Commands & Scripts [On-Prem]

What are Commands & Scripts?

Commands and Scripts are customizable sets of instructions that you can create to execute specific tasks on your machines. They can range from single commands to intricate scripts written in Python, Bash, PowerShell, and more. The sky is the limit for the possibilities you can unlock with automated execution of commands and scripts.

Supported Scripts

- Bash
- Batch
- PowerShell
- Python
- VBScript

Local execution vs Remote execution

Commands and scripts can be executed in two different modes: local execution and remote execution.

- **Local Execution:** Commands and script files are executed on your local machine. When triggered in a workflow, the On-Prem Agent accesses the script/command from the local machine and runs it. This functionality is perfect for tasks within the local environment.
- **Remote Execution:** Here, commands and script files are executed on a remote machine connected to a private network. Upon trigger, the On-Prem Agent executes the command or script on the specified remote machines. This is beneficial for centralized control and management in an organization.

Common use-cases

With the addition of Commands and Scripts to your workflows, you can automate various processes, including:

- **Data processing operations**, such as file conversions, data transformations, and quality checks.
- **File and directory operations**, such as automating backup processes, FTP file sharing, renaming, moving, or deleting files and directories in bulk.
- **Sending notifications and alerts** to remote machines based on triggers or events.
- **Network operations** such as mapping network drives, querying network status or configuration.

Automating command and script executions

To set up a command or script action into your flow, you need to follow the steps below:

1. Install the On-Prem Agent
2. Add your commands or scripts to the agent's dedicated file
3. Create a Commands & Scripts app connection
4. Configure your action
5. Switch on the flow

These steps are explained in detail below:

1. Install the On-Prem Agent

Learn how to create a new on-prem agent

2. Add your commands or scripts to the dedicated file

The Commands & Scripts tool reads your commands and scripts from a file called *script.properties* in the On-Prem Agent's installed directory. After adding your commands or scripts, they appear as dropdown options during action configuration.

Adding a command or script to be executed locally

To add a command or script to the designated file:

1. Navigate to the agent's directory and locate the scripts.properties file *(...../FlowAgent/user/script/script.properties)*
2. Add your commands/scripts using the following parameters:

   name: The display name of the command to be shown when configuring the action

   working_directory: The full path of the folder where the command or script should be executed

   arguments: The input arguments that you want to pass to the command or script from your workflow. These parameters will appear as fields when you configure your action. Arguments must be separated by commas. Append 'mandatory' before an argument value. Example: mandatory:source, destination

   command: The command you want to execute. The command should be reframed with a $ prefix for the arguments. For example, cp /home/sourcefile /home/destination must be written as cp $source $destination

   or, if you are configuring a script,

script: The absolute path of the script file's location

timeout_in_seconds: The duration for the command or script to finish executing, after which the process will be terminated. Maximum timeout = 120 seconds.

output_required: Whether you want the output to be recorded and displayed in the flow's history. If set to false, only the status of the execution will be shown.

output_local_save: If set to true, a log file will be created under user/script/output and the output will be added to the log file. If an output exceeds the limit of 10000 characters, a truncated output will be displayed in the flow's history. Full output can be viewed in the log file if this option is set to true.

An example of how a command is added:

```
1.name=List
1.working_directory=/Users/Zylker-08/
1.arguments=mandatory:directory
1.command=ls -l $directory
1.output_required=true
```

An example of how a script is added:

```
12.name=HTML
12.working_directory=/Users/Zylker-08/script/
12.arguments=mandatory:content
12.script=/Users/Zylker-08/script/html.vbs
```

Important notes:

- You need to add a number and a period as a prefix to all parameters of a command/script to map them together using unique identifiers. Do not use the same prefix for different commands/scripts. However, the prefix need not be in any particular order.
- Once you've used a command/script in a flow, do not change the prefix of that particular command/script's parameters.
- The maximum timeout you can set for a script execution is 120 seconds, over which the execution will be terminated automatically.
- You need to provide all the necessary permissions to execute your scripts in your machine.

Adding a command or script to be executed remotely

You can follow the same steps above to add a command or script that is intended to be run remotely. However, the parameter, **timeout_in_seconds**, is not available for Remote Commands & Scripts actions.

You also need to allow-list the host machine by adding the "host" parameter. **The host field is mandatory**. The following illustration shows how you can use the host parameter to provide or revoke access to different machines in the same network:

- Add host=1.2.2.2 for the command to be listed when configuring actions for the host 1.2.2.2 alone.
- Add host=*, for the command to be listed for all remote hosts.
- Add host=*/1.2.2.2, for the command to be listed when configuring actions for all host machines except 1.2.2.2.

An example of how a command is added for remote execution:

1.name=List
1.working_directory=/Users/Zylker-12/script
1.arguments=mandatory:source,mandatory:destination
1.command=cp$source$destination
1.host=1.1.1.1
1.output_required=true

An example of how a script can be added:

12.name=CreateFolder
12.working_directory=/Users/Zylker-12/script/batch
12.arguments=mandatory:FolderName
12.script=/Users/Zylker-12/script/batch/CreateFolder.bat
12.host=1.1.1.1

Important notes:

- You need to add a number and a period as a prefix to all parameters of a command/script to map them together using unique identifiers. Do not use the same prefix for different commands/scripts. However, the prefix need not be in any particular order.
- Once you've used a command/script in a flow, do not change the prefix of that particular command/script's parameters.
- The maximum timeout you can set for a script execution is 120 seconds, over which the execution will be terminated automatically.
- You need to provide all the necessary permissions to execute your scripts in your machine.

3. Restart the agent

To restart the agent:

1. Navigate to the terminal or command prompt window tab where the agent is currently running.
2. Stop the agent by pressing Ctrl+C (Windows/Linux) or Control+C (macOS).
3. Restart the agent by entering startagent.bat (Windows) or sh startagent.sh (macOS/Linux).

4. Create a Commands & Scripts connection

You need to create separate connections for local and remote command/script executions. However, the agent used to create these connections can be the same.

Creating Commands & Scripts(Local) connection
To create a new connection, follow these steps:

1. Drag and drop the required action to your flow.
2. If there are no other Commands & Scripts connections in your account, click **Connect**. Otherwise, click **+New**.
   Alternatively, you can create a new connection by navigating to **Settings**, then clicking **Connections**. Click **Create connection** and choose Commands & Scripts.
3. Choose an on-prem agent that you've already created.
4. Enter a connection name.
5. In the case of Commands & Scripts (Remote), you'll also need to configure the following fields:
   Host: The name of the machine where you want the commands/scripts to be executed.
   Username: The username of the target system where you intend to execute commands or scripts.
   Password: The password of the target system where you intend to execute commands or scripts. Ensure you have appropriate permissions to access and execute commands on this machine.
   Port: The port through which your primary machine accesses the remote machine. Usually, the port is 22.
6. Click **Authorize**.

5. Configure the action

To configure the action, you need to:

1. Create a new flow.
2. Navigate to **Logic** on the left panel.
3. Drag and drop a Commands & Scripts action based on your particular requirement such as the script language, and whether you want to run it in your local machine or want to run it remotely.
4. While configuring the action, choose the connection that you created in the earlier step from the connection dropdown.

5. Choose the command or script that you want to execute from the **Commands/Scripts** dropdown.
   Note: For remote execution actions, only the commands and scripts allow-listed for the respective hosts will be displayed.
6. Once you choose the command or script, the arguments configured in the *scripts.properties* file associated with the selected command or script will be displayed as separate fields. Map them with the relevant variables from the previous steps or manually type in the values.
7. Click **Done**.
8. Test the flow once, and switch it on.

Actions available in Zoho Flow

Commands & Scripts

Actions (both local and remote executions)

Execute command
Use this action to execute single line commands, such as list, copy, and move.

Execute PowerShell script
Execute VBScript
Execute Bash script
Execute batch script
Execute Python script
Use these actions to perform database backups, monitor system logs, monitor network traffic, automate software updates, manage user groups, create and modify users in Active Directory, perform web scraping, and much more.

Common Problems

My script is not getting executed. (Error 13 Permission denied)
Before running a script, provide all required permissions to execute the script.

I am getting an error "Error 2 No such file/directory"
Check if the file exists. Also check if script name is properly specified in the *script.properties* file.
If you are specifying only the relative working directory, add ./ for the path (for eg, ./startagent.sh) or specify the full(absolute) path in file_path arguments.

I am getting an error, "Arguments not listed"
Make sure parameter grouping(numbering) is done properly. For example, if you're using 12.arguments for the first script, and use 12.arguments again for a different script, the agent groups both parameters as part of the script that's indexed with the number 12.

When performing remote script execution in macOS, I'm getting an error, "Operation not permitted"

Check file and folder access permissions. You need to set executable permissions for both the script and its folder.

I'm facing problems with using host name when executing remotely

You can use the machine's IP address instead of the host name. While using IP, if DHCP is configured you need to reconnect using the new IP.

I'm getting the error, "Connection refused", when executing a command remotely in macOS.

Navigate to System Settings > Sharing. Check if **remote login** is **enabled**. If not, enable it and try executing the command again.