

Global Variable String Parameter Plugin

Plugin Information

View Global Variable String Parameter [on the plugin site](#) for more information.

Global Variable String Parameter Plugin

Provides a parameter with support for global node properties via \$VARIABLE or \${VARIABLE}

Utilization

After installing, in your job choose Add Parameter and select Global Variable String Parameter. Use \$VARIABLE or \${VARIABLE} in either the default value or on the build now page to substitute global node parameters.

Version History

Version 1.2 (Sept 17 2013)

- Fix race condition where concurrent builds could cause variables not to substitute

Version 1.1 (Nov 02 2012)

- Initial Release

Example

1. Create a global variable

Global properties

Environment variables

List of key-value pairs

name

value

Tool Locations

2. Create a global variable string parameter in a job

This build is parameterized

Disable Build (No build is performed, project is re-enabled.)

Execute concurrently

Advanced Project Configuration

- Boolean Value
- Choice
- Extended Choice Parameter
- File Parameter
- Global Variable String Parameter**
- List Subversion tags (and more)
- Password Parameter
- Run Parameter
- String Parameter
- Text Parameter

Source Code Management

CVS

None

3. Reference the global variable in the default value

Global Variable String Parameter

Name

Default Value

Description

4. Add a simple echo statement to demonstrate the substitution

Execute shell

Command

[See the list of available environment variables](#)

5. The variable remains unsubstituted on the Build Now page

Project test2

This build requires parameters:

Test2

6. But is correctly substituted in the console output

```
http://cydwin.com/cydwin-ug-net/using.html#using-params  
+ echo Substitution 'Success!'  
Substitution Success!  
Finished: SUCCESS
```

7. The variable can also be changed from the build now screen to be incorrect

Project test2

This build requires parameters:

Test2

8. And no longer substitutes

```
+ echo '${THIS' 'WON'\''T' 'WORK}'  
${THIS WON'T WORK}  
Finished: SUCCESS
```