +**. The New Input Parameter dialog appears.
4. Provide the the following values:
   Name - *greetName*
   Description - *Name to use for the "Hello" greeting*
   Label - *Greet Name*
5. Select **Custom Validation** to enter a validation script.


New Input Parameter
In Procedure: Trial Procedure

Name: greetName
Description: Name to use for the "Hello" greeting
Label: Greet Name
Parameter Type: Text Entry
Default Value:
Required [✓] Defer Expansion [ ]
Custom Validation
Render Condition
Dependencies
Cancel OK

6. In the **Provide DSL** box, enter the following DSL code. This code prevents entries that include the space character.

```
if(args.parameters['greetName']?.contains(" "))
    return "No spaces allowed!"
else
    return null
```

Select **OK**.

Note

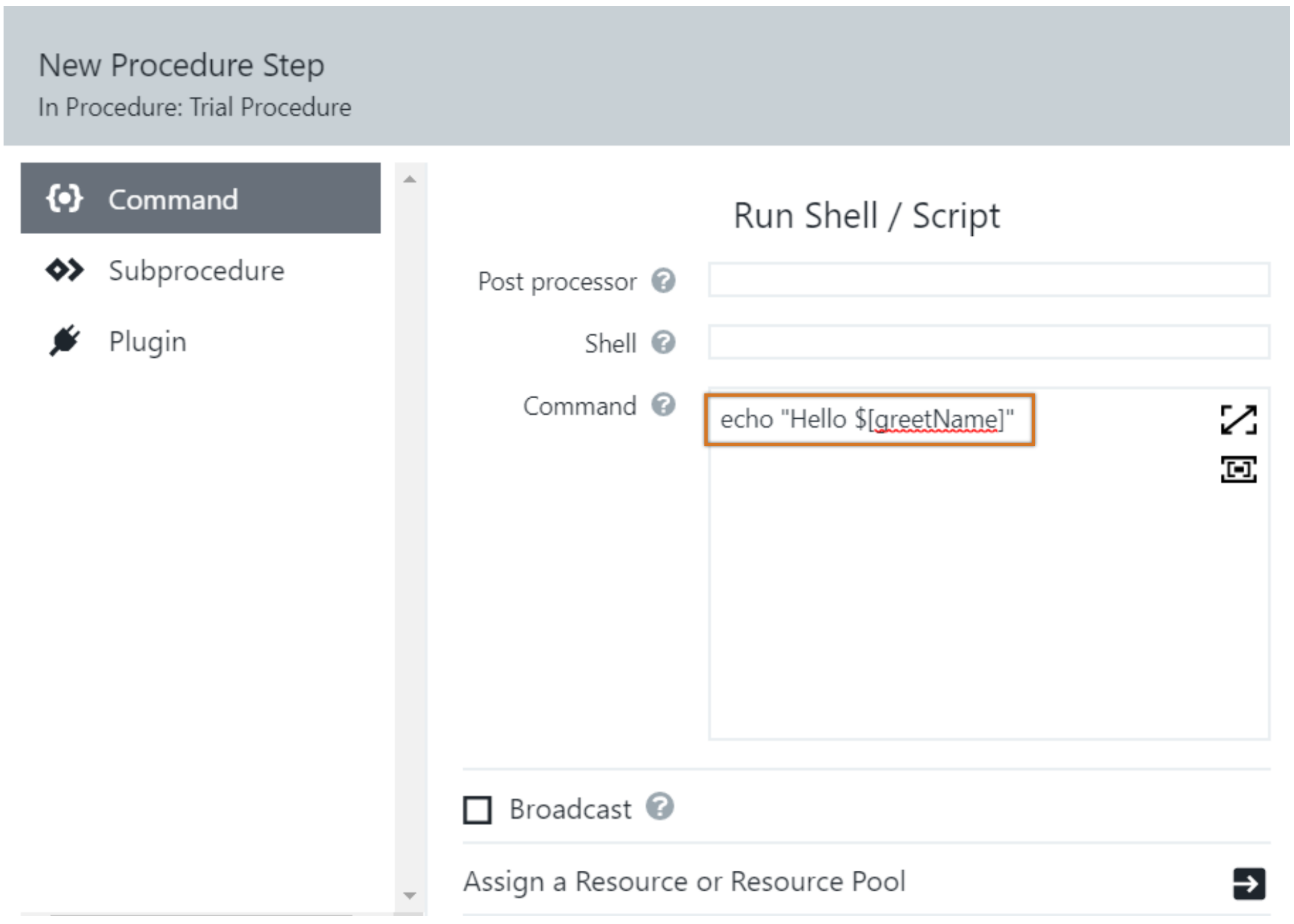
When the validation script is saved, CloudBees CD/RO runs the DSL script to validate it. The `args.parameters['greetName']` returns null in this context because the parameter hasn't been set yet. The question mark tells DSL to ignore what follows when this is null.

7. Select **OK** to complete the parameter definition.
8. To exit the Parameters dialog, select **X** in the upper left or the **Esc** key.

Use the parameter in a step

Duplicate the Hello World step by selecting the Step icon in the upper right, or the plus (+) next to **Steps** in the Hierarchy Menu on the left.

1. Select **Copy Existing**.
2. Select the *Hello World* step to copy.
3. Enter the new step name *Hello <greetName>* and update the description with the text *A step to echo out a greeting to someone*. Select **OK**. (The new step name appears after the step is saved.)
4. To edit the new step, select the step name or select **Step Definition** under the Actions menu.
5. Modify the command block so that it includes only the command `echo "Hello $[greetName]"`

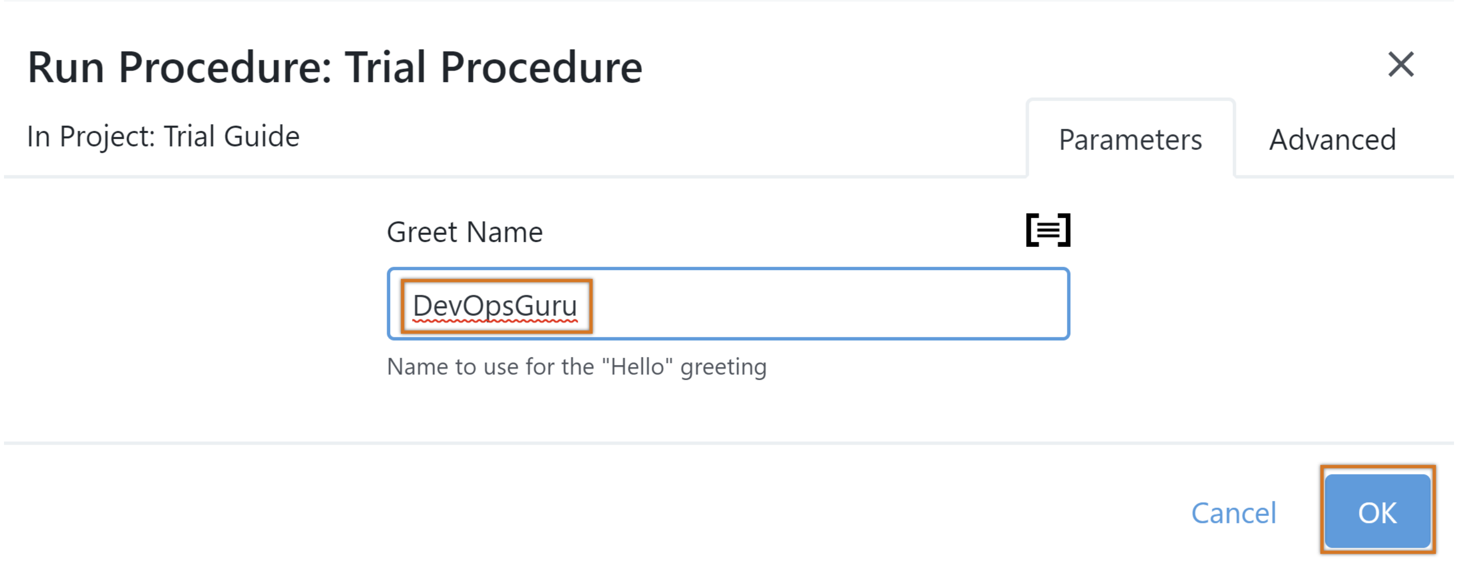


```
echo "Hello $[greetName]"
```

6. Select **OK**.

Run the modified procedure

1. Select the Run Procedure button (green arrow) to select a run option.
2. Select either **New Run** or **Last Run**. The latter will remember any user-supplied parameter settings. (In this case, there were none, so both options behave the same way.)
3. Try supplying this name, which includes spaces: *A DevOps Guru*. (Because of the validation code you supplied, this parameter field does not accept such an entry.)
4. Now enter the name *DevOpsGuru* (without spaces) and select **OK**.



Run Procedure: Trial Procedure

In Project: Trial Guide

Parameters | Advanced

Greet Name: DevOpsGuru

Name to use for the "Hello" greeting

Cancel | OK

5. Navigate to the Job Details using the **Action** menu and examine the step's log file. You should see **Hello DevOpsGuru**.
6. Exit this job's dialog.

Tip

The Rerun button (green arrow in a circle) can be used to run the procedure again with this job's parameter values already filled in, based on the previous run.

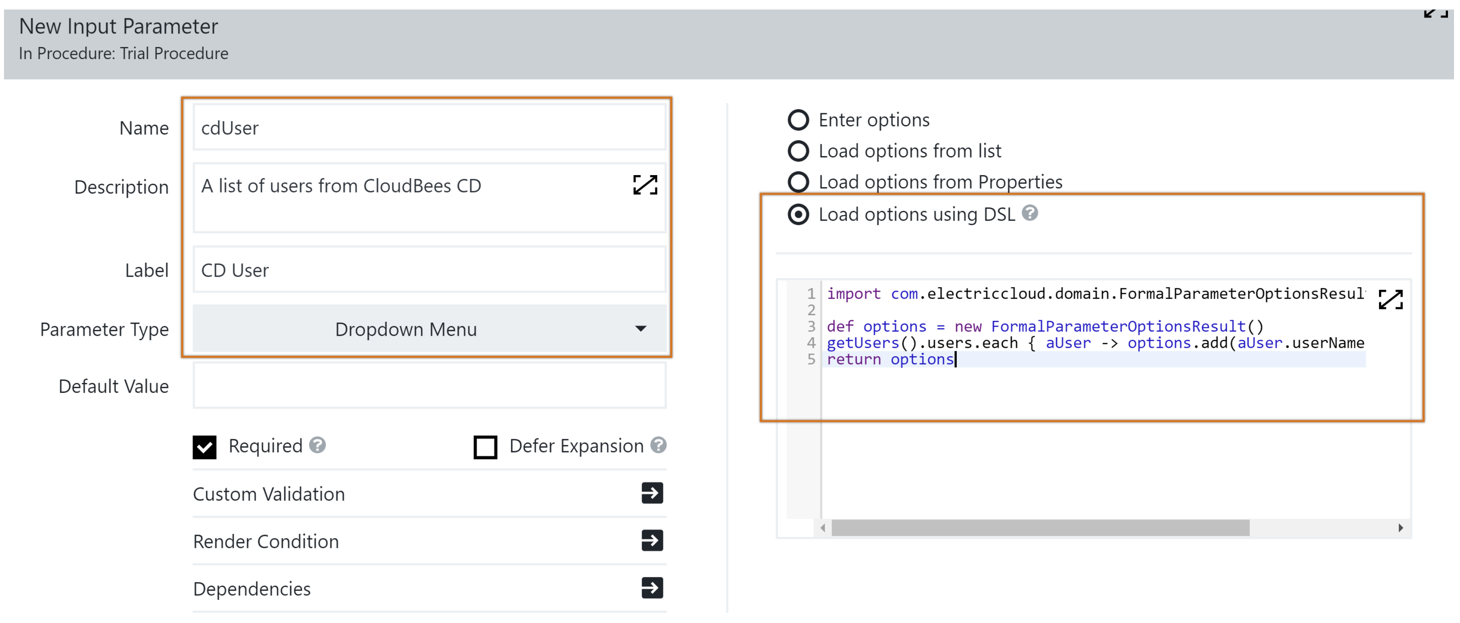
Create a multi-option parameter

Create a new input parameter that enables you to select from a list of options before running the procedure.

1. Navigate to the procedure using the breadcrumbs.
2. Using the Procedure menu on the right, select **Parameters**. The Parameters dialog appears.
3. Create a new input parameter by selecting **Input+** in the upper right. Supply the following information:
   Name - *cdUser*
   Description - *A list of users from CloudBees CD*
   Label - *CD User*
4. In **Parameter Type**, select **Dropdown Menu,** and then select **Load options using DSL**.
5. Supply the following DSL code:

```
import com.electriccloud.domain.FormalParameterOptionsResult

def options = new FormalParameterOptionsResult()
getUsers().users.each { aUser -> options.add(aUser.userName, aUser.fullUserName)}
return options
```



New Input Parameter
In Procedure: Trial Procedure

Name: cdUser
Description: A list of users from CloudBees CD
Label: CD User
Parameter Type: Dropdown Menu
Default Value:

[✓] Required   [ ] Defer Expansion

Custom Validation
Render Condition
Dependencies

Enter options
Load options from list
Load options from Properties
[✓ Load options using DSL]

```
import com.electriccloud.domain.FormalParameterOptionsResul

def options = new FormalParameterOptionsResult()
getUsers().users.each { aUser -> options.add(aUser.userName
return options
```

When the script is saved, CloudBees CD/RO runs the DSL script to validate it. You can use the DSL IDE to create and debug them. If there are dependencies on other parameters, you'll need to include their definitions in the test script. For example:

```
def args = [parameters: [aParameter: "Parm value"] ]
```

6. Select **OK** to save the new parameter and then close the Parameters dialog. You are returned to the step object list.

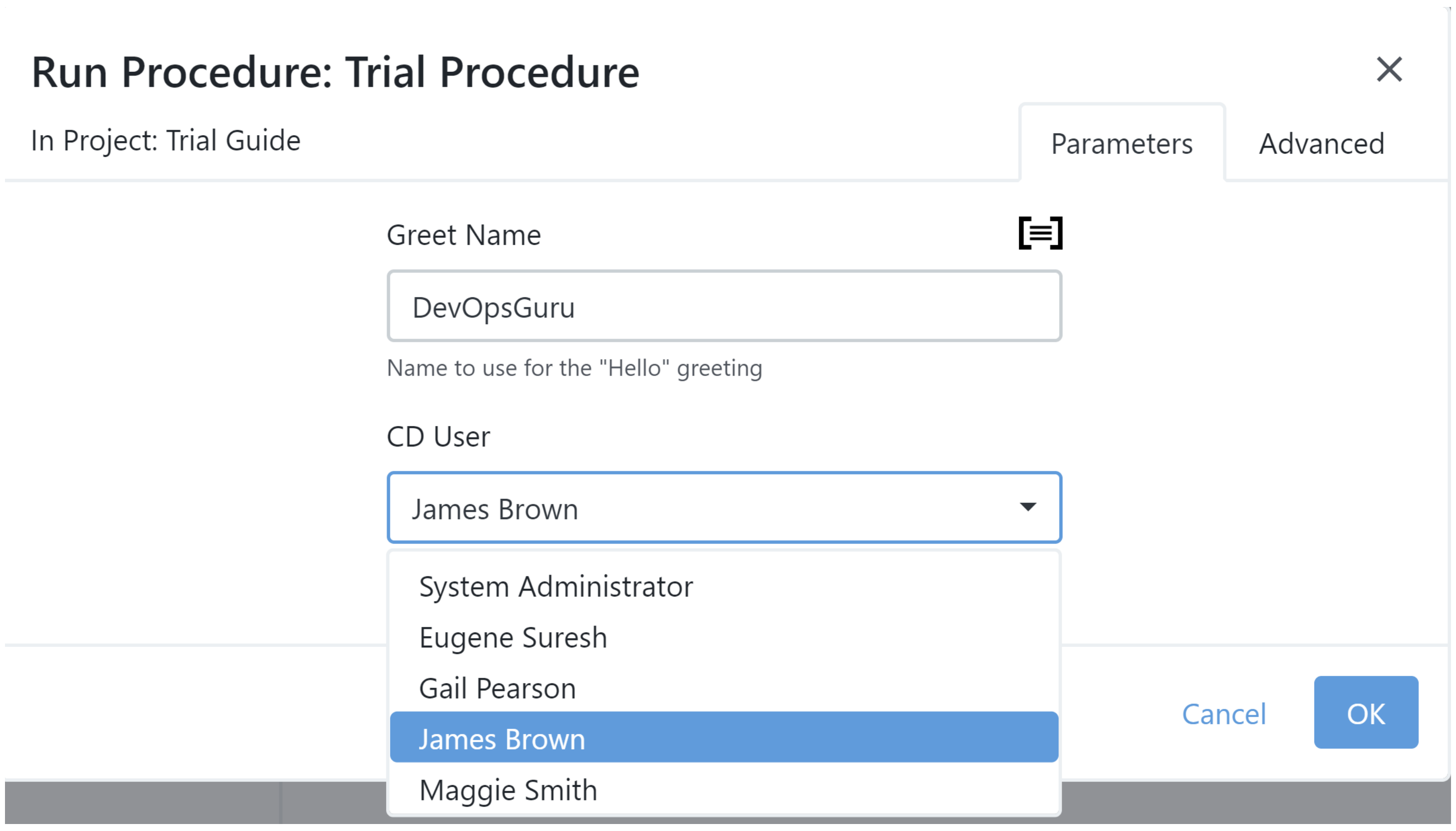
Use the multi-option parameter in a step

Duplicate the second step and modify it to use the new parameter.

1. From the list of step objects, create a new step using one of the methods discussed earlier.
2. In the New Procedure Step dialog, select **Copy Existing**, and then select the *Hello <greetName>* step.
3. Change the name of the step to *Hello CD User*.
4. Select **Definition**.
5. Modify the Command block to include the parameter substitution `$[cdUser]` so that the new command block is `echo "Hello $[cdUser]"` .
6. Select **OK**.

Rerun the procedure

1. Select the Run button (green arrow) and then select **Last Run**.
2. Fill in the parameter values. The pull-down options are populated with a list of users that have been defined in your CloudBees CD/RO instance.



Greet Name: DevOpsGuru
Name to use for the "Hello" greeting
CD User: James Brown
System Administrator
Eugene Suresh
Gail Pearson
James Brown
Maggie Smith

3. To view the step log for the *Hello CD User* step, select **Job Details** from the **Actions** menu, then select the **View Log** icon in the upper right. You should see your "Hello" text displayed.

Customize the job name and job step summary

Now, modify the procedure to create a job name based on the name of the procedure and a count.

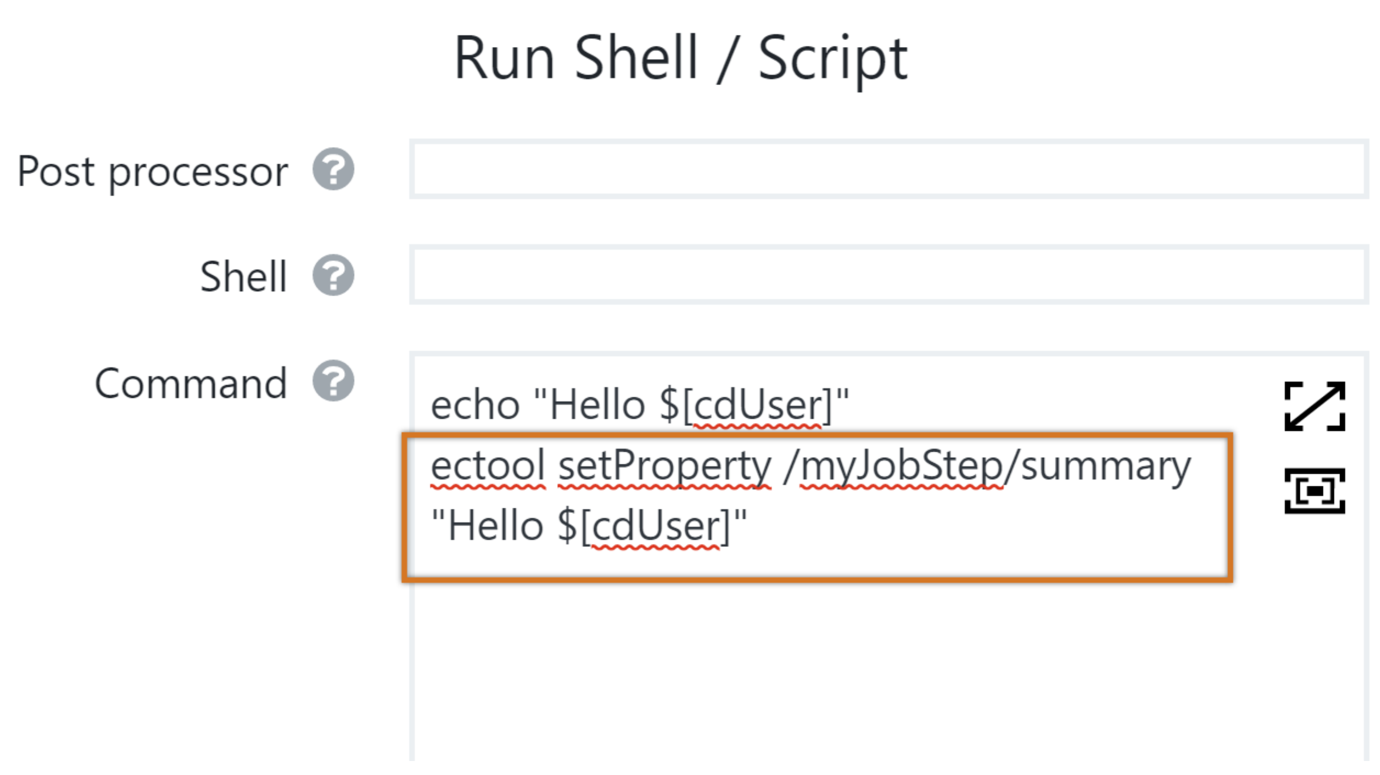
1. Navigate to the *Trial Procedure* editor using the breadcrumb, and then select **Details** from the Procedure menu on the right.
2. In the **Job Name Template** field, add the following syntax to override the blank default value:

   ```
   $[/myProcedure]-$[/increment /myProcedure/count]
   ```

   This code creates a job name based on the name of the procedure (from `/myProcedure`) and a property (count) stored on the procedure property sheet (`/myProcedure`). The increment function creates the property if it doesn't exist, and increments it by one.
3. Select **OK**.

4. Edit the last step definition using the **Action** menu. In the **Command** field, below the echo command, add the following:

```
ectool setProperty /myJobStep/summary "Hello $[cdUser]"
```



```
echo "Hello $[cdUser]"
ectool setProperty /myJobStep/summary
"Hello $[cdUser]"
```

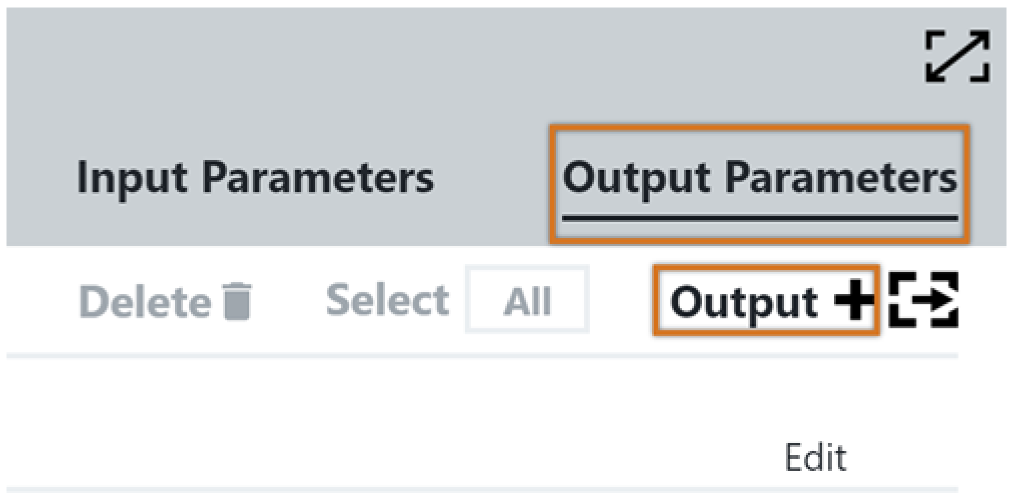
Run the procedure with the new job name

1. Select the Run button and select **Last Run**. Note the new job name format.
2. Navigate to the job details and note the new job step summary value.

Create and use an output parameter

Navigate to the *Trial* procedure to create an output parameter.

1. From the Trial procedure, select the Procedure menu on the right, and then select **Parameters**. The Parameters dialog appears.
2. Select the **Output Parameters** tab and add an output parameter called **whoRanMe**.



Select **OK**. Close the Parameters dialog.
3. Add a new procedure step and name it *Set output parameter*.

Tip

Select the **+** beside **Steps** in the Hierarchy Menu to add a step.

4. Navigate to the **Definition** tab.
5. Add the following shell command in the **Command** field:

```
ectool setOutputParameter whoRanMe "$[/myJob/launchedByUser]"
```

Select **OK**.
6. Run the procedure using the **Last Run** option to reuse the parameter values previously entered.
7. Navigate to the job details and select **Parameters**. Note that the **Output Parameter** value *whoRanMe* is set as expected.

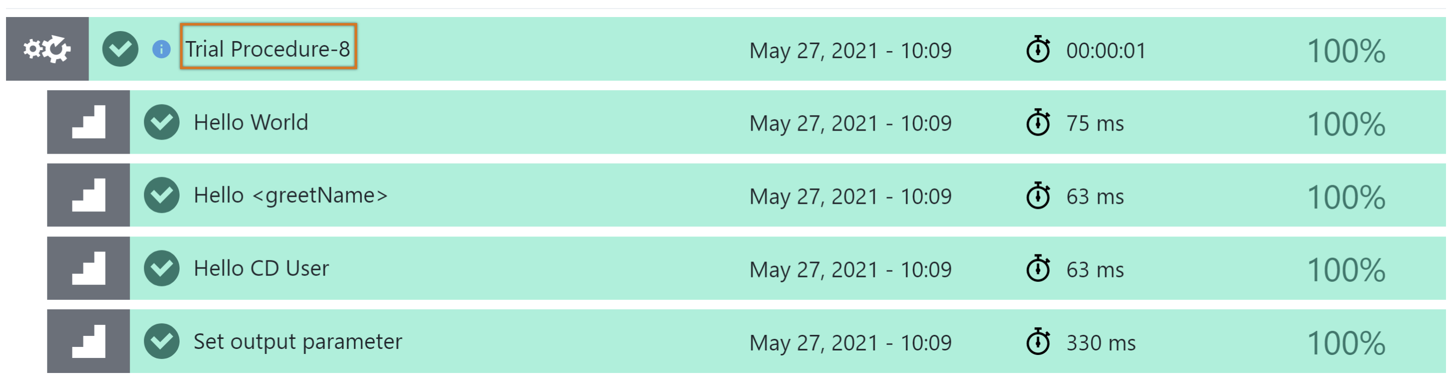
Create a step condition

1. Navigate to the procedure editor using the breadcrumb.
2. Open the first step, *Hello World*, and select the **Condition** tab on the upper right.
3. Add the following **Run Condition** (not **Precondition**) expression so that the step runs only after the procedure has been run five times:

   ```
   $[/javascript myProcedure.count > 5]
   ```

   Select **OK**.
4. Run the procedure. Use the **Last Run** option.
5. Select **Rerun** a few more times.

   Note that the *Hello World* step runs after the count property exceeds five. Here is an example of the results after eight runs.



Trial Procedure-8	May 27, 2021 - 10:09	00:00:01	100%
Hello World	May 27, 2021 - 10:09	75 ms	100%
Hello <greetName>	May 27, 2021 - 10:09	63 ms	100%
Hello CD User	May 27, 2021 - 10:09	63 ms	100%
Set output parameter	May 27, 2021 - 10:09	330 ms	100%

Example 2: Creating a service catalog item ➞