

# Projects

A *project* is a place to separate management activity. All Rundeck activities occur within the context of a project. Multiple projects can be maintained on the same Rundeck server. Projects are independent from one another, so you can use them to organize unrelated systems within a single Rundeck installation. This can be useful for managing different teams, infrastructures, environments or applications.

Projects can be [archived](#) and imported to other Rundeck servers to help promote changes between environments.

A new installation will not contain any projects so Rundeck will present you with a dialog to create one.

## Creating A Project

Creating a New Project: Setting the name



Create a new Project

Details Execution History Clean Execution Mode User Interface Default Node Executor Default File Copier

**Project Name**

**Label**

**Description**

Cancel Create

In the Execution History Clean Settings, you can set the length of time you would like to keep execution history:

Create a new Project

DetailsExecution History CleanExecution ModeUser InterfaceDefault Node ExecutorDefault File Copier

**Execution History Clean:**

☒ **Enable**  
Enable cleaner executions history

Days to keep executions. Default: 60

Minimum executions to keep. Default: 50

Maximum size of the deletion. Default: 500

Schedule clean history job (Cron expression). Default: 0 0 0 1/1 \* ? \* (Every days on 12:00 AM)

-choose an example-

Ranges: 1-3 . Lists: 1,4,6 . Increments: 0/15 "every 15 units starting at 0".  
See: Cron reference for formatting help

CancelCreate

Jobs are set to run automatically, this can be disabled as well as scheduling jobs can be disabled via the Execution Mode tab:

Create a new Project

DetailsExecution History CleanExecution ModeUser InterfaceDefault Node ExecutorDefault File Copier

**Execution Mode**

Job execution and schedule configuration at project level.

Enable/Disable Execution Now

☐ Disable Execution

☐ Disable Schedule

CancelCreate

The user interface setting within the project allows for view settings such as the Job Group Expansion Level, Showing the Project Readme/Homepage, and the MOTD

Create a new Project

DetailsExecution History CleanExecution ModeUser InterfaceDefault Node ExecutorDefault File Copier

**User Interface**

Additional configuration for the user interface for this project

Job Group Expansion Level

1

In the Jobs page, expand Job groups to this depth by default. [Less](#)

- 0 : collapse all Groups
- 1 : expand all Groups.

Display the Project Readme

☐ Projects List

☐ Project Home Page

Display the Project MOTD

☐ Projects List

☐ Project Home Page

☐ Navbar Indicator

☐ Allow Unsanitized HTML output

[Less](#)

If enabled, and a command's log filter specifies the `no-strip` meta tag, the output from the command will not be run through the tag sanitizer. This option only works if the framework level property is also enabled.

CancelCreate

Node Executor settings deal with how commands are executed on remote nodes, note there is a local setting to run rundeck strictly on the local machine, and the script execution setting which delegates command execution to an external script:

Create a new Project

Details Execution History Clean Execution Mode User Interface Default Node Executor Default File Copier

The Node Executor is responsible for executing commands and scripts on remote nodes.

SSH Executes a command on a remote node via SSH.

- SSH
- Local
- SSHJ-SSH
- Stub
- Script Execution
- Ansible Ad-Hoc Node Executor
- openssh / executor
- WinRM Node Executor Python

SSH Key Passphrase Storage Path

File Path to the SSH Key to use

Path to the SSH Key to use within Rundeck Storage. E.g. "keys/path/key1.pem"

Path to the Password to use within Rundeck Storage. E.g. "keys/path/my.password". Can be overridden by a Node attribute named 'ssh-password-storage-path'.

Path to the key's Passphrase to use within Rundeck Storage. E.g. "keys/path/my.password". Can be overridden by a Node attribute named 'ssh-key-passphrase-storage-path'.

SSH Authentication

privateKey

Type of SSH Authentication to use

Force PTY

Always force use of new pty

Connection Timeout

0

SSH Connection timeout in milliseconds. (0 for no timeout)

Command Timeout

0

Maximum duration of commands: timeout in milliseconds. (0 for no timeout)

The File Copier is responsible for copying scripts to the remote nodes before they are executed, this means the scripts will be copied to the remote machine before being run (note the default locations for any files being copied is c:\windows\temp on windows systems and /tmp on linux based systems:

Create a new Project

Details Execution History Clean Execution Mode User Interface Default Node Executor Default File Copier

The File Copier is responsible for copying scripts as files to remote nodes before they are executed.

SCP Copies a script file to a remote node via SCP.

- SCP
- Script Execution
- Ansible File Copier
- Stub
- SSHJ-SCP
- WinRM Python File Copier
- openssh / file-copier

File Path

File Path to the SSH Key to use

Path to the SSH Key to use within Rundeck Storage. E.g. "keys/path/key1.pem"

Path to the Password to use within Rundeck Storage. E.g. "keys/path/my.password". Can be overridden by a Node attribute named 'ssh-password-storage-path'.

Path to the key's Passphrase to use within Rundeck Storage. E.g. "keys/path/my.password". Can be overridden by a Node attribute named 'ssh-key-passphrase-storage-path'.

SSH Authentication

privateKey

Type of SSH Authentication to use

SSH Key Passphrase Storage Path

Bind Address

Set a bind address from the server

Cancel Create

## Git Configuration with Projects

Note with SCM Configurations there is an ability to both export jobs to a Git Repository and an Import setting for projects: Configuring these

Setup SCM

[Enable or configure SCM integration.](#)

**SCM Export**

SCM Export Plugins can export Job definitions to an external SCM Repo

**Git Export**

Export Jobs to a Git Repository

[Setup](#)

**SCM Import**

SCM Import Plugins can import Job definitions from an external SCM Repo

**Git Import**

Import Jobs from a Git Repository

[Setup](#)

## Archiving a Project

Maybe a project needs to be archived. When creating an archive, there are some settings that can be modified to select just what is needed on the archive. Some of these include:  
Jobs/Executions/Configuration/Readme/MOTD/ACL Policies/SCM Configuration/Webhooks

Project: [Redacted]

Include ☒ All

- ☒ Jobs
- ☒ Executions
- ☒ Configuration
- ☒ Readme/Motd
- ☒ ACL Policies
- ☒ SCM configuration
- ☒ Webhooks

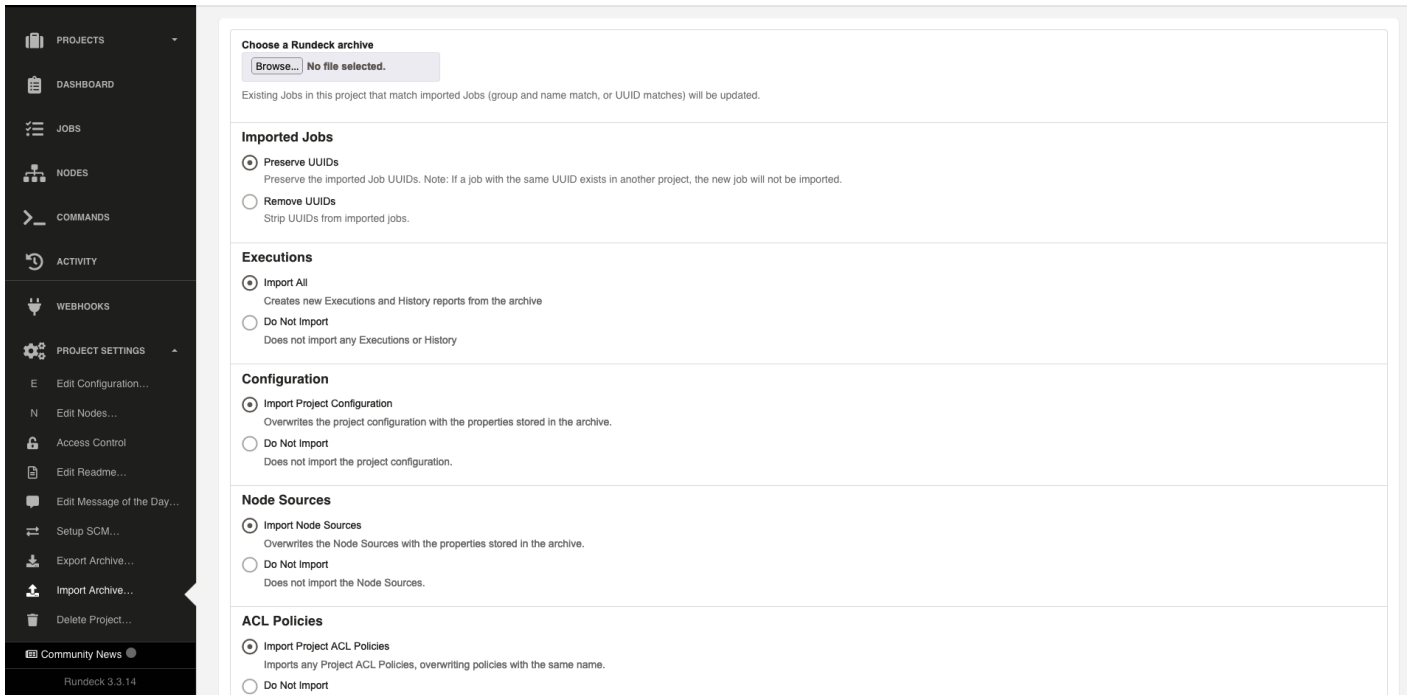
Referenced Jobs: ☐ Do not modify referenced jobs at export.  
☐ Strip Names. If possible, use only the UUID on referenced jobs.  
☐ Strip UUID. If possible, use only the group/name on referenced jobs.

**Webhooks**

☐ Include Webhook Auth Tokens  
If not included, tokens will be regenerated upon import.

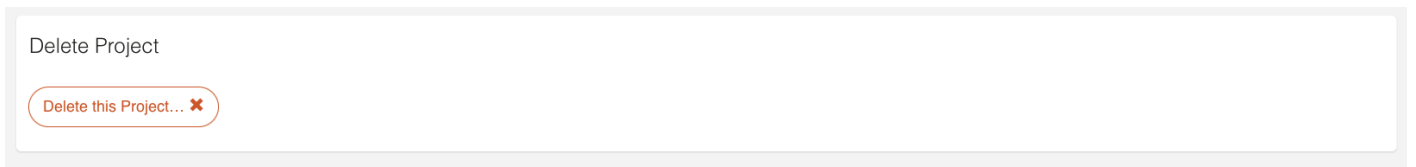
[Cancel](#) [Export Archive](#) [Export To Another Instance...](#)

On the other side of this Importing an Archive can also be done. Some settings to note when importing include: Preserving UUIDs on the imported Jobs, Importing Executions and History, The Project Configuration, Node Sources, ACL Policies, SCM Configuration, Referenced Job Validations, and Webhooks



## Deleting a Project

Maybe you are done with the project, deleting a project can be done via the GUI via:



### Revision #3

Created 22 September 2021 00:40:10 by jmoore53

Updated 30 November 2021 04:05:35 by jmoore53