

Building a matrix project

Multi-Configuration Projects

A **multi-configuration project** is useful for instances where your builds will make many similar build steps, and you would otherwise be duplicating steps.

As of 1.561, in [Matrix Project Plugin](#).

Configuration matrix

The **Configuration Matrix** allows you to specify what steps to duplicate, and create a multiple-axis graph of the type of builds to create.

For example, let us say that we have a build that we want to create for several different targets `alpha`, `beta`, and we want to produce both `debug` and `release` outputs. With a freestyle job, we would have the following ant commands:

```
Ant: set-target-alpha debug compile
Ant: set-target-beta debug compile
Ant: set-target-gamma debug compile
Ant: set-target-alpha release compile
Ant: set-target-beta release compile
Ant: set-target-gamma release compile
```

We can reduce this to one ant target using variables:

```
Ant: $target $releasetype compile
```

And by adding two **User Defined Axis** in the **Configuration Matrix**:

Name: `target`

Value: `set-target-alpha set-target-beta set-target-gamma`

Name: `releasetype`

Value: `debug release`

The **Names** `target` and `releasetype` are exposed to the ant scripts as environment variables. The **Value** is a space delimited list of points to iterate through for each axis.

The immediate upshot is that if we have another release type to add (for example, `debug-optimized`), we can simply add the value to the `releasetype` user defined axis rather. The single change will add configuration to build the `debug-optimized` release type against all three values of `target`.

Misc

NOTE: you have to choose "**Build multi-configuration project**" when creating a project, it **can not** be changed later. If you skip this step, you will be very confused and not get very far.

Some links explaining this feature a bit more:

- [Experience with Hudson - Building matrix project](#)
- [Hudson 1.122 and build/test matrix](#)
- [More old info on wiki](#)
- [Automated Builds using Windows -- New](#)

Newbie Note: Each configuration is akin to an individual job. It has its own build history, logs, environment, etc. The history of your multi-config job only shows you a list of configurations executed. You have to drill into each configuration to see the history and console logs.

Executors used by a multi-configuration project

Reference: [Re: Will a multi-configuration / matrix job always use up one executor on the master?](#)

A matrix build project (that uses the Slave axis) will use one additional executor ("master executor"?) on a random node to coordinate the executions on the nodes defined by the Slave axis.

This executor is added as a *temporary* executor(also called as "flyweight task") to the node and does not use up a configured executor slot on this node (this behavior is controlled by `hudson.model.Hudson.flyweightSupport`). Only this "master executor" or "flyweight task" will be affected by the "Restrict where this project can be run" option, under Advanced Project Options. "Actual" or non-flyweight build execution can be controlled with the "Slaves" axis that can be added under the project's Configuration Matrix: individual nodes and/or labels containing multiple nodes can be selected, as well as filtering axis combinations.

You can also control which nodes can run flyweight tasks using the [Exclude flyweight tasks](#) plugin, or you can pin the flyweight executor to a specific node using the [Matrix Tie Parent Plugin](#)

Configuration Matrix