Jenkins Configuration as Code Reference

configuration-as-code © RC	Deprecation
— deprecated	Possible values:
	1 OSSIDIE Values.
	• <u>reject</u>
	• <u>warn</u>
restricted	Restriction
	Possible values:
	• <u>reject</u>
	• <u>beta</u>
	• <u>warn</u>
unknown	Unknown
	Possible values:
	• <u>reject</u>
	• <u>warn</u>
– version	Version
version	Possible values:
	• ONE
	<u> </u>
1 . 1 A ROOT ELEMENT	
redentials ROOT ELEMENT folder	FolderCredentialsProvider
– providerImpl	
_	SystemCredentialsProvider
– user	UserCredentialsProvider
1 · A ROOT ELEMENT	
enkins ® ROOT ELEMENT	
- agentProtocols	
 authorizationStrategy 	AuthorizationStrategy Possible values:
	Possible values:
	• <u>unsecured</u>
	• <u>legacy</u>
	<u>loggedInUsersCanDoAnything</u><u>globalMatrix</u>
	• projectMatrix
	list of Cloud
– clouds	
crumbIssuer	
	Possible values:

disableRememberMe boolean disabledAdministrativeMonitors ------ list of String globalNodeProperties ----list of NodeProperty Possible values: disableDeferredWipeout authorizationMatrix <u>envVars</u> <u>envInject</u> toolLocation labelAtoms -----list of LabelAtom String labelString markupFormatter -----MarkupFormatter In such places as project description, user description, view description, and build description, Jenkins allows users to enter some free-form text that describes something. This configuration determines how such free-form text is converted to HTML. By default, Jenkins treats the text as HTML and use it as-is unmodified (and this is default mainly because of the backward compatibility.) While this is convenient and people often use it to load <iframe>, <script>. and so on to mash up data from other sources, this capability enables malicious users to mount XSS attacks. If the risk outweighs the benefit, install additional markup formatter plugins and use them. Possible values: • plainText • rawHtml mode Mode Possible values: • NORMAL **EXCLUSIVE** MyViewsTabBar myViewsTabBar Possible values: standard noUsageStatistics -----Boolean String nodeName nodeProperties list of NodeProperty Possible values: disableDeferredWipeout <u>authorizationMatrix</u> <u>envVars</u> <u>envInject</u> • toolLocation nodes list of Node Possible values:

• jenkins

• permanent

numExecutors -----int primaryView -----View Possible values: • <u>all</u> • <u>buildPipeline</u> dashboard <u>proxy</u> • <u>list</u> <u>multiJob</u> • myView ProjectNamingStrategy projectNamingStrategy Possible values: • standard • pattern ------ ProxyConfiguration proxy quietPeriod ----- Integer remotingSecurity ----- AdminWhitelistRule scmCheckoutRetryCount -----int securityRealm ----- SecurityRealm Possible values: • activeDirectory • <u>legacy</u> local • ldap • <u>pam</u> • none slaveAgentPort ----int systemMessage ----- String updateCenter ----- UpdateCenter ----- list of View views Possible values: • <u>all</u> • <u>buildPipeline</u> • dashboard • proxy <u>list</u> • multiJob • myView

https://ci.my4030.ir/configuration-as-code/reference

ViewsTabBar viewsTabBar Possible values:

• standard

$global Credentials Configuration \ ^{\textcircled{\tiny{POOT}\ ELEMENT}}$

configuration ----- Configuration

security ROOT ELEMENT

_	apiToken	ApiTokenPropertyConfiguration
_	copyartifact	CopyArtifactConfiguration
_	crumb	GlobalCrumbIssuerConfiguration
_	envInjectPluginConfiguration	EnvInjectPluginConfiguration
_	masterKillSwitchConfiguration	MasterKillSwitchConfiguration
_	queueItemAuthenticator	${\tt QueueItemAuthenticatorConfiguration}$
_	sSHD	SSHD
_	scriptApproval	ScriptApproval

 $update Site Warnings Configuration \\ ----- \\ \texttt{UpdateSiteWarningsConfiguration}$

unclassified $^{\textcircled{ROOT}}$ ROOT ELEMENT

-	administrative Monitors Configuration	AdministrativeMonitorsConfiguration
-	ansiColorBuildWrapper	DescriptorImpl
	artifactManager	ArtifactManagerConfiguration
	artifactoryBuilder	DescriptorImpl
	buildDiscarders	GlobalBuildDiscarderConfiguration
-	buildStepOperation	DescriptorImpl
-	builtInNode	MasterBuildConfiguration
-	cVSSCM	DescriptorImpl
-	casCGlobalConfig	CasCGlobalConfig
-	defaultFolderConfiguration	AbstractFolderConfiguration
-	defaultView	GlobalDefaultViewConfiguration
-	email-ext	ExtendedEmailPublisherDescriptor
-	envInjectNodeProperty	EnvInjectNodePropertyDescriptor
-	envVarsFilter	EnvVarsFilterGlobalConfiguration
-	fingerprints	GlobalFingerprintConfiguration
-	gitHubConfiguration	GitHubConfiguration
-	gitHubPluginConfig	GitHubPluginConfig
-	gitParameter	DescriptorImpl

gitSCM	DescriptorImpl
globalConfigFiles	GlobalConfigFiles
globalDefaultFlowDurabilityLevel	DescriptorImpl
globalLibraries	GlobalLibraries
- ivyBuildTrigger	DescriptorImpl
- ivyModuleSet	DescriptorImpl
- junitTestResultStorage	JunitTestResultStorageConfiguration
- location	JenkinsLocationConfiguration
lockableResourcesManager	LockableResourcesManager
— mailer	DescriptorImpl
mavenModuleSet	DescriptorImpl
myView	GlobalConfigurationImpl
nodeProperties	GlobalNodePropertiesConfiguration
phaseJobsConfig	DescriptorImpl
pipeline-model-docker	GlobalConfig
plugin	GlobalPluginConfiguration
pollSCM	DescriptorImpl
projectNamingStrategy	GlobalProjectNamingStrategyConfiguration
quietPeriod	GlobalQuietPeriodConfiguration
rebuildDescriptor	RebuildDescriptor
resourceRoot	ResourceDomainConfiguration
rocketChatNotifier	DescriptorImpl
sSHBuildWrapper	DescriptorImpl
- scmRetryCount	GlobalSCMRetryCountConfiguration
- shell	DescriptorImpl
simple-theme-plugin	SimpleThemeDecorator
singleConditionalBuilder	SingleConditionalBuilderDescriptor
- subversionSCM	DescriptorImpl
throttleJobProperty	DescriptorImpl
timestamper	TimestamperConfig
upstream	DescriptorImpl
usageStatistics	UsageStatistics
viewsTabBar	GlobalConfigurationImpl
warningsParsers	ParserConfiguration
- warningsPlugin	WarningsPluginConfiguration

tool ROOT ELEMENT
ant DescriptorImpl

F	dockerTool	DescriptorImpl
L	git	DescriptorImpl
F	gradle	DescriptorImpl
F	jdk	DescriptorImpl
F	jgit	DescriptorImpl
F	jgitapache	DescriptorImpl
F	maven	DescriptorImpl
F	mavenGlobalConfig	GlobalMavenConfig
F	mercurialInstallation	DescriptorImpl
F	nodejs	DescriptorImpl

providerImpl ® ROOT ELEMENT

$system Credentials Provider \ ^{\texttt{CONFIGURATOR}}$

domainCredentials -----list of DomainCredentials

$domain Credentials \ ^{\text{CONFIGURATOR}}$

redentials

Possible values:

usernamePassword
gitHubApp
basicSSHUserPrivateKey
file
string
x509ClientCert
certificate

----- Domain

$username Password \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>Credentials</u> Username with password

domain

L	description	String
-	id	String
-	password	String The password.
_	scope	CredentialsScope Possible values:
		SYSTEMGLOBALUSER
	username	String The username.

usernameSecret

boolean

Historically, not only passwords but usernames were masked in the build log. Since these can interfere with diagnostics, and cause unrelated occurrences of a common word to be masked, you may choose to leave usernames unmasked if they are not sensitive. Note that regardless of this setting, the username will be displayed to anyone permitted to reconfigure the credentials.

$git Hub App \ ^{\text{CONFIGURATOR}}$

Implementation of <u>Credentials</u>
GitHub App

GitHub App	
apiUri	String
	<pre>GitHub API endpoint such as https://github.example.com/api/v3/.</pre>
_ appID	String
	GitHub app ID, can be found on the App's settings, on the General page in the About section
- description	String
_ id	String
– owner	String
	The organisation or user that this app is to be used for. Only required if this app is installed to multiple organisations.
– privateKey	Secret
	Private key for authenticating to GitHub with, it must be in PKCS#8 format, GitHub will give it to you in PKCS#1.
	You can convert it with openssl pkcs8 -topk8 -inform PEM -outform PEM -in current-
	key.pem -out new-key.pem -nocrypt
– scope	CredentialsScope
	Possible values:
	• SYSTEM
	• GLOBAL
	• <u>USER</u>

$basic SSHUser Private Key \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>Credentials</u> SSH Username with private key

description String

id String

passphrase String

privateKeySource PrivateKeySource
Possible values:

directEntry

scope CredentialsScope
Possible values:

- SYSTEM
- GLOBAL
- USER

username Secret String boolean

By default, usernames are not masked in the build log. You may choose to mask a credential's username if you consider it to be sensitive information. However, this can interfere with diagnostics. For example, if the username is a common word it will cause unrelated occurrences of that word to also be masked.

Regardless of this setting, the username will be displayed in the selection dropdown to anyone permitted to reconfigure the credentials.

directEntry CONFIGURATOR

Implementation of <u>PrivateKeySource</u>

Enter directly

privateKey ----- Secret

file CONFIGURATOR

Implementation of Credentials

Secret file

.iei	. IIIe	
-	description	String
F	fileName	String
F	id	String
-	scope	CredentialsScope
		Possible values:
		SYSTEMGLOBALUSER
	secretBytes	SecretBytes

string CONFIGURATOR

Implementation of Credentials

Secret text



x509ClientCert CONFIGURATOR

Implementation of <u>Credentials</u> X.509 Client Certificate

clientCertificateString

- clientKeySecret	Secret
description	String
- id	String
- scope	Possible values:
	SYSTEMGLOBALUSER
- serverCaCertificate	String

certificate CONFIGURATOR

Implementation of **Credentials**

Certificate

• <u>uploaded</u>

password ----- String

The password. The use of separate integrity and encryption passwords is not supported.

scope ----- CredentialsScope

Possible values:

- SYSTEM
- GLOBAL
- <u>USER</u>

uploaded CONFIGURATOR

Implementation of <u>KeyStoreSource</u>

Upload PKCS#12 certificate

- uploadedKeystore ----- SecretBytes

domain CONFIGURATOR

Possible values:

- <u>dockerServer</u>
- hostnameSpecification
- hostnamePortSpecification
- mavenServerIdSpecification
- pathSpecification
- schemeSpecification

dockerServer CONFIGURATOR

Implementation of **DomainSpecification**

Docker Server Credentials

hostnameSpecification CONFIGURATOR Implementation of DomainSpecification

Hostname

excludes	- String
	A comma separated list of excluded hostnames. (The * wildcard is permitted in hostnames, for
	<pre>example: jenkins-ci.org, *.jenkins-ci.org, jenkinsci.github.io.)</pre>
	The empty list implies no hostnames are excluded. The excludes list is processed after the includes
	list.
includes	- String
	A comma separated list of included hostnames. (The * wildcard is permitted in hostnames, for
	example: jenkins-ci.org, *.jenkins-ci.org, jenkinsci.github.io.)
	The empty list implies no hostnames are excluded. The excludes list is processed after the includes
	list.

 $hostnamePortSpecification \begin{tabular}{l} {\tt CONFIGURATOR} \\ {\tt Implementation of } {\tt \underline{DomainSpecification}} \\ {\tt Implementation of } {\tt \underline{DomainSpecification}} \\ {\tt Implementation of } {\tt \underline{DomainSpecification}} \\ {\tt \underline{DomainSpecification}}$

Hostname:port

– excludes	String
	A comma separated list of excluded hostnames. (The * wildcard is permitted in hostnames, for
	<pre>example: jenkins-ci.org:*, *.jenkins-ci.org:80, jenkinsci.github.io:443.)</pre>
	The empty list implies no <i>hostname:port</i> is excluded. The excludes list is processed after the includes
	list.
– includes	String
	A comma separated list of included hostnames. (The * wildcard is permitted in hostnames, for
	example: jenkins-ci.org:*, *.jenkins-ci.org:80, jenkinsci.github.io:443.)
	example: jenkins-ci.org:*, *.jenkins-ci.org:80, jenkinsci.github.io:443.) The empty list implies no <i>hostname:port</i> is excluded. The excludes list is processed after the includes

 $maven Server Id Specification \begin{tabular}{l} \label{table} {\tt CONFIGURATOR} \\ {\tt Implementation} \ of \begin{tabular}{l} \label{table} {\tt Domain Specification} \end{tabular}$ Maven serverId

excludes	- String
	A comma separated denylist of serverIds (the * wildcard is permitted in serverIds. For example:
	<pre>jenkins-ci.org, *.jenkins-ci.org, mycomp*)</pre>
	The empty list implies no serverIds are excluded. The excludes list is processed after the includes list.
– includes	- String
	A comma separated allowlist of serverIds (the * wildcard is permitted in serverIds. For example:
	jenkins-ci.org, *.jenkins-ci.org, mycomp*
	The empty list implies no serverIds are excluded. The excludes list is processed after the includes list.

pathSpecification CONFIGURATOR

Implementation of <u>DomainSpecification</u> URI path

caseSensitive	boolean
	Select this option to require that the paths match taking character case into account.
excludes	String

A comma separated list of excluded paths. (ANT style * and ** wildcards are permitted in paths, for example: /jenkins/github/*,/jenkins-ci/**/org,jenkinsci.github.io.)

The empty list implies no paths are excluded. The excludes list is processed after the includes list.

includes

A comma separated list of included paths. (ANT style * and ** wildcards are permitted in paths, for example: /jenkins/github/*,/jenkins-ci/**/org,jenkinsci.github.io.)

The empty list implies no paths are excluded. The excludes list is processed after the includes list.

schemeSpecification CONFIGURATOR

Implementation of **DomainSpecification** URI scheme

String schemes

A comma separated list of URI schemes (as defined in RFC-3986). For example: http, https, ssh, sftp, imap.

folderCredentialsProvider ROOT ELEMENT

userCredentialsProvider ® ROOT ELEMENT

unsecured CONFIGURATOR

Implementation of AuthorizationStrategy Anyone can do anything

legacy CONFIGURATOR

Implementation of <u>AuthorizationStrategy</u> Legacy mode

$logged In Users Can Do Anything\ ^{\text{CONFIGURATOR}}$

Implementation of <u>AuthorizationStrategy</u>

Logged-in users can do anything

allowAnonymousRead

If checked, this will allow users who are not authenticated to access Jenkins in a read-only mode.

$global Matrix \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

Implementation of AuthorizationStrategy

grantedPermissions list of String This attribute only exists for compatibility for Configuration as Code YAML files before Matrix Authorization Strategy Plugin 1.4. Use permissions instead. Setting both is unsupported and the result undefined.

permissions list of String

> A list of strings with the following format: "PermissionGroup/PermissionName:UserOrGroup". Examples for list entries:

- "Overall/Read:anonymous"
- "Job/Build:joe.chin"

projectMatrix CONFIGURATOR

Implementation of <u>AuthorizationStrategy</u>

grantedPermissions

This attribute only exists for compatibility for Configuration as Code YAML files before Matrix

Authorization Strategy Plugin 1.4. Use permissions instead. Setting both is unsupported and the result undefined.

permissions -----list of String

A list of strings with the following format: "PermissionGroup/PermissionName:UserOrGroup". Examples for list entries:

- "Overall/Read:anonymous"
- "Job/Build:joe.chin"

all CONFIGURATOR

Implementation of $\underline{\text{View}}$

All

descriptionnameString

properties -----list of ViewProperty

buildPipeline CONFIGURATOR

Implementation of <u>View</u> Build Pipeline View

— alwaysAllowManualTrigger ----- boolean

Select this option if you want to be able to execute again a successful pipeline step. If the build is parameterized, this will re-execute the step using the same parameter values that were used when it was previously executed.

buildCard ----- BuildCardExtension

Possible values:

• standardBuildCard

 $build View Title \\ ----- \\ String$

columnHeaders ----- PipelineHeaderExtension

Possible values:

- buildVariablesHeader
- parametersActionHeader
- simpleColumnHeader
- simpleRowHeader
- nullColumnHeader
- filteredVariablesHeader

consoleOutputLinkStyle ----- String

cssUrl ----- String

description ----- String

gridBuilder ----- ProjectGridBuilder

Possible values:

• downstream

name ----- String

noOfDisplayedBuilds ----- String

Select the number of build pipelines to display in the view.

L	
properties	list of ViewProperty
refreshFrequency	int
	Frequency at which the Build Pipeline Plugin updates the build cards in seconds
rowHeaders	PipelineHeaderExtension
	Possible values:
	• <u>buildVariablesHeader</u>
	• parametersActionHeader
	• <u>simpleColumnHeader</u>
	• <u>simpleRowHeader</u>
	• <u>nullColumnHeader</u>
	• <u>filteredVariablesHeader</u>
and a start of the	String
– selectedJob	Str ing
- showPipelineDefinitionHeader	boolean
	Select this option if you want to show the pipeline definition header in the pipeline view. If this
	option is not selected, then a pipeline that has never been run will not show any details about its jobs
	and appear like a blank form. Job details will only appear after the pipeline has been run at least once.
- showPipelineParameters	boolean
- showPipelineParametersInHeaders	boolean
triggerOnlyLatestJob	boolean
	Select this option to restrict the display of a Trigger button to only the most recent successful build
	pipelines.
	This option will also limit retries to just unsuccessful builds of the most recent build pipelines.
	Yes: Only the most recent successful builds displayed on the view will have a manual trigger button
	for the next build in the pipeline.
	<i>No:</i> All successful builds displayed on the view will have a manual trigger button for the next build in the pipeline.

$build Variables Header \ ^{\texttt{CONFIGURATOR}}$

Implementation of PipelineHeaderExtension All build variables and parameters

 $parameters Action Header \begin{tabular}{l} \tt CONFIGURATOR \\ Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ \end{tabular}$ Include parameters only

 $simple Column Header \begin{tabular}{l} \tt CONFIGURATOR \\ Lower Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ Lower Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ Lower Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ Lower Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ Lower Implementation of \begin{tabular}{l} \tt Pipeline Header Extension \\ Lower Implementation \\ Lower Implem$ Just the build name and number

$simple Row Header \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>PipelineHeaderExtension</u> Just the pipeline number

$null Column Header \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>PipelineHeaderExtension</u> No header

12/27/2021 $filtered Variables Header \ ^{\tt CONFIGURATOR}$ Implementation of PipelineHeaderExtension Specify which build variables filteredVariables -----list of FilteredVariable CONFIGURATOR variableName $standard Build Card \ ^{\texttt{CONFIGURATOR}}$ Implementation of BuildCardExtension Standard build card downstream CONFIGURATOR Implementation of **ProjectGridBuilder** Based on upstream/downstream relationship firstJob String Select the initial or parent Job in the build pipeline view. dashboard CONFIGURATOR Implementation of **View** Dashboard bottomPortlets list of DashboardPortlet Possible values: • statSlaves <u>buildPipelineDashboard</u> statBuilds <u>iframePortlet</u> <u>imagePortlet</u> hudsonStdJobsPortlet statJobs <u>jobsPortlet</u> latestBuilds <u>issuesChartPortlet</u> issuesTablePortlet testStatisticsChart **testStatisticsPortlet testTrendChart** unstableJobsPortlet list of ListViewColumn columns Possible values: • status weather <u>jobName</u> **lastSuccess** <u>lastFailure</u> <u>lastStable</u> lastDuration

builtOnColumn

buildButton

- gitBranchSpecifierColumn
- mercurialRevisionColumn

- statusColumn
- consoleColumn
- jobColumn
- lastDurationColumn
- <u>lastFailureColumn</u>
- lastSuccessColumn
- weatherColumn
- <u>itemColumn</u>
- <u>issuesTotalColumn</u>
- descriptionColumn
- branchStatusColumn

description	String
– hideJenkinsPanels	boolean
– includeRegex	String
- includeStdJobList	boolean

jobFilters

- list of ViewJobFilter

Possible values:

- gitHubBranchFilter
- gitHubPullRequestFilter
- statusFilter

_	jobNames	list of String
_	leftPortletWidth	String

leftPortlets ----- list of DashboardPortlet

Possible values:

- statSlaves
- <u>buildPipelineDashboard</u>
- statBuilds
- <u>iframePortlet</u>
- <u>imagePortlet</u>
- <u>hudsonStdJobsPortlet</u>
- statJobs
- jobsPortlet
- latestBuilds
- <u>issuesChartPortlet</u>
- <u>issuesTablePortlet</u>
- testStatisticsChart
- testStatisticsPortlet
- <u>testTrendChart</u>
- unstableJobsPortlet

-	name	String
_	properties	list of ViewProperty
_	recurse	boolean
_	rightPortletWidth	String
_	rightPortlets	list of DashboardPortlet

Possible values:

- statSlaves
- <u>buildPipelineDashboard</u>
- statBuilds
- iframePortlet
- imagePortlet
- <u>hudsonStdJobsPortlet</u>
- statJobs
- jobsPortlet
- latestBuilds
- <u>issuesChartPortlet</u>
- <u>issuesTablePortlet</u>
- testStatisticsChart
- testStatisticsPortlet
- testTrendChart
- unstableJobsPortlet

topPortlets -----list of DashboardPortlet

Possible values:

- statSlaves
- <u>buildPipelineDashboard</u>
- statBuilds
- iframePortlet
- imagePortlet
- hudsonStdJobsPortlet
- statJobs
- j<u>obsPortlet</u>
- latestBuilds
- issuesChartPortlet
- <u>issuesTablePortlet</u>
- testStatisticsChart
- testStatisticsPortlet
- testTrendChart
- unstableJobsPortlet

useCssStyle -----boolean

statSlaves CONFIGURATOR

 $Implementation \ of \ \underline{DashboardPortlet}$

Agent statistics

- name ---- String

$build Pipeline Dashboard \ ^{\tt CONFIGURATOR}$

Implementation of <u>DashboardPortlet</u> Build Pipeline Dashboard View

gridBuilder ----- ProjectGridBuilder

Possible values:

• <u>downstream</u>

2/27/2021		ht
name	String	
noOfDisplayedBuilds	String	
selectedJob	String	
statBuilds CONFIGURATOR Implementation of DashboardPortlet Build statistics		
name	String	
iframePortlet CONFIGURATOR Implementation of DashboardPortlet Iframe Portlet		
divStyle	String	
- iframeSource	String	
_ name	String	
imagePortlet CONFIGURATOR Implementation of DashboardPortlet Image		
imageUrl ·····	String	
name	String	
hudsonStdJobsPortlet CONFIGURATOR Implementation of DashboardPortlet Jenkins jobs list		
name	String	
StatJobs CONFIGURATOR Implementation of DashboardPortlet Job statistics name	String	
jobsPortlet CONFIGURATOR Implementation of DashboardPortlet Jobs Grid		
columnCount	int	
- fillColumnFirst	boolea	n
name	String	
latestBuilds CONFIGURATOR Implementation of DashboardPortlet Latest builds		
- name	String	
numBuilds	int	

 $issues Chart Portlet \ ^{\tt CONFIGURATOR}$

Implementation of <u>DashboardPortlet</u> Static analysis issues chart

	height	int
	– hideCleanJobs	- boolean
	name	String
	– selectTools	boolean
	– tools	list of ToolSelection
CON	FIGURATOR	
Too	lSelection	
	_ id	String

$issues Table Portlet \ ^{\tt CONFIGURATOR}$

Implementation of <u>DashboardPortlet</u> Static analysis issues per tool and job

hideCleanJobs boolean

name String

selectTools boolean

showIcons boolean

tools list of ToolSelection

$testStatisticsChart \ ^{\texttt{CONFIGURATOR}}$

Implementation of **DashboardPortlet**

Test Statistics Chart

— name ----- Strin

$testStatisticsPortlet \ ^{\texttt{CONFIGURATOR}}$

Implementation of **DashboardPortlet**

Test Statistics Grid

— failureColor	String
- hideZeroTestProjects	boolean
_ name	String
- skippedColor	String
- successColor	String
useAlternatePercentagesOnLimits	boolean
useBackgroundColors	boolean

$testTrendChart \ ^{\texttt{CONFIGURATOR}}$

Implementation of **DashboardPortlet**

Test Trend Chart

— displayStatus	DisplayStatus Possible values:
— dateShift	int
– dateRange	int

- ALL
 - **SUCCESS**

			SKIPPED FAILED
	graphHeight	int	
_	graphWidth	int	
L	name	Strir	ng

unstableJobsPortlet CONFIGURATOR

Implementation of **DashboardPortlet** Unstable Jobs

name showOnlyFailedJobs -----

status CONFIGURATOR

Implementation of ListViewColumn

weather CONFIGURATOR

Implementation of ListViewColumn Weather

jobName CONFIGURATOR

Implementation of ListViewColumn Name

lastSuccess CONFIGURATOR

Implementation of ListViewColumn Last Success

$lastFailure \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Last Failure

lastStable CONFIGURATOR

Implementation of <u>ListViewColumn</u> Last Stable

$last Duration \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Last Duration

buildButton CONFIGURATOR

Implementation of ListViewColumn **Build Button**

$builtOnColumn \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Built On

$git Branch Specifier Column \ ^{\text{CONFIGURATOR}}$

Implementation of <u>ListViewColumn</u>

Git Branches

12/27/2021

$mercurial Revision Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Mercurial Branch

$status Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn MultiJob - Status

$console Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn MultiJob - Console

jobColumn CONFIGURATOR

Implementation of ListViewColumn MultiJob - Job

lastDurationColumn CONFIGURATOR

Implementation of ListViewColumn MultiJob - Last Duration

$lastFailureColumn \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn MultiJob - Last Failure

lastSuccessColumn CONFIGURATOR

Implementation of ListViewColumn MultiJob - Last Success

weatherColumn CONFIGURATOR

Implementation of ListViewColumn MultiJob - Weather

$item Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Name

$issues Total Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn

Number of static analysis issues

		In order to simplify th
L	type	StatisticProperties
-	tools	list of ToolSelection
_	selectTools	boolean
-	name	String

In order to simplify the user interface you can select the property that will be compared with the specified threshold. Basically, three different properties are available:

Total

Selects the total number of issues in the current build.

New

Selects the total number of new issues in the current build with respect to the reference build.

Delta

Selects the difference of the total number of issues of the current build subtracted by the total number of issues in the reference build.

Additionally, you can select to choose all issues, or issues of a given severity only (error, high, normal or low).

Possible values:

- TOTAL
- TOTAL ERROR
- TOTAL HIGH
- TOTAL NORMAL
- TOTAL LOW
- NEW
- NEW ERROR
- NEW_HIGH
- NEW_NORMAL
- NEW LOW
- DELTA
- DELTA ERROR
- DELTA HIGH
- **DELTA NORMAL**
- DELTA LOW
- FIXED

$description Column \ ^{\text{CONFIGURATOR}}$

Implementation of <u>ListViewColumn</u> Project description

$branch Status Column \ ^{\text{CONFIGURATOR}}$

Implementation of ListViewColumn Status

gitHubBranchFilter CONFIGURATOR

Implementation of ViewJobFilter GitHub Branch Jobs Only

$git HubPullRequestFilter \ ^{\tt CONFIGURATOR}$

Implementation of ViewJobFilter GitHub Pull Request Jobs Only

statusFilter CONFIGURATOR Implementation of ViewJobFilter Status Filter

----- boolean statusFilter

proxy CONFIGURATOR

Implementation of View Include a global view

1			
ŀ	-	description	String
-	_	name	 String
-	_	properties	 list of ViewProperty
-	_	proxiedViewName	 String

list CONFIGURATOR

Implementation of View

List View

columns -----list of ListViewColumn

Possible values:

- status
- weather
- jobName
- lastSuccess
- lastFailure
- lastStable
- lastDuration
- buildButton
- builtOnColumn
- gitBranchSpecifierColumn
- mercurialRevisionColumn
- statusColumn
- consoleColumn
- jobColumn
- lastDurationColumn
- <u>lastFailureColumn</u>
- lastSuccessColumn
- weatherColumn
- itemColumn
- <u>issuesTotalColumn</u>
- descriptionColumn
- branchStatusColumn

description String includeRegex String

jobFilters -----list of ViewJobFilter

Possible values:

- gitHubBranchFilter
- gitHubPullRequestFilter
- statusFilter

jobNames -----list of String

name ----- String

properties ----- list of ViewProperty

recurse boolean

$multiJob \ ^{\text{CONFIGURATOR}}$

Implementation of <u>View</u>

MultiJob View

columns -----list of ListViewColumn

Possible values:

- status
- weather
- jobName
- lastSuccess
- <u>lastFailure</u>

- <u>lastStab</u>le
- lastDuration
- buildButton
- builtOnColumn
- gitBranchSpecifierColumn
- mercurialRevisionColumn
- statusColumn
- consoleColumn
- j<u>obColumn</u>
- lastDurationColumn
- <u>lastFailureColumn</u>
- lastSuccessColumn
- weatherColumn
- <u>itemColumn</u>
- issuesTotalColumn
- descriptionColumn
- branchStatusColumn

description
 includeRegex
 jobFilters
 Possible values:

- ossible values.
 - gitHubPullRequestFilter

• gitHubBranchFilter

• statusFilter

jobNames -----list of String

name ----- String

properties ----- list of ViewProperty

recurse -----boolean

myView CONFIGURATOR

Implementation of <u>View</u>

My View

CONFIGURATOR

Disable deferred wipeout on this node

$authorization Matrix Node Property \ ^{\text{CONFIGURATOR}}$

inheritanceStrategy ----- InheritanceStrategy

Possible values:

- <u>inheriting</u>
- <u>inheritingGlobal</u>
- <u>nonInheriting</u>

permissions

list of String

inheriting $^{\text{CONFIGURATOR}}$

Implementation of <u>InheritanceStrategy</u> Inherit permissions from parent ACL

$inheriting Global \ ^{\texttt{CONFIGURATOR}}$

Implementation of **InheritanceStrategy** Inherit globally defined permissions

$nonInheriting \ ^{\text{CONFIGURATOR}}$

Implementation of **InheritanceStrategy** Do not inherit permission grants from other ACLs

envVars CONFIGURATOR

Environment variables

----- list of Entry env

entry CONFIGURATOR

value

CONFIGURATOR

Prepare jobs environment

propertiesFilePath String

Specifies location of a property file with additional variables to be injected.

If you specify a relative path, then Jenkins will firstly try to resolve this path against the HOME directory of the slave/master user. If it fails to find a file there, then it will try resolution against the working directory of the node/slave. This behavior is a subject to change (see JENKINS-23666).

Warning! The variables specified by this file are being cached and refreshed only on node connection. If you modify this entry or the file contents for a Jenkins slave/agent, will need to reconnect this slave/agent to the master in order to get changes applied. If you modify settings for the Master node, you will have to restart the master.

unsetSystemVariables boolean

toolLocation CONFIGURATOR

Tool Locations

list of ToolLocation locations

toolLocation CONFIGURATOR

home

idk CONFIGURATOR

Implementation of **ToolInstallation**

JDK

home

name Strir	ng
propertieslist of	f ToolProperty
Possil	ble values:
•	installSource
propertieslist of	f ToolProperty
Possil	ble values:
•	installSource

installSource CONFIGURATOR

Install automatically

installers -----list of ToolInstaller

Possible values:

- <u>fromDocker</u>
- <u>zip</u>
- j<u>dkInstaller</u>
- <u>maven</u>
- antFromApache
- gradleInstaller
- nodeJSInstaller
- mirrorNodeJSInstaller
- batchFile
- command

installers -----list of ToolInstaller

Possible values:

- <u>fromDocker</u>
- <u>zip</u>
- jdkInstaller
- <u>maven</u>
- antFromApache
- gradleInstaller
- <u>nodeJSInstaller</u>
- mirrorNodeJSInstaller
- <u>batchFile</u>
- command

fromDocker CONFIGURATOR

Implementation of <u>ToolInstaller</u> Download from docker.com

- label String
- version String

A version of Docker to install. Tested values currently include:

- 17.09.1-ce
- 1.12.6
- 1.10.0

 latest (may not actually be the latest version even of Docker CE Stable! prefer an explicit version)

zip CONFIGURATOR

Implementation of <u>ToolInstaller</u>

Extract *.zip/*.tar.gz

	label	 String
_	subdir	 String

Optional subdirectory of the downloaded and unpacked archive to use as the tool's home directory.

url ----- String

URL from which to download the tool in binary form. Should be either a ZIP or a GZip-compressed TAR file. The timestamp on the server will be compared to the local version (if any) so you can publish updates easily. The URL must be accessible from the Jenkins controller but need not be accessible from agents.

jdkInstaller CONFIGURATOR

Implementation of **ToolInstaller**

Install Oracle Java SE Development Kit from the website

_	acc	ceptLi	cense	 	 	boolean
_	id					String

maven CONFIGURATOR

Implementation of <u>ToolInstaller</u>

Install from Apache

— id ----- String

$ant From Apache \ ^{\text{CONFIGURATOR}}$

Implementation of <u>ToolInstaller</u>

Install from Apache

- id ----- String

gradleInstaller CONFIGURATOR

Implementation of <u>ToolInstaller</u>

Install from Gradle.org

- id ----- String

$nodeJSInstaller\ {}^{\text{CONFIGURATOR}}$

Implementation of ToolInstaller

Install from nodejs.org

-	force32Bit	boolean
-	- id	String
-	npmPackages	String
-	npmPackagesRefreshHours	long

$mirror Node JS In staller \ ^{\texttt{CONFIGURATOR}}$

 $Implementation \ of \ \underline{ToolInstaller}$

Install from nodejs.org mirror

credentialsId String

Credentials for the nodejs.org mirror if required. Only basic authentication is supported.

 force32Bit
 boolean

 id
 String

 mirrorURL
 String

A nodejs.org mirror URL from where download a valid installable package of NodeJS in case there are limitations or restrictions and you are not able to reach the official NodeJS.org distribution site.

npmPackages Strin

batchFile CONFIGURATOR

Implementation of **ToolInstaller**

Run Batch Command

F	command	String
_	label	String
L	toolHome	String

command CONFIGURATOR

Implementation of **ToolInstaller**

Run Shell Command

command	String
label	String
toolHome	String

standard CONFIGURATOR

Implementation of <u>ProjectNamingStrategy</u> Default

pattern CONFIGURATOR

Implementation of <u>ProjectNamingStrategy</u>

Pattern

– description	- String
	Provide a human-readable description to explain naming constraints. This will be used as the error message when the job name does not match the pattern.
forceExistingJobs	boolean
– namePattern	- String

standard CONFIGURATOR

Implementation of **CrumbIssuer**

Default Crumb Issuer

excludeClientIPFromCrumb

Some HTTP proxies filter out information that the default crumb issuer uses to calculate the nonce value. If an HTTP proxy sits between your browser client and your Jenkins server and you receive a 403 response when submitting a form to Jenkins, checking this option may help. Using this option makes the nonce value easier to forge.

12/27/2021

plainText CONFIGURATOR

Implementation of <u>MarkupFormatter</u> Plain text

 $rawHtml \ ^{\text{CONFIGURATOR}}$

Implementation of MarkupFormatter

Safe HTML

disableSyntaxHighlighting ----- boolean

 $label Atom \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

name ----- String

properties -----list of LabelAtomProperty

 $active Directory \ ^{\text{CONFIGURATOR}}$

Implementation of SecurityRealm

Active Directory

bindName String

bindPassword ----- String

- cache ----- CacheConfiguration

captchaSupport ----- CaptchaSupport

customDomain ----- Boolean

domain ----- String

domains -----list of ActiveDirectoryDomain

Active Directory plugin allows you to use a multiple domains configuration. Normally, Jenkins will discover the Active Directory domain controllers via DNS, and connect them in the order of priority specified in DNS. You can override this behavior by explicitly specifying the host names of the domain controllers (and optionally the LDAP port, by appending ":NNN".)

Multiple domain controllers can be specified by using ',' as a separator.

This is useful when the advertised list of domain controller priority is messed up.

environmentProperties ------ list of EnvironmentProperty

groupLookupStrategy ----- GroupLookupStrategy

Determines how group membership lookup is performed.

This option is here because we the Jenkins developers are between a rock and a hard place when it comes to determining groups that an user belongs to. We have several approaches implemented over the time, but none of them seem to satisfy everyone. So we basically threw hands in the air and asking you to find one that works for you.

The only people who should try to tweak this setting is (1) those who find login very slow or (2) those who find that Jenkins isn't picking up all the groups that you belong to. The general rule of thumb is that you try both options while paying attention to the time it takes to login and the groups reported in the "/whoAmI" page, and pick one that works for you.

Automatic

Do the best to pick the "best" algorithm. Specifically, Jenkins tries LDAP_MATCHING_RULE_IN_CHAIN first, and if it's going too slow, then stick to the "recursive queries".

Recursive queries

For each group X that Jenkins discovers, recursively list up all the other groups that has X as a member, by issuing a separate LDAP query. This increases the number of queries, but each query remains simple. Note that several people reported that this approach fails to find some groups that the user is actually a member of, but the developers of this plugin do not have access to such AD deployment to investigate the claim.

LDAP MATCHING RULE IN CHAIN

Use <u>Microsoft extension to LDAP</u> that was specifically added to Windows 2003 to perform recursive group membership lookup. The general consensus in the developer community appears to be that <u>this is the way to go</u>, but several Jenkins users have reported that <u>this approach makes lookup intolerably slow</u>.

Token-Groups user attribute

Use only the <u>Token-Groups</u> attribute. This attribute will contain all the <u>security groups</u> that a user is a member of, including all security groups via recursion of other security groups. If the user is a member of any <u>distribution groups</u>, or security groups via membership of a distribution group then these will not be included in the returned groups. However if users are members of large number of distribution groups and your security setup does not require the use of distribution groups then this will provide the ultimate performance.

It is less than ideal to ask users to make a choice like this. If you have insights to share to improve this situation, or even just data point, <u>please add your thoughts to JENKINS-22830</u>.

Possible values:

- AUTO
- RECURSIVE
- CHAIN
- TOKENGROUPS

internalUsersDatabase

ActiveDirectoryInternalUsersDatabase

This option allows to fallback into the Jenkins Internal User Database for the specific user selected (ONLY ONE USER).

Useful when the AD server is not reachable for any reason and you cannot log-in. This option allows a specific user not to disable security on disk and being able to log-in even under networking issues. You can still perform administration tasks under networking issues.

With this option is enabled the plugin will fallback into the Jenkins Internal User Database for the specific user selected when there is a NamingException. The password of this user will be in sync with the Active Directory as it gets updated each time this user logins into the instance.

remove Irrelevant Groups

boolean

Makes Jenkins ignore every user's Active Directory groups that are not being used by the active Authorization Strategy. This can significantly improve performance in environments with a large number of groups but a small number of corresponding rules defined by the Authorization Strategy. The user is required to re-login in order for the changes to take effect.

Note: this setting may be incompatible with some Authorization Strategies.

site String

startTls ----- Boolean

This property allows you to enable/disable StartTLS. In case the Active Directory plugin is set-up to use TLS, then StartTLS will not try to start. StartTLS will only tries to start in case the communication is started on plain.

12/27/2021	https://ci.my4030.ir/configuration-as-code/reference
cacheConfiguration CONFIGURATOR	
size	
- ttl	int
CONFIGURATOR	
Active Directory name	String
value	String
activeDirectoryInternalUsersI	Database CONFIGURATOR
jenkinsInternalUser	String
CONFIGURATOR	
bindName	String
- bindPassword	String
_ name	String
ervers	String
_ site	String
– tlsConfiguration	Possible values:
	 TRUST ALL CERTIFICATES JDK TRUSTSTORE
legacy CONFIGURATOR Implementation of SecurityRealm Delegate to servlet container	
captchaSupport	CaptchaSupport

local CONFIGURATOR

Implementation of <a>SecurityRealm Jenkins' own user database

enableCaptcha

users

captchaSupport

boolean allowsSignup This option allows users to create accounts by themselves, via the "sign up" link on the top right shoulder of the page. Make sure to not grant significant permissions to authenticated users, as anyone on the network will be able to get them. When this checkbox is unchecked, someone with the administrator role would have to create accounts. By default, Jenkins does not use captcha verification if the user creates an account by themself. If you'd like to enable captcha verification, install a captcha support plugin, e.g. the Jenkins <u>JCaptcha</u> Plugin. captchaSupport CaptchaSupport

boolean

list of UserWithPassword

$userWith Password \ ^{\tt CONFIGURATOR}$

Possible values:

- apiToken
- <u>userCredentialsProperty</u>
- mailer

•

- lastGrantedAuthorities
- <u>myView</u>
- preferredProvider
- paneStatus
- password
- sshPublicKey
- userSeed
- search
- timezone

apiToken CONFIGURATOR
Implementation of UserProperty

$userCredentials Property \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>UserProperty</u>

Credentials

API Token

- domainCredentials ----- list of DomainCredentials

mailer CONFIGURATOR

Implementation of <u>UserProperty</u>

E-mail

emailAddress ----- String

$myView \ ^{\text{CONFIGURATOR}}$

Implementation of <u>UserProperty</u>

My Views

primaryViewName ----- String

----- String

preferredProvider CONFIGURATOR

Implementation of <u>UserProperty</u>

Notification URL

providerId

Allows the user to select their preferred user interface when clicking links to Jenkins from

notifications (e.g. Email, Slack, or GitHub).

sshPublicKey CONFIGURATOR Implementation of UserProperty

https://ci.my4030.ir/configuration-as-code/reference

SSH Public Keys

authorizedKeys String

> List SSH public keys that are associated with the user account. These keys can be used for example by Jenkins CLI.

timezone CONFIGURATOR

Implementation of <u>UserProperty</u>

User Defined Time Zone

Strina timeZoneName

Specify user defined time zone for displaying time rather than system default.

ldap CONFIGURATOR

Implementation of SecurityRealm

LDAP

CacheConfiguration cache -----

> Some LDAP servers may be slow, or rate limit client requests. In such cases enabling caching may improve performance of Jenkins with the risk of delayed propagation of user changes from LDAP and increased memory usage on the Jenkins controller.

Note: The default configuration is to leave the cache turned off.

CaptchaSupport captchaSupport

configurations list of LDAPConfiguration

disableMailAddressResolver ----boolean

disableRolePrefixing hoolean

> For backwards compatibility reasons, the LDAP plugin will create a role ROLE F00 for every role/group Foo of a user by default. If you're not using these ROLE_* roles in your security configuration and don't want this duplication of roles, you can check this option so they'll no longer be created.

IdStrategy groupIdStrategy

> Almost all LDAP schemas use an attribute (cn) for the group name that is specified to be case insensitive. Unless you are connecting to a LDAP server that uses a different search attribute than the defaults you should leave this as Case insensitive.

Recommendations:

- If your group search filter is blank or contains cn={0} then select Case insensitive.
- Otherwise consult the LDAP schema of your LDAP server to determine the correct selection, with Case insensitive being the best choice for the lazy.

Possible values:

- caseInsensitive
- caseSensitive
- caseSensitiveEmailAddress

userIdStrategy IdStrategy

> Almost all LDAP schemas use an attribute (uid or mail) for the user name that is specified to be case insensitive. Unless you are connecting to a LDAP server that uses a different search attribute than the

defaults you should leave this as Case insensitive.

Recommendations:

- If your user search filter is uid={0} then select Case insensitive.
- If your user search filter is mail={0} then select Case insensitive

 Otherwise consult the LDAP schema of your LDAP server to determine the correct selection, with Case insensitive being the best choice for the lazy.

Note:

If you are using an email address attribute for the user search you will want to ask the mail server administrator whether they have case sensitive mailboxes. <u>RFC 2821 Section 2.4</u> specifies:

The local-part of a mailbox MUST BE treated as case sensitive. Therefore, SMTP implementations MUST take care to preserve the case of mailbox local-parts. Mailbox domains are not case sensitive.

This is in conflict with the case sensitivity rules for the standard LDAP attribute used to hold email addresses (mail) which specifies caseIgnoreIA5Match, e.g. as noted in <u>RFC</u> 2798

Note: RFC 1274 uses the longer name 'rfc822Mailbox' and syntax OID of 0.9.2342.19200300.100.3.5. All recent LDAP documents and most deployed LDAP implementations refer to this attribute as 'mail' and define the IA5 String syntax using using the OID 1.3.6.1.4.1.1466.115.121.1.26, as is done here.

So when the user search filter is mail={0} it is pointless to select anything other than Case insensitive as the LDAP server will be ignoring case when searching anyway. If you do need to select a case sensitive strategy because you are searching on a custom attribute you should note that the Case sensitive (email address) strategy applies RFC 2821's rules on case sensitivity (i.e. the local-part is case sensitive and the mailbox domain is case insensitive), thus with that strategy JOE@ACME.COM and JOE@acme.com will be considered the same Jenkins user while joe@acme.com will be considered as a different Jenkins user.

Note: UNIX login names are case sensitive

Possible values:

- caseInsensitive
- caseSensitive
- caseSensitiveEmailAddress

CONFIGURATOR



caseInsensitive CONFIGURATOR

Implementation of **IdStrategy**

Case insensitive

$case Sensitive \ ^{\texttt{CONFIGURATOR}}$

 $Implementation \ of \ \underline{IdStrategy}$

Case sensitive

$case Sensitive Email Address \ ^{\texttt{CONFIGURATOR}}$

Implementation of <u>IdStrategy</u> Case sensitive (email address)

CONFIGURATOR

ldap

displayNameAttributeName ----- String

When a user's details are resolved from LDAP, the specified attribute in those user details will be consulted to retrieve the display name for the user.

environmentProperties ----- list of EnvironmentProperty

groupMembershipStrategy ----- LDAPGroupMembershipStrategy

Possible values:

- fromUserRecord
- fromGroupSearch

groupSearchBase ----- String

One of the searches Jenkins does on LDAP is to locate the list of groups for a user.

This field determines the query to be run to identify the organizational unit that contains groups. The query is almost always "ou=groups" so try that first, though this field may be left blank to search from the root DN.

If login attempts result in "Administrative Limit Exceeded" or similar error, try to make this setting as specific as possible for your LDAP structure, to reduce the scope of the query. If the error persists, you may need to edit the **Group search filter. Change:**

groupSearchFilter = "(| (member={0}) (uniqueMember={0}) (memberUid={1}))"; to query only of the field used in your LDAP for group membership, such as:

groupSearchFilter = "(member={0})";

Then restart Jenkins and retry the login.

groupSearchFilter String

When Jenkins is asked to determine if a named group exists, it uses a default filter of:

(& (cn={0}) (| (objectclass=groupOfNames) (objectclass=groupOfUniqueNames) (objectclass=posixGroup)))

relative to the Group search base to determine if there is a group with the specified name ($\{0\}$ is substituted by the name being searched for)

If you know your LDAP server only stores group information in one specific object class, then you can improve group search performance by restricting the filter to just the required objectclass.

Note: if you are using the LDAP security realm to connect to Active Directory (as opposed to using the <u>Active Directory plugin</u>'s security realm) then you will need to change this filter to:

(& (cn={0}) (objectclass=group))

Note: if you leave this empty, the default search filter will be used, unless the hudson.security.LDAPSecurityRealm.groupSearch has been set to modify the default.

Tf also also di also anatono traditante a the a anathematica di anti-

ignoreIfUnavailable ----- boolean

II cnecked, this option indicates that authentication and user/group membership lookups which fail due to communication errors (e.g., LDAP server is unreachable, manager password is incorrect) will be reattempted using the next configuration. Other types of failures, such as a user attempting to authenticate with an incorrect password, will not be reattempted.

If unchecked (the default), authentication and user/group membership lookups which fail due to communication errors will not be reattempted using the next configuration.

The default is intended to prevent users or groups with the same name in distinct LDAP domains from being inadvertently mapped to the same user or group in Jenkins if one of the domains is unavailable. Consequently, this option should only be enabled if all configured domains have distinct user and group names.

inhibitInferRootDN

boolean

mailAddressAttributeName

String

When a user's details are resolved from LDAP, the specified attribute in those user details will be consulted to retrieve the email address for the user unless the email address resolver is disabled.

managerDN

String

If your LDAP server doesn't support <u>anonymous binding</u> (IOW, if your LDAP server doesn't even allow a query without authentication), then Jenkins would have to first authenticate itself against the LDAP server, and Jenkins does that by sending "manager" DN and password.

A DN typically looks like CN=MyUser, CN=Users, DC=mydomain, DC=com although the exact sequence of tokens depends on the LDAP server configuration. It can be any valid DN as long as LDAP allows this user to query data.

This configuration is also useful when you are connecting to Active Directory from a Unix machine, as AD doesn't allow anonymous bind by default. But if you can't figure this out, you can also change AD setting to allow anonymous bind. See this document for how to.

managerPasswordSecret

Secret

Password for the manager DN specified above.

rootDN

String

For authenticating user and determing the roles given to this user, Jenkins performs multiple LDAP queries. Since an LDAP database is conceptually a big tree and the search is performed recursively, in theory if we can start a search starting at a sub-node (as opposed to root), you get a better performance because it narrows down the scope of a search. This field specifies the DN of such a subtree.

But in practice, LDAP servers maintain an extensive index over the data, so specifying this field is rarely necessary — you should just let Jenkins figure this out by talking to LDAP.

If you do specify this value, the field normally looks something like "dc=sun,dc=com"

server

String

Specify the name of the LDAP server host name(s) (like ldap.sun.com). If your LDAP server uses a port other than 389 (which is the standard for LDAP), you can also append a port number here, like ldap.acme.org:1389. To connect to LDAP over SSL (AKA LDAPS), specify it with the ldaps://protocol, like ldaps://ldap.acme.org or ldaps://ldap.acme.org:1636 (if the port is other than the default 636).

Where there are multiple redundant servers separate them by spaces, if the first named server is not available then requests will be attempted by each of the named servers in order. NOTE: This is to support LDAP server fail-over, once an any answer has been received from a server, that answer is taken as definitive

userSearch String One of the searches Jenkins does on LDAP is to locate the user record given the user name. This field determines the query to be run to identify the user record. The query is almost always "uid={0}" as per defined in RFC 2798, so in most cases you should leave this field empty and let this default kick in. If your LDAP server doesn't have uid or doesn't use a meaningful uid value, try "mail={0}", which lets people login by their e-mail address. If you do specify a different query, specify an LDAP query string with marker token "{0}", which is to be replaced by the user name string entered by the user. String userSearchBase One of the searches Jenkins does on LDAP is to locate the user record given the user name. If you specify a relative DN (from the root DN) here, Jenkins will further narrow down searches to the sub-tree. But in practice, LDAP servers maintain an extensive index over the data, so specifying this field is rarely necessary. If you do specify this value, the field normally looks something like "ou=people" CONFIGURATOR String name String value $from User Record \ ^{\text{CONFIGURATOR}}$ Implementation of LDAPGroupMembershipStrategy Parse user attribute for list of LDAP groups attributeName $from Group Search \ {}^{\text{CONFIGURATOR}}$ Implementation of LDAPGroupMembershipStrategy Search for LDAP groups containing user filter String pam CONFIGURATOR Implementation of **SecurityRealm** Unix user/group database CaptchaSupport captchaSupport serviceName String standard CONFIGURATOR Implementation of MyViewsTabBar **Default My Views TabBar** $update Center \ {}^{\hbox{\scriptsize CONFIGURATOR}}$ list of UpdateSite updateSite CONFIGURATOR

String id String

proxy CONFIGURATOR **Proxy Configuration**

name String

If your Jenkins server sits behind a firewall and does not have the direct access to the internet, and if your server JVM is not configured appropriately (See JDK networking properties for more details) to enable internet connection, you can specify the HTTP proxy server name in this field to allow Jenkins to install plugins on behalf of you. Note that Jenkins uses HTTPS to communicate with the update center to download plugins.

Leaving this field empty means Jenkins will try to connect to the internet directly.

If you are unsure about the value, check the browser proxy configuration.

String noProxyHost

> Specify host name patterns that shouldn't go through the proxy, one host per line. "*" is the wild card host name (such as "*.jenkins.io" or "www*.jenkins-ci.org")

int port

This field works in conjunction with the proxy server field to specify the HTTP proxy port.

Secret secretPassword

String testUrl

String userName

> This field works in conjunction with the proxy server field to specify the username used to authenticate with the proxy.

If this proxy requires Microsofts NTLM authentication scheme then the domain name can be encoded within the username by prefixing the domain name followed by a back-slash '\' before the username, e.g "ACME\John Doo".

standard CONFIGURATOR

Implementation of ViewsTabBar

Default Views TabBar

permanent CONFIGURATOR

Implementation of Node

Permanent Agent

String labelString ComputerLauncher launcher Possible values: • jnlp command • managedWindowsServiceLauncher

Mode mode

Possible values:

- NORMAL
- EXCLUSIVE

name ·····	- String
nodeDescription	- String
- nodeName	String
nodeProperties	list of NodeProperty Possible values:
	 disableDeferredWipeout authorizationMatrix envVars envInject toolLocation
- numExecutors	- int
- remoteFS	String
- retentionStrategy	RetentionStrategy Possible values:
	 <u>always</u> <u>schedule</u> <u>demand</u>

always CONFIGURATOR

Implementation of **RetentionStrategy** Keep this agent online as much as possible

schedule CONFIGURATOR

Implementation of <u>RetentionStrategy</u>
Bring this agent online according to a schedule

keepUpWhenActive	boolean
- startTimeSpec	String
upTimeMins	int

demand CONFIGURATOR

Implementation of **RetentionStrategy**

Bring this agent online when in demand, and take offline when idle

– idleDelay	 long
– inDemandDelay	long

jnlp CONFIGURATOR

Implementation of <u>ComputerLauncher</u>
Launch agent by connecting it to the controller

— tunnel	String
– vmargs	String If the agent JVM should be launched with additional VM arguments, such as "-Xmx256m", specify those here. List of all the options are available here .
webSocket	boolean
	Use WebSocket to connect to the Jenkins master rather than the TCP port. See <u>JEP-222</u> for
	background.

RemotingWorkDirSettings workDirSettings

CONFIGURATOR

otingWorkDirSettings	
disabled	- boolean
	Allows disabling Remoting Work Directory for the agent. In such case the agent will be running
	in the legacy mode without logging enabled by default.
failIfWorkDirIsMissing	- boolean
	If defined, Remoting will fail at startup if the target work directory is missing. The option may
	be used to detect infrastructure issues like failed mount.
internalDir	String
	Defines a storage directory for the internal data. This directory will be created within the
	Remoting working directory.
workDirPath	String
	If defined, a custom Remoting work directory will be used instead of the Agent Root Directory.
	This option has no environment variable resolution so far, it is recommended to use only
	absolute paths.
	failIfWorkDirIsMissinging

command CONFIGURATOR

Implementation of **ComputerLauncher**

Launch agent via execution of command on the controller

String

Single command to launch an agent program, which controls the agent computer and communicates with the controller. Jenkins assumes that the executed program launches the agent.jar program on the correct machine.

A copy of agent. jar can be downloaded from here.

In a simple case, this could be something like ssh hostname java -jar ~/bin/agent.jar.

Note: the command can't rely on a shell to parse things, e.g. echo foo > bar; baz. If you need to do that, either use sh -c or write the expression into a script and point to the script.

It is often a good idea to write a small shell script, like the following, on an agent so that you can control the location of Java and/or agent.jar, as well as set up any environment variables specific to this node, such as PATH.

#!/bin/sh exec java -jar ~/bin/agent.jar

You can use any command to run a process on the agent machine, such as RSH, as long as stdin/stdout of the process on the controller will be connected to those of java -jar ~/bin/agent.jar on the agent machine eventually.

In a larger deployment, it is also worth considering to load agent. jar from a NFS-mounted common location, so that you don't have to update this file on every agent machines every time you update Jenkins.

Setting this to ssh -v hostname may be useful for debugging connectivity issue.

ssh CONFIGURATOR

Implementation of ComputerLauncher Launch agents via SSH

https://ci.my4030.ir/configuration-as-code/reference

credentialsId	String
host	String
_ javaPath	String
- jvmOptions	String
- launchTimeoutSeconds	Integer
- maxNumRetries	Integer
port	int
prefixStartSlaveCmd	String
retryWaitTime	Integer
sshHostKeyVerificationStrategy	SshHostKeyVerificationStrategy
	Possible values:
	 knownHostsFileKeyVerificationStrategy manuallyProvidedKeyVerificationStrategy manuallyTrustedKeyVerificationStrategy nonVerifyingKeyVerificationStrategy
- suffixStartSlaveCmd	String
tcpNoDelay	boolean
workDir	String

$known Hosts File Key Verification Strategy \ ^{\text{CONFIGURATOR}}$

Implementation of SshHostKeyVerificationStrategy

Known hosts file Verification Strategy

$manually Provided Key Verification Strategy \ ^{\text{CONFIGURATOR}}$

Implementation of SshHostKeyVerificationStrategy

Manually provided key Verification Strategy

String

The SSH key expected for this connection. This key should be in the form 'algorithm value' where algorithm is one of ssh-rsa or ssh-dss, and value is the Base 64 encoded content of the key.

$manually Trusted Key Verification Strategy \ ^{\text{CONFIGURATOR}}$

Implementation of SshHostKeyVerificationStrategy

Manually trusted key Verification Strategy

boolean requireInitialManualTrust

> Require a user with Computer.CONFIGURE permission to authorise the key presented during the first connection to this host before the connection will be allowed to be established.

If this option is not enabled then the key presented on first connection for this host will be automatically trusted and allowed for all subsequent connections without any manual intervention.

$nonVerifying Key Verification Strategy \ ^{\text{CONFIGURATOR}}$ Implementation of SshHostKeyVerificationStrategy

Non verifying Verification Strategy

$managed Windows Service Launcher \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

Implementation of **ComputerLauncher**

Let Jenkins control this Windows agent as a Windows service

_ account	ManagedWindowsServiceAccount
uccount	Sometimes the administrator account that can install a service remotely might not be the user
	account you want to run your Jenkins agent (one reason you might want to do this is to run
	your builds/tests in more restricted account because you don't trust them. Another reason you
	might want to do this is to run agents in some domain user account.) This option lets you do
	this. If left unchecked, this Jenkins agent will run as the 'SYSTEM' user, who has full access to
	the local system.
	Possible values:
	• <u>localSystem</u>
	• anotherUser
	• administrator
– host	String
1333	Provide the host name of the Windows host if different to the name of the Agent.
_ javaPath	String
javaratii	Path to the Java executable to be used on this node. Defaults to "java", assuming JRE is
	installed and available on system PATH (e.g. C:\Program Files\Java\jre7\bin\java.exe)
password	String
userName ·····	String
	Provide the name of the Windows user who has the administrative access on this computer,
	such as 'Administrator'. This information is necessary to start a process remotely.
	To specify a domain user, enter values like 'DOMAIN\Administrator'.
- vmargs	String
localSystem CONFIGURATOR	
Implementation of ManagedWindo	wsServiceAccount
Use Local System User	
.1 TT CONEICLEDATOR	
anotherUser CONFIGURATOR	Control on the Control of the Contro
Implementation of ManagedWindor Log on using a different account	ws5erviceAccount

ar

Log on using a different account

Secret password ----- String userName

administrator CONFIGURATOR

 $Implementation \ of \ \underline{ManagedWindowsServiceAccount}$ Use Administrator account given above

$remoting Security \ ^{\text{CONFIGURATOR}}$ enabled

CO	nfiguration	© ROOT ELEMENT
	providerFilter	CredentialsProviderFilter

Possible values:

- none
- excludes
- includes

restrictions list of CredentialsProviderTypeRestriction

Possible values:

- excludes
- includes

typeFilter CredentialsTypeFilter

Possible values:

- none
- excludes
- includes

none CONFIGURATOR

Implementation of **CredentialsTypeFilter** All available

excludes CONFIGURATOR

Implementation of **CredentialsTypeFilter**

Exclude selected

----- list of String classNames

includes CONFIGURATOR

Implementation of <u>CredentialsTypeFilter</u> Only selected

classNames list of String

excludes CONFIGURATOR

Implementation of <u>CredentialsProviderTypeRestriction</u> **Excludes**

provider ---- String

includes CONFIGURATOR

Implementation of <u>CredentialsProviderTypeRestriction</u> Includes

provider String type

none CONFIGURATOR

Implementation of CredentialsProviderFilter All available

excludes CONFIGURATOR

Implementation of <u>CredentialsProviderFilter</u>

Exclude selected

classNames list of String includes CONFIGURATOR Implementation of **CredentialsProviderFilter** Only selected classNames list of String crumb ROOT ELEMENT apiToken ROOT ELEMENT boolean creationOfLegacyTokenEnabled This option allows users to generate a legacy API token if they do not already have one. Because legacy tokens are deprecated, we recommend disabling it and having users instead generate new API tokens from the user configuration page. tokenGenerationOnCreationEnabled This option causes a legacy API token to be generated automatically for newly created users. Because legacy tokens are deprecated, we recommend disabling it and having users instead generate new API tokens from the user configuration page as needed. usageStatisticsEnabled If this option is enabled, then the date of the most recent use of each API token and the total number of times it has been used are stored in Jenkins. This allows users to see if they have unused or outdated API tokens which should be revoked. This data is stored in your Jenkins instance and will not be used for any other purpose. queueItemAuthenticator ® ROOT ELEMENT authenticators list of QueueItemAuthenticator updateSiteWarningsConfiguration ® ROOT ELEMENT masterKillSwitchConfiguration PROOT ELEMENT scriptApproval ® ROOT ELEMENT approvedSignatures ----- list of String envInjectPluginConfiguration ® ROOT ELEMENT sSHD ROOT ELEMENT port Jenkins can act as an SSH server to run a subset of Jenkins CLI commands. Plugins may also use this service to expose additional commands. Specify this TCP/IP port number. **copyartifact** ® ROOT ELEMENT mode ${\tt CopyArtifactCompatibilityMode}$ How Copy Artifact checks permissions to access artifacts to copy: **Production** RECOMMENDED.

Performs permission checks while running builds. Builds copying artifacts from another builds (source builds) require appropriate permission configurations unless source builds are allowed to read from any users.

You can configure "Permission to Copy Artifact" in jobs to copy artifacts from. It configures jobs allowed to copy artifacts from that job. You can configure that with copyArtifactPermission in pipelines.

You can configure "Access Control for Builds" in Manage Jenkins > Configure Global Security so that builds run with authentication allowed to access source builds. That configuration is shown only when plugins providing "Access Control for Builds" are installed. See those plugin documentations for details.

Migration

Performs permission checks when configuring jobs or when running builds if the name of the source job is configured with variables. This is the behavior compatible to Copy Artifact 1.43.1 or earlier.

This may cause security vulnerabilities allowing malicious users to bypass permission checks. This mode is provided only to allow users to upgrade job configurations and migrate to Production mode easily. You should migrate to Production mode as soon as you can.

Build executions that would fail in Production mode are recorded and displayed as warnings to administrators. Administrators should check those warnings and update the job configurations to successfully use Production mode. You can see more details about how to update the job configurations in the "Production" section and the page linked from the warning messages.

Possible values:

- PRODUCTION
- MIGRATION

artifactoryBuilder ® ROOT ELEMENT

artifactoryServers	list of ArtifactoryServer
jfrogPipelinesServer	JFrogPipelinesServer
useCredentialsPlugin	boolean

jFrogPipelinesServer CONFIGURATOR

- bypassProxy	boolean
- connectionRetries	· int
- credentialsConfig	CredentialsConfig
- integrationUrl	String
- timeout	int

credentialsConfig CONFIGURATOR

credentialsId	 String
 ignoreCredentialPluginDisabled 	 boolean
	Dooloon

overriding Credentials	оотеап
– password	String
username	String

artifactoryServer CONFIGURATOR

_	artifactoryUrl	String
_	bypassProxy	boolean
_	connectionRetry	int
_	deployerCredentialsConfig	CredentialsConfig
_	deploymentThreads	int
_	resolverCredentialsConfig	CredentialsConfig
_	serverId	String
	timeout	int

singleConditionalBuilder ® ROOT ELEMENT

resourceRoot ® ROOT ELEMENT

url ----- String

Jenkins serves many files that are potentially created by untrusted users, such as files in project workspaces or archived artifacts. When no resource root URL is defined, Jenkins will serve these files with the HTTP header Content-Security-Policy ("CSP"). By default it is set to a value that disables many modern web features to prevent cross-site scripting (XSS) and other attacks on Jenkins users accessing these files. While the specific value for the CSP header is user configurable (and can even be disabled), doing so is a trade-off between security and functionality.

If the resource root URL is defined, Jenkins will instead redirect requests for user-created resource files to URLs starting with the URL configured here. These URLs will not set the CSP header, allowing JavaScript and similar features to work. For this option to work as expected, the following constraints and considerations apply:

- The resource root URL must be a valid alternative choice for the Jenkins URL for requests to be processed correctly.
- The Jenkins URL must be set and it must be different from this resource root URL (in fact, a different host name is required).
- Once set, Jenkins will only serve resource URL requests via the resource root URL. All
 other requests will get HTTP 404 Not Found responses.

Once this URL has been set up correctly, Jenkins will redirect requests to workspaces, archived artifacts, and similar collections of usually user-generated content to URLs starting with the resource root URL. Instead of a path like job/name_here/ws, resource URLs will contain a token encoding that path, the user for which the URL was created, and when it was created. These resource URLs access static files as if the user for which they were created would access them: If the user's permission to access these files is removed, the corresponding resource URLs will not work anymore either. These URLs are accessible to anyone without authentication until they expire, so sharing these URLs is akin to sharing the files directly.

Security considerations

Authentication

Resource URLs do not require authentication (users will not have a valid session for the resource root URL). Sharing a resource URL with another user, even one lacking Overall/Read permission for Jenkins, will grant that user access to these files until the URLs expire.

Expiration

Resource URLs expire after 30 minutes by default. Expired resource URLs will redirect users to their equivalent Jenkins URLs, so that the user can reauthenticate, if necessary, and then be redirected back to a new resource URL that will be valid for another 30 minutes. This will generally be transparent to the user if they have a valid Jenkins session. Otherwise, they will need to authenticate with Jenkins again. However, when browsing pages with HTML frames, like Javadoc sites, the login form cannot appear in a frame. In these cases, users will need to reload the top-level frame to make the login form appear.

To change how quickly resource URLs expire, set the system property jenkins.security.ResourceDomainRootAction.validForMinutes to the desired value in minutes. Earlier expiration might make it harder to use these URLs, while later expiration increases the likelihood of unauthorized users gaining access through URLs shared with them by authorized users.

Authenticity

Resource URLs encode the URL, the user for which they were created, and their creation timestamp. Additionally, this string contains an <u>HMAC</u> to ensure the authenticity of the URL. This prevents attackers from forging URLs that would grant them access to resource files as if they were another user.

If checked, any changes in the library will be included in the changesets of a build, and changing the library would cause new builds to run for Pipelines that include this library.

sSHBuildWrapper ROOT ELEMENT

globalLibraries ® ROOT ELEMENT

libraries list of LibraryConfiguration

CONFIGURATOR

LibraryConfiguration

Tury Cominguitation	
allowVersionOverride	boolean
	If checked, scripts may select a custom version of the library by appending @someversion in the
	@Library annotation. Otherwise they are restricted to using the version you select here.
cachingConfiguration	LibraryCachingConfiguration
	If checked, versions fetched using this library will be cached on the controller.
- defaultVersion	String
	A default version of the library to load if a script does not select another. Might be a branch
	name, tag, commit hash, etc., according to the SCM.
- implicit	boolean
	If checked, scripts will automatically have access to this library without needing to request it
	via @Library.
- includeInChangesets	boolean

https://ci.my4030.ir/configuration-as-code/reference

This can be overridden in the jenkinsfile: @Library(value="name@version", changelog=true|false)

name ----- String

An identifier you pick for this library, to be used in the @Library annotation. An environment variable Library.THIS_NAME.version will also be set to the version loaded for a build (whether that comes from the Default version here, or from an annotation after the @ separator).

retriever LibraryRetriever

Possible values:

- modernSCM
- legacySCM

$modernSCM \ ^{\text{CONFIGURATOR}}$

Implementation of <u>LibraryRetriever</u> Modern SCM

libraryPath String

A relative path from the root of the SCM to the root of the library. Leave this field blank if the root of the library is the root of the SCM. Note that ".." is not permitted as a path component to avoid security issues.

scm ---- SCMSource

Possible values:

- <u>git</u>
- github
- mercurial
- fromScm
- <u>subversion</u>

git CONFIGURATOR Implementation of SCMSource Git

credentialsId String
Credentials used to scan branches and check out sources.

id String
remote String
Specify the URL of this remote repository. This uses the s

Specify the URL of this remote repository. This uses the same syntax as your git clone command.

traits ----- list of SCMSourceTrait

Possible values:

- checkoutOptionTrait
- cloneOptionTrait
- submoduleOptionTrait
- localBranchTrait
- gitHubSshCheckout
- cleanMercurial
- cleanAfterCheckoutTrait
- cleanBeforeCheckoutTrait
- gitBrowser
- mercurialBrowser

- remoteName
- <u>userIdentityTrait</u>
- gitBranchDiscovery
- gitHubBranchDiscovery
- discoverOtherRefsTrait
- gitHubForkDiscovery
- gitHubPullRequestDiscovery
- gitTagDiscovery
- gitHubTagDiscovery
- headRegexFilter
- headWildcardFilter
- gitLFSPullTrait
- ignoreOnPushNotificationTrait
- gitHubIgnoreDraftPullRequestFilter
- pruneStaleBranchTrait
- pruneStaleTagTrait
- gitTool
- mercurialInstallation
- sparseCheckoutPathsTrait
- <u>refSpecs</u>
- <u>authorInChangelogTrait</u>
- wipeWorkspaceTrait

checkoutOption CONFIGURATOR

Implementation of **GitSCMExtension** Advanced checkout behaviours

> Integer timeout

> > Specify a timeout (in minutes) for checkout.

This option overrides the default timeout of 10 minutes.

You can change the global git timeout via the property

org.jenkinsci.plugins.gitclient.Git.timeOut (see JENKINS-11286). Note that property should be set on both controller and agent to have effect (see JENKINS-22547).

$\boldsymbol{cloneOption} \ ^{\text{CONFIGURATOR}}$

shallow

Implementation of **GitSCMExtension**

Advanced clone behaviours

depth	Integer Set shallow clone depth, so that git will only download recent history of the project, saving time and disk space when you just want to access the latest commits of a repository.
– honorRefspec	boolean Perform initial clone using the refspec defined for the repository. This can save time, data
	transfer and disk space when you only need to access the references specified by the refspec.
noTags	- boolean
	Deselect this to perform a clone without tags, saving time and disk space when you just want to
	access what is specified by the refspec.
reference	String
	Specify a folder containing a repository that will be used by Git as a reference during clone operations.
	This option will be ignored if the folder is not available on the controller or agent where the

clone is being executed.

boolean

Perform shallow clone, so that git will not download the history of the project, saving time and

disk space when you just want to access the latest version of a repository.

timeout

Specify a timeout (in minutes) for clone and fetch operations.

This option overrides the default timeout of 10 minutes.

You can change the global git timeout via the property

org.jenkinsci.plugins.gitclient.Git.timeOut (see JENKINS-11286). Note that property should be set on both controller and agent to have effect (see **JENKINS-22547**).

$submodule Option \ ^{\text{CONFIGURATOR}}$

Implementation of GitSCMExtension Advanced sub-modules behaviours

dvanced sub-modules behaviours	
— depth	Integer
	Set shallow clone depth, so that git will only download recent history of the project, saving time
	and disk space when you just want to access the latest commits of a repository.
- disableSubmodules	boolean
	By disabling support for submodules you can still keep using basic git plugin functionality and
	just have Jenkins to ignore submodules completely as if they didn't exist.
parentCredentials	boolean
	Use credentials from the default remote of the parent project.
- recursiveSubmodules	boolean
	Retrieve all submodules recursively (uses 'recursive' option which requires git>=1.6.5)
reference	String
	Specify a folder containing a repository that will be used by Git as a reference during clone operations.
	This option will be ignored if the folder is not available on the controller or agent where the
	clone is being executed.
	To prepare a reference folder with multiple subprojects, create a bare git repository and add all the remote urls then perform a fetch:
	git initbare
	git remote add SubProject1 https://gitrepo.com/subproject1
	git remote add SubProject2 https://gitrepo.com/subproject2
	git fetchall
- shallow	boolean
Silais II	Perform shallow clone, so that git will not download the history of the project, saving time and
	disk space when you just want to access the latest version of a repository.
— threads	Integer
an cads	Specify the number of threads that will be used to update submodules.
	If unspecified, the command line git default thread count is used.
timeout	Integer
	Specify a timeout (in minutes) for submodules operations.
	This option overrides the default timeout of 10 minutes.
	You can change the global git timeout via the property
	org.jenkinsci.plugins.gitclient.Git.timeOut (see <u>JENKINS-11286</u>). Note that property should be
	set on both controller and agent to have effect (see <u>JENKINS-22547</u>).
- trackingSubmodules	boolean
3	Retrieve the tip of the configured branch in .gitmodules (Uses 'remote' option which requires

σit>=1 8 7)

3

$build Single Revision Only \ ^{\text{CONFIGURATOR}}$

Implementation of <u>GitSCMExtension</u> Build single revision only

changelogToBranch CONFIGURATOR

Implementation of **GitSCMExtension**

Calculate changelog against a specific branch

options ---- ChangelogToBranchOptions

CONFIGURATOR

compareRemote ----- String

Name of the repository, such as origin, that contains the branch you specify below.

compareTarget ----- String

The name of the branch within the named repository to compare against.

relativeTargetDirectory CONFIGURATOR

Implementation of **GitSCMExtension**

Check out to a sub-directory

relativeTargetDir ----- String

Specify a local directory (relative to the workspace root) where the Git repository will be

checked out. If left empty, the workspace root itself will be used.

This extension should not be used in Jenkins Pipeline (either declarative or scripted). Jenkins Pipeline already provides standard techniques for checkout to a subdirectory. Use <u>ws</u> and <u>dir</u> in

Jenkins Pipeline rather than this extension.

localBranch CONFIGURATOR

Implementation of <u>GitSCMExtension</u> Check out to specific local branch

- localBranch ----- String

cleanCheckout CONFIGURATOR

Implementation of **GitSCMExtension**

Clean after checkout

deleteUntrackedNestedRepositories - boolear

Deletes untracked submodules and any other subdirectories which contain .git directories.

cleanBeforeCheckout CONFIGURATOR

Implementation of **GitSCMExtension**

Clean before checkout

deleteUntrackedNestedRepositories — boolean

Deletes untracked submodules and any other subdirectories which contain .git directories.

perBuildTag CONFIGURATOR

Implementation of **GitSCMExtension**

Create a tag for every build

scmName CONFIGURATOR

Implementation of **GitSCMExtension**

Custom SCM name

String name $userIdentity \ {}^{\hbox{\scriptsize CONFIGURATOR}}$ Implementation of **GitSCMExtension** Custom user name/e-mail address String email If given, "GIT_COMMITTER_EMAIL=[this]" and "GIT_AUTHOR_EMAIL=[this]" are set for builds. This overrides whatever is in the global settings.

String

If given, "GIT_COMMITTER_NAME=[this]" and "GIT_AUTHOR_NAME=[this]" are set for builds. This overrides whatever is in the global settings.

ignoreNotifyCommit CONFIGURATOR

Implementation of **GitSCMExtension** Don't trigger a build on commit notifications

disableRemotePoll CONFIGURATOR

Implementation of **GitSCMExtension** Force polling using workspace

gitLFSPull CONFIGURATOR

Implementation of GitSCMExtension Git LFS pull after checkout

$preBuildMerge \ ^{\hbox{\scriptsize CONFIGURATOR}}$

Implementation of **GitSCMExtension** Merge before build

> options UserMergeOptions

CONFIGURATOR

fastForwardMode **GitPluginFastForwardMode**

Merge fast-forward mode selection.

The default, --ff, gracefully falls back to a merge commit when required.

For more information, see the **Git Merge Documentation**

Possible values:

- <u>FF</u>
- FF ONLY
- NO FF

mergeRemote String

> Name of the repository, such as origin, that contains the branch you specify below. If left blank, it'll default to the name of the first repository configured above.

mergeStrategy

Merge strategy selection. This feature is not fully implemented in JGIT.

Possible values:

- **DEFAULT**
- RESOLVE
- RECURSIVE

- OCTOPUS
- **OURS**
- **SUBTREE**
- RECURSIVE_THEIRS

mergeTarget

String

The name of the branch within the named repository to merge to, such as master.

userExclusion CONFIGURATOR

Implementation of **GitSCMExtension** Polling ignores commits from certain users

excludedUsers

String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any revisions committed by users in this list when determining if a build needs to be triggered. This can be used to exclude commits done by the build itself from triggering another build, assuming the build server commits the change with a distinct SCM user.

Using this behaviour will preclude the faster git ls-remote polling mechanism, forcing polling to require a workspace thus sometimes triggering unwanted builds, as if you had selected the Force polling using workspace extension as well.

Each exclusion uses exact string comparison and must be separated by a new line. User names are only excluded if they exactly match one of the names in this list.

auto_build_user

The example above illustrates that if only revisions by "auto_build_user" have been committed to the SCM a build will not occur.

pathRestriction CONFIGURATOR

Implementation of **GitSCMExtension** Polling ignores commits in certain paths

excludedRegions

String

Each exclusion uses java regular expression pattern matching, and must be separated by a new line.

myapp/src/main/web/.*\.html myapp/src/main/web/.*\.jpeg myapp/src/main/web/.*\.gif

The example above illustrates that if only html/jpeg/gif files have been committed to the SCM a build will not occur.

includedRegions

String

Each inclusion uses java regular expression pattern matching, and must be separated by a new line. An empty list implies that everything is included.

myapp/src/main/web/.*\.html myapp/src/main/web/.*\.jpeg myapp/src/main/web/.*\.gif

The example above illustrates that a build will only occur, if html/jpeg/gif files have been committed to the SCM. Exclusions take precedence over inclusions, if there is an overlap between included and excluded regions.

messageExclusion CONFIGURATOR	
Implementation of <u>GitSCMExtension</u> Polling ignores commits with certain message	c c
excludedMessage	
excludediviessage	If set, and Jenkins is set to poll for changes, Jenkins will ignore any revisions committed with
	message matched to <u>Pattern</u> when determining if a build needs to be triggered. This can be used
	to exclude commits done by the build itself from triggering another build, assuming the build
	server commits the change with a distinct message.
	Exclusion uses Pattern matching
	.*\[maven-release-plugin\].*
	The example above illustrates that if only revisions with "[maven-release-plugin]" message in first comment line have been committed to the SCM a build will not occur.
	You can create more complex patterns using embedded flag expressions.
	(?s).*F00.*
	This example will search FOO message in all comment lines.
pruneStaleBranch CONFIGURATOR Implementation of GitSCMExtension Prune stale remote-tracking branches	
pruneTags CONFIGURATOR	
Implementation of GitSCMExtension	
Prune stale tags	
pruneTags	boolean
sparseCheckoutPaths CONFIGURATO Implementation of GitSCMExtension	IR
Sparse Checkout paths	
- sparseCheckoutPaths	list of SparseCheckoutPath
CONFIGURATOR	
Path	
– path	String
buildChooserSetting CONFIGURATOR Implementation of GitSCMExtension Strategy for choosing what to build	
- buildChooser	BuildChooser
	Possible values:
	• <u>ancestry</u>
	• <u>default</u>
	• inverse
CONFIGURATOR	
ancestry CONFIGURATOR	
Implementation of BuildChooser Ancestry	
- ancestorCommitSha1	String

maximumAgeInDays

Integer

default CONFIGURATOR

Implementation of **BuildChooser** Default

inverse CONFIGURATOR

Implementation of **BuildChooser**

Inverse

 $author In Changelog \ {\tt CONFIGURATOR}$

Implementation of **GitSCMExtension** Use commit author in changelog

wipeWorkspace CONFIGURATOR

Implementation of **GitSCMExtension** Wipe out repository & force clone

CONFIGURATOR

Advanced checkout behaviours

extension -----CheckoutOption

CONFIGURATOR

Advanced clone behaviours

.....CloneOption extension

CONFIGURATOR

Advanced sub-modules behaviours

SubmoduleOption extension

CONFIGURATOR

Check out to matching local branch

$git Hub Ssh Check out \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

Checkout over SSH

String credentialsId Credentials used to check out sources. Must be a SSH key based credential.

CONFIGURATOR

Clean Build

CONFIGURATOR

Clean after checkout

CleanCheckout extension

CONFIGURATOR

Clean before checkout

CleanBeforeCheckout

CONFIGURATOR

Configure Repository Browser

-----GitRepositoryBrowser browser

1 CONFIGURATOR	
assembla CONFIGURATOR	
Implementation of RepositoryBrowser Assembla	
- spaceName	String
assemblaWeb CONFIGURATOR	
Implementation of RepositoryBrowser AssemblaWeb	
– repoUrl	String
	Specify the root URL serving this repository (such as https://www.assembla.com/code/PROJECT/git/).
collabNetSVN CONFIGURATOR Implementation of RepositoryBrowser CollabNet	
– url	URL
	The repository browser URL for the root of the project. For example, a Java.net project called myproject would use https://myproject.dev.java.net/source/browse/myproject.
fishEyeSVN CONFIGURATOR Implementation of RepositoryBrowser FishEye	
rootModule	String
	Specify the root Subversion module that this FishEye monitors. For example, for
	http://fisheye6.cenqua.com/browse/ant/, this field would be ant because it displays the
	directory "/ant" of the ASF repo. If FishEye is configured to display the whole SVN repository,
	leave this field empty.
_ url	URL
	Specify the URL of this module in FishEye. (such as
	http://fisheye6.cenqua.com/browse/ant/)
fisheyeGit CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> FishEye	
repoUrl	String
repoerr	Specify the URL of this repository in FishEye (such as http://fisheye6.cenqua.com/browse/ant/).
fishEyeCVS CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> FishEye	
- url	URL
	Specify the root URL of FishEye for this repository (such as this.)
kilnGit CONFIGURATOR	
Implementation of RepositoryBrowser Kiln	
repoUrl	String
_	Specify the root URL serving this repository (such as
	https://khanacademy.kilnhg.com/Code/Website/Group/webapp).

 $tFS2013Git\ {}^{\texttt{CONFIGURATOR}}$

 $Implementation \ of \ \underline{RepositoryBrowser}$

Microsoft Team Foundation Server/Visual	Studio Team Services
- repoUrl	String
	Either the name of the remote whose URL should be used, or the URL of this module in TFS
	(such as http://fisheye6.cenqua.com/tfs/PROJECT/_git/REPO/). If empty (default), the URL of
	the "origin" repository is used.
	If TFS is also used as the repository server, this can usually be left blank.
openGrok CONFIGURATOR	
Implementation of RepositoryBrowser	
OpenGrok	
_ url	URL
	Specify the root URL of OpenGrok for this repository.
phabricator CONFIGURATOR	
Implementation of RepositoryBrowser Phabricator	
- repo	String
_ url	URL
scmManager CONFIGURATOR Implementation of RepositoryBrowser SCM-Manager	
_ url	String
	Specify the root URL serving this repository (such as
	http://YOURSCMMANAGER/scm/repo/NAMESPACE/NAME/).
sVNWeb CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> SVN::Web	
_ url	URL
sventon CONFIGURATOR	
Implementation of RepositoryBrowser	
repositoryInstance	String
url	UKL
sventon2 CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> Sventon 2.x	
- repositoryInstance	String
- url	URL
viewCVS CONFIGURATOR	
Implementation of RepositoryBrowser ViewCVS	
url	URL
	Specify the root URL of ViewCVS for this repository (such as this).

viewSVN CONFIGURATOR	
Implementation of RepositoryBrowser	
ViewSVN	
url	
	Specify the root URL of ViewSVN for this repository (such as this).
visualSVN CONFIGURATOR	
Implementation of RepositoryBrowser VisualSVN	
url	String
	Specify the root URL of VisualSVN for this repository (such as https://demo-
	server.visualsvn.com/!/#tortoisesvn).
webSVN CONFIGURATOR	
Implementation of RepositoryBrowser WebSVN	
- url	URL
bitbucketServer CONFIGURATOR	
Implementation of RepositoryBrowser bitbucketserver	
repoUrl	String
repoor	Specify the Bitbucket Server root URL for this repository (such as
	https://bitbucket:7990/OWNER/REPO/).
bitbucketWeb CONFIGURATOR	
Implementation of RepositoryBrowser	
bitbucketweb	
repoUrl	String
	Specify the root URL serving this repository (such as https://bitbucket.org/OWNER/REPO/).
cGit CONFIGURATOR	
Implementation of RepositoryBrowser	
cgit	
- repoUrl	
	Specify the root URL serving this repository (such as
	http://cgit.example.com:port/group/REPO/).
fishEye CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> fisheye	
– url	
	Specify the root URL serving this repository, such as: http://www.example.org/browse/hg/
gitBlit CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> gitblit	
projectName	String
	Specify the name of the project in GitBlit.
repoUrl ·····	String
	Specify the root URL serving this repository.

githubWeb CONFIGURATOR Implementation of RepositoryBrowser githubweb	
- repoUrl	String Specify the HTTP URL for this repository's GitHub page (such as https://github.com/jquery/jquery).
gitiles CONFIGURATOR Implementation of RepositoryBrowser	
repoUrl ······	String
Topoon	Specify the root URL serving this repository (such as https://gwt.googlesource.com/gwt/).
gitLab CONFIGURATOR	
version	String
	Specify the major and minor version of GitLab you use (such as 9.1). If you don't specify a version, a modern version of GitLab (>= 8.0) is assumed.
gitList CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> gitlist	
repoUrl ·····	String
	Specify the root URL serving this repository (such as http://gitlistserver:port/REPO/).
gitorious Web CONFIGURATOR Implementation of Repository Browser gitorious web	
- repoUrl	Specify the root URL serving this repository (such as https://gitorious.org/gitorious/mainline).
gitWeb CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> gitweb	
- repoUrl	Specify the root URL serving this repository (such as https://github.com/jenkinsci/jenkins.git).
gogsGit CONFIGURATOR Implementation of RepositoryBrowser gogs	
repoUrl	String
	Specify the root URL serving this repository (such as http://gogs.example.com:port/username/some-repo-url.git).
googleCode CONFIGURATOR Implementation of RepositoryBrowser googlecode	
url	String
	Specify the root URL serving this repository (such as this).
hgWeb CONFIGURATOR	
Implementation of <u>RepositoryBrowser</u> hgweb	

String url Specify the root URL serving this repository (such as this). kallithea CONFIGURATOR Implementation of RepositoryBrowser String kilnHG CONFIGURATOR Implementation of RepositoryBrowser kilnhg String url Specify the root URL serving this repository (such as this). phabricator CONFIGURATOR Implementation of RepositoryBrowser String repo Specify the repository name in phabricator (such as the foo part of phabricator.example.com/diffusion/foo/browse). String repoUrl Specify the phabricator instance root URL (such as http://phabricator.example.com). redmineWeb CONFIGURATOR Implementation of RepositoryBrowser redmineweb String repoUrl Specify the root URL serving this repository (such as http://SERVER/PATH/projects/PROJECT/repository). $rhodeCode \ {}^{\tt CONFIGURATOR}$ Implementation of RepositoryBrowser rhodecode String repoUrl Specify the HTTP URL for this repository's RhodeCode page (such as http://rhodecode.mydomain.com:5000/projects/PROJECT/repos/REPO/). $rhodeCode \ ^{\texttt{CONFIGURATOR}}$ Implementation of RepositoryBrowser rhodecode url String Specify the root URL serving this repository (such as this). $rhode Code Legacy \ {}^{\hbox{\scriptsize CONFIGURATOR}}$ Implementation of RepositoryBrowser rhodecode (pre-1.2.0) url String stash CONFIGURATOR Implementation of RepositoryBrowser String repoUrl Specify the HTTP URL for this repository's Stash page (such as

http://stash.mydomain.com:7990/projects/PROJECT/repos/REPO/).

 $view Git Web \ ^{\texttt{CONFIGURATOR}}$ Implementation of RepositoryBrowser viewgit String projectName Specify the name of the project in ViewGit (e.g. scripts, scuttle etc. from http://code.fealdia.org/viewgit/). String repoUrl Specify the root URL serving this repository (such as http://code.fealdia.org/viewgit/). CONFIGURATOR **Configure Repository Browser** browser HgBrowser CONFIGURATOR Configure remote name String remoteName CONFIGURATOR Custom user name/e-mail address UserIdentity extension $gitBranchDiscovery \ ^{\text{CONFIGURATOR}}$ Discover branches

$git Hub Branch Discovery \ ^{\text{CONFIGURATOR}}$

Discover branches

strategyId -----int

Determines which branches are discovered.

Exclude branches that are also filed as PRs

If you are discovering origin pull requests, you may not want to also build the source branches for those pull requests.

Only branches that are also filed as PRs

Similar to discovering origin pull requests, but discovers the branch rather than the pull request. This means env. GIT_BRANCH will be set to the branch name rather than PR-#.

Also, status notifications for these builds will only be applied to the commit and not to the pull request.

All branches

Ignores whether the branch is also filed as a pull request and instead discovers all branches on the origin repository.

CONFIGURATOR

Discover other refs
nameMapping String

Mapping for how the ref can be named in for example the @Library.

Example: test-@{1}

Where @{1} replaces the first wildcard in the ref when discovered.

By default it will be "namespace_before_wildcard-@{1}". E.g. if ref is "test/*/merged" the default mapping would be "test-@{1}".

String ref

> The pattern under /refs on the remote repository to discover, can contain a wildcard. Example: test/*/merged

gitHubForkDiscovery CONFIGURATOR

Discover pull requests from forks

strategyId

Determines how pull requests are discovered:

Merging the pull request with the current target branch revision

Discover each pull request once with the discovered revision corresponding to the result of merging with the current revision of the target branch.

The current pull request revision

Discover each pull request once with the discovered revision corresponding to the pull request head revision without merging.

Both the current pull request revision and the pull request merged with the current target branch revision

Discover each pull request twice. The first discovered revision corresponds to the result of merging with the current revision of the target branch in each scan. The second parallel discovered revision corresponds to the pull request head revision without merging.

SCMHeadAuthority trust

> One of the great powers of pull requests is that anyone with read access to a repository can fork it, commit some changes to their fork and then create a pull request against the original repository with their changes. There are some files stored in source control that are important. For example, a Jenkinsfile may contain configuration details to sandbox pull requests in order to mitigate against malicious pull requests. In order to protect against a malicious pull request itself modifying the Jenkinsfile to remove the protections, you can define the trust policy for pull requests from forks.

Other plugins can extend the available trust policies. The default policies are:

Nobody

Pull requests from forks will all be treated as untrusted. This means that where Jenkins requires a trusted file (e.g. Jenkinsfile) the contents of that file will be retrieved from the target branch on the origin repository and not from the pull request branch on the fork repository.

Collaborators

Pull requests from collaborators to the origin repository will be treated as trusted, all other pull requests from fork repositories will be treated as untrusted. Note that if credentials used by Jenkins for scanning the repository does not have permission to query the list of collaborators to the origin repository then only the origin account will be treated as trusted - i.e. this will fall back to Nobody. NOTE: all collaborators are trusted, even if they are only members of a team with read permission.

Everyone

All pull requests from forks will be treated as trusted. NOTE: this option can be dangerous if used on a public repository hosted on GitHub.

From users with Admin or Write permission

Pull requests forks will be treated as trusted if and only if the fork owner has either Admin or Write permissions on the origin repository. This is the recommended policy. Note that this strategy requires the Review a user's permission level API, as a result on GitHub Enterprise Server versions before 2.12 this is the same as trusting Nobody.

Possible values:

- gitHubTrustContributors
- gitHubTrustEveryone
- gitHubTrustPermissions
- gitHubTrustNobody
- branch
- gitHubBranchHeadAuthority
- originChangeRequest
- tag
- <u>tag</u>

gitHubTrustContributors CONFIGURA

Implementation of <u>SCMHeadAuthority</u> Collaborators

gitHubTrustEveryone CONFIGURATOR

Implementation of <u>SCMHeadAuthority</u> Everyone

gitHubTrustPermissions CONFIGURATOR

Implementation of <u>SCMHeadAuthority</u>
From users with Admin or Write permission

$gitHubTrustNobody \ ^{\text{CONFIGURATOR}}$

Implementation of <u>SCMHeadAuthority</u> Nobody

gitHubPullRequestDiscovery

CONFIGURATO

Discover pull requests from origin

strategyId -----int

Determines how pull requests are discovered:

Merging the pull request with the current target branch revision

Discover each pull request once with the discovered revision corresponding to the result of merging with the current revision of the target branch.

The current pull request revision

Discover each pull request once with the discovered revision corresponding to the pull request head revision without merging.

Both the current pull request revision and the pull request merged with the current target branch revision

Discover each pull request twice. The first discovered revision corresponds to the result of merging with the current revision of the target branch in each scan. The second parallel discovered revision corresponds to the pull request head revision without merging.

gitTagDiscovery CONFIGURATOR Discover tags

gitHubTagDiscovery CONFIGURATOR Discover tags

headRegexFilter CONFIGURATOR

Filter by name (with regular expression)

regex

A Java regular expression to restrict the names. Names that do not match the supplied regular expression will be ignored.

NOTE: this filter will be applied to all branch like things, including change requests

headWildcardFilter CONFIGURATOR

Filter by name (with wildcards)

excludes

Space-separated list of name patterns to ignore even if matched by the includes list. For example: release alpha-* beta-*

NOTE: this filter will be applied to all branch like things, including change requests

includes

String

Space-separated list of name patterns to consider. You may use * as a wildcard; for example: master release*

NOTE: this filter will be applied to all branch like things, including change requests

CONFIGURATOR

Git LFS pull after checkout

CONFIGURATOR

Ignore on push notifications

$git Hub Ignore Draft Pull Request Filter \ {}^{\tt CONFIGURATOR}$

Ignore pull requests marked as drafts

CONFIGURATOR

Prune stale remote-tracking branches

CONFIGURATOR

Prune stale tags

CONFIGURATOR

Select Git executable

gitTool ----- String

CONFIGURATOR

Select Mercurial Installation

installation ----- String

CONFIGURATOR

Sparse Checkout paths

extension ----- SparseCheckoutPaths

CONFIGURATOR

Specify ref specs

templates ---- list of RefSpecTemplate

CONFIGURATOR

Ref Spec

value ---- Strin

A ref spec to fetch. Any occurrences of <code>@{remote}</code> will be replaced by the remote name (which

defaults to origin) before use.

CONFIGURATOR

Use commit author in changelog

CONFIGURATOR

Wipe out repository & force clone

github CONFIGURATOR

Implementation of SCMSource GitHub

apiUri	String
	The server to connect to. The list of servers is configured in the Manage Jenkins » Configure
	Jenkins > GitHub Enterprise Servers screen.
configuredByUrl	boolean
credentialsId	String
	Credentials used to scan branches and pull requests, check out sources and mark commit statuses.
	Note that only "username with password" credentials are supported. Existing credentials of other kinds will be filtered out. This is because Jenkins uses the GitHub API, which does not support other ways of authentication.
	If none is given, only the public repositories will be scanned, and commit status will not be set on GitHub.
	If your organization contains private repositories, then you need to specify a credential from a user who has access to those repositories. This is done by creating a "username with password" credential where the password is <u>GitHub personal access tokens</u> . The necessary scope is "repo".
_ id	String
repoOwner	String
	Specify the name of the GitHub Organization or GitHub User Account.
repository	String
	The repository to scan.
repositoryUrl	String
	Specify the HTTPS URL of the GitHub Organization / User Account and repository.
	GitHub examples:
	 https://github.com/jenkinsci/github-branch-source-plugin https://github.com/jenkinsci/github-branch-source-plugin.git
	GitHub Enterprise examples:
	 https://myccompany.github.com/jenkinsci/github-branch-source-plugin
	 https://myccompany.github.com/jenkinsci/github-branch-source-plugin.git
- traits	list of SCMSourceTrait
	The behaviours control what is discovered from the GitHub repository. The behaviours are grouped into a number of categories:

Within repository

These behaviours determine what gets discovered. If you do not configure at least one discovery behaviour then nothing will be found!

General

These behaviours affect the configuration of each discovered branch / pull request.

Possible values:

- checkoutOptionTrait
- cloneOptionTrait
- submoduleOptionTrait
- localBranchTrait
- gitHubSshCheckout
- cleanMercurial
- cleanAfterCheckoutTrait
- cleanBeforeCheckoutTrait
- gitBrowser
- mercurialBrowser
- remoteName
- userIdentityTrait
- gitBranchDiscovery
- gitHubBranchDiscovery
- discoverOtherRefsTrait
- gitHubForkDiscovery
- gitHubPullRequestDiscovery
- gitTagDiscovery
- gitHubTagDiscovery
- headRegexFilter
- headWildcardFilter
- gitLFSPullTrait
- ignoreOnPushNotificationTrait
- $\bullet \ \ \underline{gitHubIgnoreDraftPullRequestFilter}$
- pruneStaleBranchTrait
- pruneStaleTagTrait
- gitTool
- mercurialInstallation
- sparseCheckoutPathsTrait
- refSpecs
- authorInChangelogTrait
- wipeWorkspaceTrait

mercurial CONFIGURATOR

Implementation of <u>SCMSource</u> Mercurial

ŀ	credentialsId	String
		Optional credentials to use when cloning or pulling from the remote repository. Supports username/password with HTTP(S) URLs, and SSH private key with SSH URLs.
-	- id	String
	source	String Specify the repository to track. This can be URL or a local file path. If you are specifying HTTE credentials, do <i>not</i> include a username in the URL.
	traits ·····	list of SCMSourceTrait Possible values:

- checkoutOptionTrait
- cloneOptionTrait
- <u>submoduleOptionTrait</u>
- localBranchTrait
- gitHubSshCheckout
- cleanMercurial
- cleanAfterCheckoutTrait
- cleanBeforeCheckoutTrait
- gitBrowser
- mercurialBrowser
- remoteName
- userIdentityTrait
- gitBranchDiscovery
- gitHubBranchDiscovery
- discoverOtherRefsTrait
- gitHubForkDiscovery
- gitHubPullRequestDiscovery
- gitTagDiscovery
- gitHubTagDiscovery
- headRegexFilter
- headWildcardFilter
- gitLFSPullTrait
- ignoreOnPushNotificationTrait
- gitHubIgnoreDraftPullRequestFilter
- pruneStaleBranchTrait
- pruneStaleTagTrait
- gitTool
- mercurialInstallation
- sparseCheckoutPathsTrait
- refSpecs
- authorInChangelogTrait
- wipeWorkspaceTrait

fromScm CONFIGURATOR Implementation of SCMSource

Single repository & branch

gie repository & branch	
- id	String
_ name	String
	The name of the SCM head/trunk/branch/tag that this source provides.
- scm	SCM
	Possible values:
	• <u>none</u>
	• <u>cvs</u>
	• <u>cvsProjectset</u>
	• <u>git</u>
	• mercurial
	• <u>subversion</u>
- traits	list of SCMSourceTrait
	Possible values:

• checkoutOptionTrait

- cloneOptionTrait
- <u>submoduleOptionTrait</u>
- localBranchTrait
- gitHubSshCheckout
- cleanMercurial
- cleanAfterCheckoutTrait
- cleanBeforeCheckoutTrait
- gitBrowser
- mercurialBrowser
- remoteName
- <u>userIdentityTrait</u>
- gitBranchDiscovery
- gitHubBranchDiscovery
- discoverOtherRefsTrait
- gitHubForkDiscovery
- gitHubPullRequestDiscovery
- gitTagDiscovery
- gitHubTagDiscovery
- headRegexFilter
- headWildcardFilter
- gitLFSPullTrait
- ignoreOnPushNotificationTrait
- gitHubIgnoreDraftPullRequestFilter
- pruneStaleBranchTrait
- pruneStaleTagTrait
- gitTool
- mercurialInstallation
- sparseCheckoutPathsTrait
- refSpecs
- authorInChangelogTrait
- wipeWorkspaceTrait

none CONFIGURATOR

Implementation of <u>SCM</u> None

cvs Configurator

 $\begin{array}{c} \textbf{Implementation of } \underline{\textbf{SCM}} \\ \textbf{CVS} \end{array}$

canUseUpdate boolean

If checked, Jenkins will use 'cvs update' whenever possible for builds. This makes a build faster. But this also causes the artifacts from the previous build to remain in the file system when a new build starts, making it not a true clean build.

checkoutCurrentTimestamp ----- boolean

Advanced option. Should probably be left unchecked.

The build quiet period is designed to assist with CVS checkouts by waiting for a specific period of time without commits. Normally you want the checkout to reflect the time when the quiet period was exited successfully. Select this option if you need to re-enable the legacy behaviour of Jenkins, i.e. using the time that the build started checking out as the timestamp for the checkout operation. Note: enabling this option can result in the quiet period being defeated

especially in those cases where the build is not able to start immediately after exiting the quiet

period.

cleanOnFailedUpdate ----- boolean

If the job is configured to use CVS update and the update step fails for any reason then the

workspace will be wiped-out and a clean checkout done instead.

disableCvsQuiet -----boolean

Instructs CVS to show all logging output. CVS normally runs in quiet mode but this option

disables that.

forceCleanCopy ----- boolean

If checked, Jenkins will add the 'C' option to the CVS update command to force it to over-write any files with local modifications, rather than attempt a merge or leave them as they are.

legacy boolean

Hudson 1.20 and earlier used to create redundant directories inside the workspace. For example, if the CVS module name is "foo/bar", it first created "foo/bar" and then put everything below. With this option checked off, there will be no more such unnecessary intermediate directories.

If you have multiple modules to check out, this option is forced (otherwise they'll overlap.)

This affects other path specifiers, such as artifact archivers --- you now specify "build/foo.jar" instead of "foo/build/foo.jar".

pruneEmptyDirectories ----- boolean

Remove empty directories after checkout using the CVS '-P' option.

repositories ----- list of CvsRepository

skipChangeLog -----boolean

Prevent the changelog being generated after checkout has completed. This will stop any changes being shown on the changes screen but reduces load on your CVS server.

CONFIGURATOR

CVS Repository

compressionLevel -----int

cvsRoot ---- String

The CVS connection string Jenkins uses to connect to the server. The format is the same as

\$CVSROOT environment variable (:protocol:user@host:path)

excludedRegions list of ExcludedRegion

If set, and Jenkins is set to poll for changes, Jenkins will ignore any files and/or folders in this list when determining if a build needs to be triggered.

Each exclusion uses regular expression pattern matching, and must be separated by a new line.

src/main/web/.*\.html
src/main/web/.*\.jpeg

src/main/web/.*\.gif

The example above illustrates that if only html/jpeg/gif files have been committed to the SCM a build will not occur.

More information on regular expressions can be found here.

password ----- String

nasswordRequired ----- boolean

https://ci.my4030.ir/configuration-as-code/reference

12/27/2021 ----- CVSRepositoryBrowser repositoryBrowser ----- list of CvsRepositoryItem repositoryItems CONFIGURATOR **Excluded regions** pattern ----- String CONFIGURATOR **CVS Repository Item** location CvsRepositoryLocation Possible values: • branchRepositoryLocation • headRepositoryLocation • tagRepositoryLocation list of CvsModule modules $branch Repository Location \ ^{\text{CONFIGURATOR}}$ Implementation of **CvsRepositoryLocation** Branch branchName String useHeadIfNotFound -----boolean

$head Repository Location \ ^{\text{CONFIGURATOR}}$

Implementation of CvsRepositoryLocation Head

$tag Repository Location \ ^{\text{CONFIGURATOR}}$

Implementation of CvsRepositoryLocation

Tag tagName useHeadIfNotFound boolean

CONFIGURATOR

CVS Module

localName

The name to be applied to this module in the local workspace. If this is left blank then the remote module name will be used. This is similar to the 'checkout-as' function available on many CVS clients. projectsetFileName String The name of the file in this module to parse for projectset entries. remoteName String

The name of the module in the repository at CVSROOT

String

cvsProjectset CONFIGURATOR

Implementation of SCM

CVS Projectset

----- CVSRepositoryBrowser browser

canUseUpdate	boolean If checked, Jenkins will use 'cvs update' whenever possible for builds. This makes a build faster. But this also causes the artifacts from the previous build to remain in the file system when a new build starts, making it not a true clean build.
- cleanOnFailedUpdate	If the job is configured to use CVS update and the update step fails for any reason then the workspace will be wiped-out and a clean checkout done instead.
- disableCvsQuiet	Instructs CVS to show all logging output. CVS normally runs in quiet mode but this option disables that.
- forceCleanCopy	boolean
password	This password will be used for the checkout of any modules parsed from the projectset file if no match was found against the parsed CVSROOT using the globally configured authentication.
pruneEmptyDirectories	Remove empty directories after checkout using the CVS '-P' option.
- repositories	list of CvsRepository
– skipChangeLog	Prevent the changelog being generated after checkout has completed. This will stop any changes being shown on the changes screen but reduces load on your CVS server.
username	This username will be used for the checkout of any modules parsed from the projectset file if no match was found against the parsed CVSROOT using the globally configured authentication.

git CONFIGURATOR Implementation of **SCM**

Git

branches	list of BranchSpec
	List of branches to build. Jenkins jobs are most effective when each job builds only a single
	branch. When a single job builds multiple branches, the changelog comparisons between
	branches often show no changes or incorrect changes.
– browser	GitRepositoryBrowser
	Defines the repository browser that displays changes detected by the git plugin.
- buildChooser	BuildChooser
	Possible values:
	• ancestry
	• <u>default</u>
	• inverse

 $do Generate Submodule Configurations \quad {\tt boolean}$

Removed facility that was intended to test combinations of git submodule versions. Removed in git plugin 4.6.0. Ignores the user provided value and always uses false as its value.

extensions list of GitSCMExtension

> Extensions add new behavior or modify existing plugin behavior for different uses. Extensions help users more precisely tune plugin behavior to meet their needs.

Extensions include:

- Clone extensions modify the git operations that retrieve remote changes into the agent workspace. The extensions can adjust the amount of history retrieved, how long the retrieval is allowed to run, and other retrieval details.
- Checkout extensions modify the git operations that place files in the workspace from the
 git repository on the agent. The extensions can adjust the maximum duration of the
 checkout operation, the use and behavior of git submodules, the location of the workspace
 on the disc, and more.
- Changelog extensions adapt the source code difference calculations for different cases.
- Tagging extensions allow the plugin to apply tags in the current workspace.
- Build initiation extensions control the conditions that start a build. They can ignore
 notifications of a change or force a deeper evaluation of the commits when polling.
- Merge extensions can optionally merge changes from other branches into the current branch of the agent workspace. They control the source branch for the merge and the options applied to the merge.

Possible values:

- checkoutOption
- cloneOption
- submoduleOption
- buildSingleRevisionOnly
- changelogToBranch
- relativeTargetDirectory
- localBranch
- cleanCheckout
- cleanBeforeCheckout
- perBuildTag
- scmName
- userIdentity
- ignoreNotifyCommit
- disableRemotePoll
- gitLFSPull
- preBuildMerge
- userExclusion
- pathRestriction
- messageExclusion
- pruneStaleBranch
- pruneTags
- sparseCheckoutPaths
- buildChooserSetting
- authorInChangelog
- wipeWorkspace

extensions

list of GitSCMExtension

Extensions add new behavior or modify existing plugin behavior for different uses. Extensions help users more precisely tune plugin behavior to meet their needs.

Extensions include:

- Clone extensions modify the git operations that retrieve remote changes into the agent workspace. The extensions can adjust the amount of history retrieved, how long the retrieval is allowed to run, and other retrieval details.
- Checkout extensions modify the git operations that place files in the workspace from the git repository on the agent. The extensions can adjust the maximum duration of the

checkout operation, the use and behavior of git submodules, the location of the workspace on the disc, and more.

- Changelog extensions adapt the source code difference calculations for different cases.
- Tagging extensions allow the plugin to apply tags in the current workspace.
- Build initiation extensions control the conditions that start a build. They can ignore
 notifications of a change or force a deeper evaluation of the commits when polling.
- Merge extensions can optionally merge changes from other branches into the current branch of the agent workspace. They control the source branch for the merge and the options applied to the merge.

Possible values:

- checkoutOption
- cloneOption
- submoduleOption
- buildSingleRevisionOnly
- changelogToBranch
- relativeTargetDirectory
- localBranch
- cleanCheckout
- cleanBeforeCheckout
- perBuildTag
- scmName
- userIdentity
- ignoreNotifyCommit
- disableRemotePoll
- gitLFSPull
- preBuildMerge
- userExclusion
- pathRestriction
- messageExclusion
- pruneStaleBranch
- pruneTags
- sparseCheckoutPaths
- buildChooserSetting
- authorInChangelog
- wipeWorkspace

gitTool ----- String

Absolute path to the git executable.

This is different from other Jenkins tool definitions. Rather than providing the directory that contains the executable, you must provide the complete path to the executable. Setting '/usr/bin/git' would be correct, while setting '/usr/bin/' is not correct.

submoduleCfg ---- list of SubmoduleConfig

Removed facility that was intended to test combinations of git submodule versions. Removed in git plugin 4.6.0. Ignores the user provided value(s) and always uses empty values.

userRemoteConfigs ----- list of UserRemoteConfig

Specify the repository to track. This can be a URL or a local file path. Note that for superprojects (repositories with submodules), only a local file path or a complete URL is valid. The following are examples of valid git URLs.

ياد بادا بالمادار المادار الما

- ssn://git@gitnub.com/gitnub/git.git
- git@github.com:github/git.git (short notation for ssh protocol)
- ssh://user@other.host.com/~/repos/R.git (to access the repos/R.git repository in the user's home directory)
- https://github.com/github/git.git

If the repository is a super-project, the location from which to clone submodules is dependent on whether the repository is bare or non-bare (i.e. has a working directory).

- If the super-project is bare, the location of the submodules will be taken from .qitmodules.
- If the super-project is not bare, it is assumed that the repository has each of its
 submodules cloned and checked out appropriately. Thus, the submodules will be taken
 directly from a path like \${SUPER_PROJECT_URL}/\${SUBMODULE}, rather than relying on
 information from .qitmodules.

For a local URL/path to a super-project, *git rev-parse --is-bare-repository* is used to detect whether the super-project is bare or not.

For a remote URL to a super-project, the ending of the URL determines whether a bare or non-bare repository is assumed:

- If the remote URL ends with /.git, a non-bare repository is assumed.
- If the remote URL does NOT end with /.qit, a bare repository is assumed.

$submodule Config {\ }^{{\scriptsize {\sf CONFIGURATOR}}}$

-	branches		list of String
			Removed in git plugin 4.6.0.
	submodul	eName	String
			Removed in git plugin 4.6.0.

CONFIGURATOR

– credentialsId	Credential used to check out sources.
— name	ID of the repository, such as origin, to uniquely identify this repository among other remote repositories. This is the same "name" that you use in your git remote command. If left empty, Jenkins will generate unique names for you.
	You normally want to specify this when you have multiple remote repositories.
refspecA refspec	c controls the remote refs to be retrieved and how they map to local refs. If left blank, it will

A refspec controls the remote refs to be retrieved and how they map to local refs. If left blank, it will default to the normal behaviour of git fetch, which retrieves all the branch heads as remotes/REPOSITORYNAME/BRANCHNAME. This default behaviour is OK for most cases.

In other words, the default refspec is "+refs/heads/*:refs/remotes/REPOSITORYNAME/*" where REPOSITORYNAME is the value you specify in the above "name of repository" textbox.

When do you want to modify this value? A good example is when you want to just retrieve one branch. For example, +refs/heads/master:refs/remotes/origin/master would only retrieve the master branch

and nothing else.

The plugin uses a default refspec for its initial fetch, unless the "Advanced Clone Option" is set to honor refspec. This keeps compatibility with previous behavior, and allows the job definition to decide if the refspec should be honored on initial clone.

Multiple refspecs can be entered by separating them with a space character.

+refs/heads/master:refs/remotes/origin/master +refs/heads/develop:refs/remotes/origin/develop retrieves the master branch and the develop branch and nothing else.

See the refspec definition in Git user manual for more details.

url

String

Specify the URL or path of the git repository. This uses the same syntax as your git clone command.

CONFIGURATOR

Branch Spec

name

String

Specify the branches if you'd like to track a specific branch in a repository. If left blank, all branches will be examined for changes and built.

The safest way is to use the refs/heads/

syntax. This way the expected branch is unambiguous.

If your branch name has a / in it make sure to use the full reference above. When not presented with a full path the plugin will only use the part of the string right of the last slash. Meaning foo/bar will actually match bar.

If you use a wildcard branch specifier, with a slash (e.g. release/), you'll need to specify the origin repository in the branch names to make sure changes are picked up. So e.g. origin/release/

Possible options:

•

dranchName>

Tracks/checks out the specified branch. If ambiguous the first result is taken, which is not necessarily the expected one. Better use refs/heads/<bra>cbranchName>.

E.g. master, feature1, ...

• refs/heads/<branchName>

Tracks/checks out the specified branch.

E.g. refs/heads/master, refs/heads/feature1/master, ...

<remoteRepoName>/<branchName>

Tracks/checks out the specified branch. If ambiguous the first result is taken, which is not necessarily the expected one.

Better use refs/heads/
branchName>.

E.g. origin/master

remotes/<remoteRepoName>/<branchName>

Tracks/checks out the specified branch.

E.g. remotes/origin/master

• refs/remotes/<remoteRepoName>/<branchName>

Tracks/checks out the specified branch.

E.g. refs/remotes/origin/master

<tagName>

This does not work since the tag will not be recognized as tag.

Use refs/tags/<tagName> instead.

E.g. git-2.3.0

refs/tags/<tagName>

Tracks/checks out the specified tag.

E.g. refs/tags/git-2.3.0

< <commitId>

Checks out the specified commit.

E.g. 5062ac843f2b947733e6a3b105977056821bd352, 5062ac84, ...

\${ENV_VARIABLE}

It is also possible to use environment variables. In this case the variables are evaluated and the result is used as described above.

E.g. \${TREEISH}, refs/tags/\${TAGNAME}, ...

<Wildcards>

The syntax is of the form: REPOSITORYNAME/BRANCH. In addition, BRANCH is recognized as a shorthand of */BRANCH, '*' is recognized as a wildcard, and '**' is recognized as wildcard that includes the separator '/'. Therefore, origin/branches* would match origin/branches-foo but not origin/branches/foo, while origin/branches** would match both origin/branches-foo and origin/branches/foo.

• :<regular expression>

The syntax is of the form: :regexp. Regular expression syntax in branches to build will only build those branches whose names match the regular expression.

Examples:

- o :^(?!(origin/prefix)).*
 - matches: origin or origin/master or origin/feature
 - does not match: origin/prefix or origin/prefix_123 or origin/prefix-abc
- origin/release-\d{8}
 - matches: origin/release-20150101
 - does not match: origin/release-2015010 or origin/release-201501011 or origin/release-20150101-something
- o :^(?!origin/master\$|origin/develop\$).*
 - matches: origin/branch1 or origin/branch-2 or origin/master123 or origin/develop-123
 - does not match: origin/master or origin/develop

mercurial CONFIGURATOR

Implementation of <u>SCM</u> Mercurial

r	curial	
ŀ	browser	HgBrowser
ŀ	- clean	boolean
		When this option is checked, each build will wipe any local modifications or untracked files in
		the repository checkout. This is often a convenient way to ensure that a build is not using any
		artifacts from earlier builds.
ŀ	credentialsId	String
		Optional credentials to use when cloning or pulling from the remote repository. Supports
		username/password with HTTP(S) URLs, and SSH private key with SSH URLs.
ŀ	disableChangeLog	boolean
		When checked, Hudson will not calculate the Mercurial changelog for each build. Disabling the
		changelog can decrease the amount of time needed to update a very large repository.
-	installation	String
ŀ	modules	String

Reduce unnecessary builds by specifying a comma or space delimited list of "modules" within the repository. A module is a directory name within the repository that this project lives in. If this field is set, changes outside the specified modules will not trigger a build (even though the whole repository is checked out anyway due to the Mercurial limitation.)

revision ---- String

Specify the branch or tag name you would like to track. (If you do not type anything, the default value is the default branch.)

revisionType ----- RevisionType

Specify the kind of revision you expect Jenkins to update your working copy to.

Possible values:

- BRANCH
- **TAG**
- CHANGESET
- REVSET

sourceString

Specify the repository to track. This can be URL or a local file path. If you are specifying HTTP credentials, do *not* include a username in the URL.

subdir String

If not empty, check out the Mercurial repository into this subdirectory of the job's workspace. For example: my/sources (use forward slashes). If changing this entry, you probably want to clean the workspace first.

$subversion \ ^{\text{CONFIGURATOR}}$

additionalCredentials

Implementation of <u>SCM</u> Subversion

list of AdditionalCredentials

If there are additional credentials required in order to obtain a complete checkout of the source, they can be provided here.

The realm is how the repository self-identifies to a client. It usually has the following format:

- proto is the protocol, e.g. http or svn.
- host is the host how it's accessed by Jenkins, e.g. as IP address 192.168.1.100, host name synserver, or host name and domain syn.example.org.
- port is the port, even if not explicitly specified. By default, this is 80 for HTTP, 443 for HTTPS, 3690 for the svn protocol.
- Realm Name is how the repository self-identifies. Common options include VisualSVN Server, Subversion Authentication or the UUID of the repository.

To find out the realm, you could do any of the following:

- If you access the repository via HTTP or HTTPS: Open the repo in a web browser
 without saved credentials. It will use the Realm Name (see above) in the authentication
 dialog.
- · Use the command line svn program.
 - If you don't have stored the credentials, run e.g. svn info https://svnserver/repo and it will tell you the realm when asking you to enter a password, e.g.:
 Authentication realm: <svn://svnserver:3690> VisualSVN Server.
 - If you have already stored the credentials to access the repository, look for the realm name in one of the files in ~/.subversion/auth/svn/simple; it will be two

lines below the line svn:realmstring.

When accessing a repository via the svn+ssh protocol, the realm has the format
username@svn+ssh://host:port – note that the username is before the svn+ssh:// (unlike
the URL used for normal SVN operations), and that there are no angle brackets and no
realm name. For this protocol the default port is 22.

Make sure to enter the realm *exactly* as shown, starting with a < (except for repositories accessed via svn+ssh – see above).

browser

SubversionRepositoryBrowser

excluded Commit Messages

String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any revisions with commit messages containing any of the given regular expressions when determining if a build needs to be triggered.

excludedRegions

String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any files and/or folders in this list when determining if a build needs to be triggered.

Each exclusion uses regular expression pattern matching, and must be separated by a new line.

/trunk/myapp/src/main/web/.*\.html
/trunk/myapp/src/main/web/.*\.jpeg
/trunk/myapp/src/main/web/.*\.gif

The example above illustrates that if only html/jpeg/gif files have been committed to the SCM a build will not occur.

More information on regular expressions can be found here.

excludedRevprop

String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any revisions that are marked with the given revision property (revprop) when determining if a build needs to be triggered. This can be used to exclude commits done by the build itself from triggering another build, assuming the build server commits the change with the correct revprop.

This type of exclusion only works with Subversion 1.5 servers and newer.

More information on revision properties can be found here.

excludedUsers

String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any revisions committed by users in this list when determining if a build needs to be triggered. This can be used to exclude

commits done by the build itself from triggering another build, assuming the build server commits the change with a distinct SCM user.

Each exclusion uses literal pattern matching, and must be separated by a new line.

auto_build_user

The example above illustrates that if only revisions by "auto_build_user" have been committed to the SCM a build will not occur.

filterChangelog

boolea

If set Jenkins will apply the same inclusion and exclusion patterns for displaying changelog entries as it does for polling for changes. If this is disabled, changes which are excluded for

polling are still displayed in the changelog.

ignoreDirPropChanges -----boole

If set, Jenkins ignores svn-property only changes of directories. These changes are ignored when determining whether a build should be triggered and are removed from a build's changeset. Main usage of this property is to ignore svn:mergeinfo changes (which would otherwise e.g. lead to a complete rebuild of a maven project, in spite of incremental build option).

includedRegions ---- String

If set, and Jenkins is set to poll for changes, Jenkins will ignore any files and/or folders that are *not* in this list when determining if a build needs to be triggered.

Each inclusion uses regular expression pattern matching, and must be separated by a new line.

This is useful when you need to check out an entire resource for building, but only want to do the build when a subset has changed.

/trunk/myapp/c/library1/.*
/trunk/myapp/c/library2/.*

If /trunk/myapp is checked out, the build will only occur when there are changes to either the c/library1 and c/library2 subtrees.

If there are also excluded regions specified, then a file is not ignored when it is in the included list and *not* in the excluded list.

More information on regular expressions can be found here.

locations ----- list of ModuleLocation

quietOperation ----- boolean

Mimics subversion command line --quiet parameter for check out / update operations to help keep the output shorter. Prints nothing, or only summary information.

workspaceUpdater ----- WorkspaceUpdater

Possible values:

- updateUpdater
- checkoutUpdater
- noopUpdater
- <u>updateWithCleanUpdater</u>
- <u>updateWithRevertUpdater</u>

updateUpdater CONFIGURATOR

Implementation of <u>WorkspaceUpdater</u> Use 'svn update' as much as possible

checkoutUpdater CONFIGURATOR

Implementation of WorkspaceUpdater
Always check out a fresh copy

$noop Updater \ {}^{\texttt{CONFIGURATOR}}$

Implementation of <u>WorkspaceUpdater</u>
Do not touch working copy, it is updated by other script.

$update With Clean Updater \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

Implementation of WorkspaceUpdater

Emulate clean checkout by first deleting unversioned/ignored files, then 'svn update'

$update With Revert Updater \ ^{\texttt{CONFIGURATOR}}$

Implementation of WorkspaceUpdater

Use 'svn update' as much as possible, with 'svn revert' before update

CO	NFI	GI	TR A	TOR

credentialsId	String Select the credential from the list of relevant credentials in order to use that credential for
	checking out the source code.
– realm	String
	This is the realm that the SvnKit library associates with a specific checkout. For most
	Subversion servers this will typically be of the format $<$ scheme://hostname(:port)> name,
	while for servers accessed via svn+ssh it is of the format
	(username@)svn+ssh://hostname(:port).

CONFIGURATOR

 cancelProcessOnExternalsFail 	boolean
	Determines if the process should be cancelled when checkout/update svn:externals failed. Will
	work when "Ignore externals" box is not checked. Default choice is to cancelled the process
	when checkout/update svn:externals failed.
- credentialsId	String
depthOption	String

- --depth option for checkout and update commands. Default value is infinity.
 - empty includes only the immediate target of the operation, not any of its file or directory children.
 - files includes the immediate target of the operation and any of its immediate file children.
 - immediates includes the immediate target of the operation and any of its immediate file or directory children. The directory children will themselves be empty.
 - infinity includes the immediate target, its file and directory children, its children's children, and so on to full recursion.
 - as-it-is takes the working depth from the current working copy, allows for setting
 update depth manually using --set-depth option.

More information can be found here.

ignoreExternalsOption

boolean

"--ignore-externals" option will be used with svn checkout, svn update commands to disable externals definition processing.

More information can be found here.

Note: there is the potential to leverage syn:externals to gain read access to the entire Subversion repository. This can happen if you follow the normal practice of giving Jenkins credentials with read access to the entire Subversion repository. You will also need to provide

the credentials to use when checking/polling out the svn:externals using the *Additional Credentials* option.

local ----- String

Specify a local directory (relative to the workspace root) where this module is checked out. If

left empty, the last path component of the URL is used as the default, just like the svn CLI. A single period (.) may be used to check out the project directly into the workspace rather than into a subdirectory.

rear enrectory (remains to <u>the mornepast root</u>) where the mount to enterned out in

remote String

subversion CONFIGURATOR

Implementation of <u>SCMSource</u> Subversion

browser	SubversionRepositoryBrowser
credentialsId	String
excludes	String
— id	String
— includes	String
- remoteBase	String
– traits	list of SCMSourceTrait

Possible values:

- checkoutOptionTrait
- cloneOptionTrait
- submoduleOptionTrait
- localBranchTrait
- gitHubSshCheckout
- cleanMercurial
- cleanAfterCheckoutTrait
- cleanBeforeCheckoutTrait
- gitBrowser
- mercurialBrowser
- remoteName
- userIdentityTrait
- gitBranchDiscovery
- <u>gitHubBranchDiscovery</u>
- <u>discoverOtherRefsTrait</u>
- gitHubForkDiscovery
- gitHubPullRequestDiscovery
- gitTagDiscovery
- gitHubTagDiscovery
- <u>headRegexFilter</u>
- <u>headWildcardFilter</u>
- gitLFSPullTrait
- ignoreOnPushNotificationTrait
- $\bullet \ \ \underline{gitHubIgnoreDraftPullRequestFilter}$
- pruneStaleBranchTrait
- pruneStaleTagTrait
- gitTool
- mercurialInstallation
- <u>sparseCheckoutPathsTrait</u>
- refSpecs
- authorInChangelogTrait
- wipeWorkspaceTrait

workspaceUpdater WorkspaceUpdater

Possible values:

- updateUpdater
- checkoutUpdater
- noopUpdater

mercurialsubversion

- updateWithCleanUpdater
- updateWithRevertUpdater

legacySCM CONFIGURATOR

Implementation of <u>LibraryRetriever</u>

Legacy SCM

A relative path from the root of the SCM to the root of the library. Leave this field blank if the root of the library is the root of the SCM. Note that ".." is not permitted as a path component to avoid security issues.

- scm - SCM

Possible values:

• none
• cvs
• cvsProjectset

CONFIGURATOR

LibraryCachingConfiguration

excludedVersionsStr
 Space separated list of versions excluded from caching. Ex: "master release10 release9"
 refreshTimeMinutes
 int
 Determines the amount of time until the cache is refreshed. A value lower or equal to zero means that the cache will never be refreshed.

gitParameter ® ROOT ELEMENT

junitTestResultStorage ® ROOT ELEMENT

storage JunitTestResultStorage

Possible values:

• file

file CONFIGURATOR

Implementation of <u>JunitTestResultStorage</u> Local file system

nodeProperties ® ROOT ELEMENT

gitHubPluginConfig ROOT ELEMENT | configs | list of GitHubServerConfig

apiUrl

clientCacheSize

2/2//2021	nttps.//ci.my40
hookSecretConfigs	list of HookSecretConfi
hookUrl	String
CONFIGURATOR	
GitHub Server	

API endpoint of a GitHub server. To use public github.com, leave this field to the default value of https://api.github.com. Otherwise if you use GitHub Enterprise, specify its API endpoint here (e.g., https://ghe.acme.com/api/v3/).

, 5

int

String

Jenkins will use this much space, measured in megabytes, in \$JENKINS_HOME to cache data retrieved from GitHub API calls. A cache will help improve the performance by avoiding unnecessary data transfer, and by doing so it also makes it less likely to hit <u>API rate limit</u> (by the use of <u>conditional GET</u> calls).

In the unlikely event that cache is causing a problem, set this to 0 to disable cache altogether.

credentialsId String

You can create your own <u>personal access token</u> in your account GitHub settings. Token should be registered with scopes:

- admin:repo_hook for managing hooks (read, write and delete old ones)
- repo to see private repos
- · repo:status to manipulate commit statuses

In Jenkins, create credentials as «Secret Text», provided by Plain Credentials Plugin.

WARNING! Credentials are filtered on changing custom GitHub URL.

If you have an existing GitHub login and password you can convert it to a token automatically with the help of «Manage additional GitHub actions».

manageHooks ----- boolean

Will this configuration be used to manage credentials for repositories where it has admin rights? If unchecked, this credentials still can be used to manipulate commit statuses, but will be ignored to manage hooks.

name ----- String

An optional name to help with the disambiguation of API URLs. If you have multiple GitHub Enterprise servers with non-helpful names such as s21356.example.com and s21368.example.com then giving these names can help users when they need to select the correct server from a drop-down list. If you do not provide a name, then a "best guess" will be made from the hostname part of the API URL.

CONFIGURATOR

Hook secret configuration

ivyBuildTrigger ROOT ELEMENT - configurations list of IvyConfiguration - extendedVersionMatching boolean

CONFIGURATOR

IvyConfiguration

-	ivyCo	nfPath	String
L	name		String

gitHubConfiguration ® ROOT ELEMENT

apiRateLimitChecker ApiRateLimitChecker

Normalize API requests: Attempt to evenly distribute GitHub API requests. This strategy can be helpful when there are numerous pipelines that are managed by multiple users and/or there are many concurrent builds, making it difficult to know when a pipeline has been paused due to exceeding the GitHub API usage limit.

Throttle at/near rate limit: Restrict GitHub API requests only when near or above rate limit. This strategy may be preferred over "Throttle for Normalize" if there are relatively few/infrequent queries to the GitHub API.

Possible values:

- ThrottleForNormalize
- ThrottleOnOver
- NoThrottle

endpoints ----- list of Endpoint

CONFIGURATOR

apiUri String

GitHub API endpoint such as https://github.example.com/api/v3/.

name String

quietPeriod ROOT ELEMENT

dockerLabel

pollSCM PROOT ELEMENT pollingThreadCount into

pipeline-model-docker ORONT ELEMENT

The label expression to use when a pipeline runs with agent { docker ... } or agent { dockerfile ... } and no explicit label is provided. Leave empty to delegate to parent or system config.

— registry ----- DockerRegistryEndpoint

String

CONFIGURATOR

Docker Hub

credentialsIdurlString

URL to the Docker registry you are using. May be left blank to use the public DockerHub

builtInNode ® ROOT ELEMENT

globalConfigFiles ® ROOT ELEMENT

Possible values:

- globalMavenSettings
- mavenSettings
- properties
- j<u>son</u>
- mavenToolchains
- <u>xml</u>
- groovyScript
- custom
- groovyTemplate
- jellyTemplate
- <u>ivy</u>
- <u>npm</u>

$global Maven Settings \ ^{\text{CONFIGURATOR}}$

Implementation of **Config** Global Maven settings.xml

– comment	String
– content	String
– id	String
isReplaceAll	Boolean
	Replace all server credentials in this Maven settings.xml file, with the ones configured here
	(additional ones will be added). If a server in the settings.xml is not in the provided list of
	credentials, it will be removed.
name ·····	String
– providerId	String
- serverCredentialMappings	list of ServerCredentialMapping

CONFIGURATOR

credentialsId	String
	The credentials to be assigned to the defined serverId. The credentials can be the plaintext
	password or the encrypted version of it if you use a 'settings-security.xml' as described in
	Maven Password Encryption Guide
serverId	String
	The id of the server in the settings.xml, please see official Mayen documentation for more
	information.

If you add at least one credential mapping to this settings.xml, then all existing server definitions in the settings.xml will be removed and replaced by the ones defined here.

mavenSettings CONFIGURATOR Implementation of Config

Maven settings.xml

 comment
 String

 content
 String

 id
 String

 isReplaceAll
 Boolean

providerId ----- String

Replace all server credentials in this Maven settings.xml file, with the ones configured here (additional ones will be added). If a server in the settings.xml is not in the provided list of credentials, it will be removed.

name String

serverCredentialMappings ----- list of ServerCredentialMapping

properties CONFIGURATOR

Implementation of **Config**

Properties file

 comment
 String

 content
 String

 id
 String

 isReplaceAll
 Boolean

 name
 String

 propertiesCredentialMappings
 list of PropertiesCredentialMapping

 providerId
 String

CONFIGURATOR

${\bf Properties Credential Mapping}$

json Configurator

Implementation of **Config**

Json file

ŀ	comment	String
ŀ	content	String
ŀ	id	String
ŀ	name	String
-	providerId	String

mavenToolchains CONFIGURATOR

Implementation of <u>Config</u> Maven toolchains.xml

 comment
 String

 content
 String

 id
 String

 name
 String

providerId ----- String

xml CONFIGURATOR

Implementation of **Config**

Simple XML file

П	-		
	_	comment	String
	_	content	String
	_	id	String
	_	name ·····	String
	_	providerId	String

groovyScript CONFIGURATOR

Implementation of **Config**

Groovy file

comment	String
– content	String
– id	String
– name	String
providerId	String

custom CONFIGURATOR

Implementation of Config

Custom file

- commen	nt	String
- content		String
- customi	zedCredentialMappings	list of CustomizedCredentialMapping
_ id		String
name		String
– provide	rId	String

CONFIGURATOR

CustomizedCredentialMapping

credentialsId	- String	
	The credentials to be assigned to the token. This can be of the following types:	
	Username/password, SSH Username/privatekey, Secret text	
tokenKey	String	

The token that will be replaced with the credential information. The token is case sensitive and is traditionally represented as all uppercase.

A TOKEN for a username/password credential will be replaced in the following ways:

\$TOKEN will be replaced with username:password

\$TOKEN_PSW will be replaced with just the password

\$TOKEN_USR will be replaced with just the username

A token for an SSH username/keypair credential will be replaced in the following ways:

\$token will be replaced with the keypair or first key if loaded from file

\$token_USR will be replaced with the username

12/27/2021 https://ci.my4030.ir/configuration-as-code/reference $\textbf{groovyTemplate} \ ^{\texttt{CONFIGURATOR}}$ Implementation of **Config Extended Email Publisher Groovy Template** comment ---------- String content ----- String name ----providerId ----- String jellyTemplate CONFIGURATOR **Implementation of Config Extended Email Publisher Jelly Template** comment String content ----- String ----- String name ----- String providerId String ivy CONFIGURATOR **Implementation of** Config **Ivy configuration** ----- String comment ----- String contentString ----- String providerId String npm CONFIGURATOR Implementation of **Config** Npm config file comment String ----- String ----- String String providerId ----- String registries list of NPMRegistry CONFIGURATOR credentialsId String

The credentials to be assigned to the defined registry. The credentials can be the plaintext password or the encrypted version of it if you use a private registry (like Artifactory or Nexus) boolean hasScopes

String

https://ci.my4030.ir/configuration-as-code/reference

A space separated list of scope. A scope follows the usual rules for package names (url-safe characters, no leading dots or underscores).

In package names is preceded by an @-symbol and followed by a slash, e.g. @somescope/somepackagename. Scopes are a way of grouping related packages together, and also affect a few things about the way npm treats the package.

url ----- String

To resolve packages by name and version, npm talks to a registry website that implements the CommonJS Package Registry specification for reading package info.

The registry URL used is determined by the scope of the package (see npm-scope). If no scope is specified, the default registry is used, which is supplied by the registry config parameter

email-ext ROOT ELEMENT

-	addAccounts	list of MailAccount
-	adminRequiredForTemplateTesting	boolean
-	allowUnregisteredEnabled	boolean
-	allowedDomains	String
F	charset	String
F	debugMode	boolean
F	defaultBody	String
-	defaultClasspath	list of GroovyScriptPath
F	defaultContentType	String
F	defaultPostsendScript	String
F	defaultPresendScript	String
-	defaultRecipients	String
	defaultReplyTo	String
	defaultSubject	String
F	defaultSuffix	String
F	defaultTriggerIds	list of String
	emergencyReroute	String
-	excludedCommitters	String
	listId	String
F	mailAccount	MailAccount
	maxAttachmentSize	- long
F	maxAttachmentSizeMb	- long
F	precedenceBulk	boolean
-	watchingEnabled	boolean

CONFIGURATOR

address ----- String

advProperties	String	
- credentialsId	String	
- smtpHost	String	
- smtpPassword	String	
- smtpPort	String	
- smtpUsername	String	
– useSsl	boolean	
useTls	boolean	
CONFIGURATOR		
path	String	

$myView \ ^{\textcircled{\tiny ROOT\ ELEMENT}}$

upstream ® ROOT ELEMENT

globalUpstreamFilterStrategy ------ UpstreamFilterStrategy

Possible values:

- <u>UseGlobalSetting</u>
- <u>UseOldest</u>
- <u>UseNewest</u>

rebuildDescriptor ROOT ELEMENT

rocketChatNotifier ® ROOT ELEMENT

011010111111111111111111111111111111111	
- buildServerUrl	String
- channel	String
password	String
rocketServerUrl	String
- trustSSL	boolean
username	String
webhookToken	String
webhookTokenCredentialId	String

mavenModuleSet ® ROOT ELEMENT

	Possible values:
localRepository	LocalRepositoryLocator
– globalMavenOpts	String

- <u>default</u>
- perExecutor
- <u>perJob</u>

default CONFIGURATOR

Implementation of <u>LocalRepositoryLocator</u>

Default ("~/.m2/repository", or the value of 'localRepository' in Maven's settings file, if defined)

$perExecutor \ ^{\text{CONFIGURATOR}}$

Implementation of <u>LocalRepositoryLocator</u>

Local to the executor

perJob CONFIGURATOR

Implementation of <u>LocalRepositoryLocator</u>

Local to the workspace

$scmRetryCount \ ^{\textcircled{\tiny{POOT}\ ELEMENT}}$

$ivy Module Set \ ^{\textcircled{\tiny{POOT}\ ELEMENT}}$

globalAntOpts ---- String

$\textbf{casCGlobalConfig} \overset{\texttt{@}}{\text{ }} {}^{\texttt{ROOT} \, \texttt{ELEMENT}}$

configurationPath ----- String

gitSCM ROOT ELEMENT

- addGitTagAction bo	olean
- allowSecondFetch bo	olean
- createAccountBasedOnEmailbo	olean
- disableGitToolChooserbo	olean
- globalConfigEmail St	ring
- globalConfigName St	ring
- hideCredentials bo	olean
- showEntireCommitSummaryInChanges	boolea
- useExistingAccountWithSameEmail - bo	olean

mailer ROOT ELEMENT

- authentication	SMTPAuthentication
- charset	String
defaultSuffix	String
replyToAddress	String
- smtpHost	String
- smtpPort	String
– useSsl	boolean
– useTls	boolean

CONFIGURATOR

Use SMTP Authentication

password Secret

12/27/2021 String username fingerprints ROOT ELEMENT fingerprintCleanupDisabled boolean **FingerprintStorage** storage **Possible values:** • file file CONFIGURATOR Implementation of **FingerprintStorage Local Fingerprint Storage** viewsTabBar ® ROOT ELEMENT shell ROOT ELEMENT buildDiscarders ® ROOT ELEMENT configuredBuildDiscarders list of GlobalBuildDiscarderStrategy **Possible values:** • jobBuildDiscarder • simpleBuildDiscarder jobBuildDiscarder CONFIGURATOR Implementation of **GlobalBuildDiscarderStrategy Project Build Discarder** $simple Build Discarder \ ^{\texttt{CONFIGURATOR}}$ Implementation of <u>GlobalBuildDiscarderStrategy</u> **Specific Build Discarder** discarder BuildDiscarder **Possible values:** • logRotator $logRotator \ {\tt CONFIGURATOR}$ Implementation of **BuildDiscarder Log Rotation** artifactDaysToKeepStr ----- String artifactNumToKeepStr ----- StringString daysToKeepStr

envVarsFilter ® ROOT ELEMENT

numToKeepStr

----- String

warningsPlugin ® ROOT ELEMENT sourceDirectories list of SourceDirectory The warnings plugin copies all affected source code files to Jenkins build folder, in order to show them along with the issues in the user interface. If these files are not part of the workspace of a build then the warnings plugin does not show them: otherwise sensitive files could be shown by accident. You can provide a allowlist of additional source code directories that are allowed to be shown in Jenkins UI here. Note, that such a directory must be an absolute path on the agent that executes the build. CONFIGURATOR String path $subversion SCM \ ^{\textcircled{\tiny{0}}\ ROOT\ ELEMENT}$ globalExcludedRevprop ----- String workspaceFormat **buildStepOperation** ® ROOT ELEMENT enabled envInjectNodeProperty ® ROOT ELEMENT simple-theme-plugin ® ROOT ELEMENT elements list of ThemeElement Possible values: cssUrl **cssText** • faviconUrl • jsUrl cssUrl CONFIGURATOR Implementation of ThemeElement CSS URL ----- String cssText CONFIGURATOR Implementation of **ThemeElement** Extra CSS text String faviconUrl CONFIGURATOR Implementation of **ThemeElement** Favicon URL url jsUrl CONFIGURATOR Implementation of ThemeElement JavaScript URL

url

String

cVSSCM ROOT ELEMENT



declaredResources -----list of LockableResource

CONFIGURATOR

Resource

urce
description String
phemeral boolean
labels String
name String

reservedBy ----- Strir

usageStatistics PROOT ELEMENT

$administrative Monitors Configuration \ ^{\textcircled{\tiny ROOT \, ELEMENT}}$

$throttle Job Property ^{\textcircled{\tiny{ROOT}\ ELEMENT}}$

categories list of ThrottleCategory

CONFIGURATOR

categoryName String
maxConcurrentPerNode Integer
maxConcurrentTotal Integer
nodeLabeledPairs list of NodeLabeledPair

CONFIGURATOR

maxConcurrentPerNodeLabeled Integer
throttledNodeLabel String

$\textbf{defaultFolderConfiguration} \ ^{\textcircled{\tiny{0}}\ \text{ROOT}\ \text{ELEMENT}}$

healthMetrics list of FolderHealthMetric

Possible values:

- worstChildHealthMetric
- primaryBranchHealthMetric

worstChildHealthMetric CONFIGURATOR

Implementation of FolderHealthMetric

Child item with worst health

recursive boolean

Controls whether items within sub-folders will be considered as contributing to the health of this folder.

$\boldsymbol{primaryBranchHealthMetric}^{\hspace{1.5cm} \texttt{CONFIGURATOR}}$

Implementation of <u>FolderHealthMetric</u> Health of the primary branch of a repository

projectNamingStrategy ® ROOT ELEMENT

timestamper © ROOT ELEMENT

— allPipelines	when checked, timestamps will be enabled for all Pipeline builds. There is no need to use the timestamps {} step in Scripted, or the timestamps() option in Declarative.
— elapsedTimeFormat	The elapsed time format defines how the timestamps will be rendered when the elapsed time option has been selected. The Commons Lang DurationFormatUtils pattern is used. Default is: ' 'HH:mm:ss.S' '.
— systemTimeFormat	String The system clock time format defines how the timestamps will be rendered. The JDK SimpleDateFormat pattern is used.
	Default is: ' 'HH:mm:ss' ' For a more detailed format use: yyyy-MM-dd HH:mm:ss.SSS' '.

phaseJobsConfig ROOT ELEMENT

warningsParsers PROOT ELEMENT

consoleLogScanningPermitted	boolean
	Console Log parsing has to be processed directly on the Jenkins controller which means running the parser groovy code (configured here) directly on the controller which <i>CAN BE UNSAFE</i> from a security point of view.
	e.g. if your groovy code is poorly written.
	If you do not accept this risk then you should not enable this feature but if you do not enable this feature then the groovy parsers that you define will only be able to parse files in the build workspace (safely processed on the agents responsible for the builds rather than natively on the controller) rather than the build console log itself.
parsers	list of GroovyParser

CONFIGURATOR

example	String
– id	String
name ·····	String
– regexp	String

The regular expression has to be in the <u>Java Pattern format</u>. You should introduce capturing groups at least for the file name, the line number and the warning message. These groups can then be referenced by the Groovy script.

Example

Here is an example that you can use as a starting point for your regular expression: $^s*(.*): (d+):(.*):$

When this expression is matched against the string file/name/relative/unix:42:MyCategory: This is an example warning message then a match is found. The following capturing groups are available:

```
file/name/relative/unix
42
42
MyCategory
This is an example warning message
```

Summary of regular-expression constructs

Construct	Matches
Characters	
X	The character x
\\	The backslash character
\0 <i>n</i>	The character with octal value $0n$ (0 <= n <= 7)
\0 <i>nn</i>	The character with octal value $0nn$ (0 <= n <= 7)
∖ 0 <i>mnn</i>	The character with octal value $0mnn$ (0 <= m <= 3, 0 <= n <= 7)
\xhh	The character with hexadecimal value 0xhh
\u <i>hhhh</i>	The character with hexadecimal value 0xhhhh
\t	The tab character ('\u0009')
\n	The newline (line feed) character ('\u000A')
\r	The carriage-return character ('\u000D')
\f	The form-feed character ('\u000C')
\a	The alert (bell) character ('\u0007')
\e	The escape character ('\u001B')
\c <i>x</i>	The control character corresponding to x
Character classes	
[abc]	a, b, or c (simple class)
[^-h-1	

[ubc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
[a-z&&[def]]	d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
[a-z&&[^m-p]]	a through z, and not m through p: [a-lg-z](subtraction)

Predefined character classes

•	Any character (may or may not match <u>line terminators</u>)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

POSIX character classes (US-ASCII only)

\p{Lower} A lower-case alphabetic character: [a-z]

<pre>\p{Upper}</pre>	An upper-case alphabetic character: [A-Z]

 \p{ASCII} All ASCII: [\x00-\x7F]

 \p{Alpha} An alphabetic character: $[\p{Lower}\p{Upper}]$

\p{Digit} A decimal digit: [0-9]

 $\begin{array}{ll} & \text{An alphanumeric character:}[\p{Alpha}\p{Digit}] \\ & \text{Punctuation: One of } "#$%&' ()*+,-./:;<=>?@[\]^`{|}^- \\ \end{array}$

\p{Graph} A visible character: [\p{Alnum}\p{Punct}]
\p{Print} A printable character: [\p{Graph}\x20]

\p{Blank} A space or a tab: [\t]

 $\begin{array}{lll} \begin{tabular}{lll} $$ \Lambda$ control character: $[\x00-\x1F\x7F]$ \\ \begin{tabular}{lll} $$ A$ hexadecimal digit: $[0-9a-fA-F]$ \\ \begin{tabular}{lll} $$ A$ whitespace character: $[\t\x0B\f\r]$ \\ \end{tabular}$

java.lang.Character classes (simple java character type)

Classes for Unicode blocks and categories

\p{InGreek} A character in the Greek block (simple block)
\p{Lu} An uppercase letter (simple category)

\p{Sc} A currency symbol

\P{InGreek} Any character except one in the Greek block (negation) [\p{L}&&[^\p{Lu}]] Any letter except an uppercase letter (subtraction)

Boundary matchers

The beginning of a line
 The end of a line
 A word boundary
 A non-word boundary
 The beginning of the input
 The end of the previous match

\Z The end of the input but for the final <u>terminator</u>, if any

\z The end of the input

Greedy quantifiers

X? X, once or not at all X* X, zero or more times X+ X, one or more times X{n} X, exactly n times X{n, X, at least X times

Reluctant quantifiers

X?? X, once or not at all X*? X, zero or more times X+? X, one or more times X{n}? X, exactly x times x, at least x times

 $X\{n,m\}$? X, at least n but not more than m times

Possessive quantifiers

 $X\{n,m\}+$ X, at least n but not more than m times

Logical operators

XY X followed by Y $X \mid Y$ Either X or Y

(X) X, as a <u>capturing group</u>

Back references

	th		
\n	Whatever the n ^{ui} can	turing group matched	

Quotation

\ Nothing, but quotes the following character \Q Nothing, but quotes all characters until \E \E Nothing, but ends quoting started by \Q

Special constructs (non-capturing)

```
(?:X)X, as a non-capturing group(?idmsux-idmsux)Nothing, but turns match flags on - off(?idmsux-idmsux:X)X, as a non-capturing group with the given flags on - off(?=X)X, via zero-width positive lookahead(?!X)X, via zero-width negative lookahead(?<=X)</td>X, via zero-width positive lookbehind(?<!X)</td>X, via zero-width negative lookbehind(?>X)X, as an independent, non-capturing group
```

Backslashes, escapes, and quoting

The backslash character ('\') serves to introduce escaped constructs, as defined in the table above, as well as to quote characters that otherwise would be interpreted as unescaped constructs. Thus the expression \\ matches a single backslash and \{ matches a left brace.

It is an error to use a backslash prior to any alphabetic character that does not denote an escaped construct; these are reserved for future extensions to the regular-expression language. A backslash may be used prior to a non-alphabetic character regardless of whether that character is part of an unescaped construct.

Backslashes within string literals in Java source code are interpreted as required by the <u>Java Language Specification</u> as either <u>Unicode escapes</u> or other <u>character escapes</u>. It is therefore necessary to double backslashes in string literals that represent regular expressions to protect them from interpretation by the Java bytecode compiler. The string literal "\b", for example, matches a single backspace character when interpreted as a regular expression, while "\b" matches a word boundary. The string literal "\(hello\)" is illegal and leads to a compile-time error; in order to match the string (hello) the string literal "\\(hello\)" must be used.

Character Classes

Character classes may appear within other character classes, and may be composed by the union operator (implicit) and the intersection operator (&&). The union operator denotes a class that contains every character that is in at least one of its operand classes. The intersection operator denotes a class that contains every character that is in both of its operand classes.

The precedence of character-class operators is as follows, from highest to lowest:

```
    Literal escape \x
    Grouping [...]
    Range a-z
    Union [a-e][i-u]
    Intersection [a-z&&[aeiou]]
```

Note that a different set of metacharacters are in effect inside a character class than outside a character class. For instance, the regular expression . loses its special meaning inside a character class, while the expression - becomes a range forming metacharacter.

Line terminators

A *line terminator* is a one- or two-character sequence that marks the end of a line of the input character sequence. The following are recognized as line terminators:

- A newline (line feed) character ('\n'),
- A carriage-return character followed immediately by a newline character ("\r\n"),
- A standalone carriage-return character ('\r'),
- A next-line character ('\u0085'),
- A line-separator character ('\u2028'), or
- A paragraph-separator character ('\u2029).

If <u>UNIX_LINES</u> mode is activated, then the only line terminators recognized are newline characters.

The regular expression . matches any character except a line terminator unless the <u>DOTALL</u> flag is specified.

By default, the regular expressions ^ and \$ ignore line terminators and only match at the beginning and the end, respectively, of the entire input sequence. If MULTILINE mode is activated then ^ matches at the beginning of input and after any line terminator except at the end of input. When in MULTILINE mode \$ matches just before a line terminator or the end of the input sequence.

Groups and capturing

Capturing groups are numbered by counting their opening parentheses from left to right. In the expression ((A)(B(C))), for example, there are four such groups:

- 1 ((A)(B(C)))
- 2 (A)
- **3** (B(C))
- **4** (C)

Group zero always stands for the entire expression.

Capturing groups are so named because, during a match, each subsequence of the input sequence that matches such a group is saved. The captured subsequence may be used later in the expression, via a back reference, and may also be retrieved from the matcher once the match operation is complete.

The captured input associated with a group is always the subsequence that the group most recently matched. If a group is evaluated a second time because of quantification then its previously-captured value, if any, will be retained if the second evaluation fails. Matching the string "aba" against the expression (a(b)?)+, for example, leaves group two set to "b". All captured input is discarded at the beginning of each match.

Groups beginning with (? are pure, *non-capturing* groups that do not capture text and do not count towards the group total.

Unicode support

This class is in conformance with Level 1 of <u>Unicode Technical Standard #18: Unicode Regular Expression Guidelines</u>, plus RL2.1 Canonical Equivalents.

Unicode escape sequences such as \u2014 in Java source code are processed as described in \u2013.3 of the Java Language Specification. Such escape sequences are also implemented directly by the regular-expression parser so that Unicode escapes can be used in expressions that are read from files or from the keyboard. Thus the strings "\u2014" and "\\u2014", while not

equal, compile into the same pattern, which matches the character with hexadecimal value 0x2014.

Unicode blocks and categories are written with the \p and \P constructs as in Perl. \p{prop} matches if the input has the property prop, while \P{prop} does not match if the input has that property. Blocks are specified with the prefix In, as in InMongolian. Categories may be specified with the optional prefix Is: Both \p{L} and \p{IsL} denote the category of Unicode letters. Blocks and categories can be used both inside and outside of a character class.

The supported categories are those of <u>The Unicode Standard</u> in the version specified by the <u>Character</u> class. The category names are those defined in the Standard, both normative and informative. The block names supported by Pattern are the valid block names accepted and defined by <u>UnicodeBlock.forName</u>.

Categories that behave like the java.lang. Character boolean is method name methods (except for the deprecated ones) are available through the same \p{prop} syntax where the specified property has the name javamethod name.

script ----- String

The Groovy script works on the <u>Matcher</u> instance that is passed to the script in the variable matcher. The script must have a single return value of type

Optional<edu.hm.hafner.analysis.Issue>. Use the edu.hm.hafner.analysis.IssueBuilder instance that is passed to the script in the variable builder to create a new issue instance. Note that the method buildOptional() creates an Optional with a wrapped Issue instance, you rarely need to call the build() method that directly creates an Issue. If a false positive has been detected by the regular expression, you must return Optional.empty(). Additional available variables:

fileName

the name of the parsed report file

lineNumber

the current line number

Example

Here is an example that you can use as a starting point for your script. It composes a new warning using the matcher of the regular expression $^s(.*):(d+):(.*):$

artifactManager ® ROOT ELEMENT

artifactManagerFactories ----- list of ArtifactManagerFactory

globalDefaultFlowDurabilityLevel @ ROOT ELEMENT

durabilityHint **FlowDurabilityHint**

Possible values:

- PERFORMANCE_OPTIMIZED
- **SURVIVABLE NONATOMIC**
- MAX SURVIVABILITY

defaultView ROOT ELEMENT

plugin ROOT ELEMENT

location ROOT ELEMENT

adminAddress String Notification e-mails from Jenkins to project owners will be sent with this address in the from header. This can be just "foo@acme.org" or it could be something like "Jenkins Daemon <foo@acme.org>"

String

Optionally specify the HTTP address of the Jenkins installation, such as http://yourhost.yourdomain/jenkins/. This value is used to let Jenkins know how to refer to itself, ie. to display images or to create links in emails.

This is necessary because Jenkins cannot reliably detect such a URL from within itself.

ansiColorBuildWrapper ® ROOT ELEMENT

colorMaps	list of AnsiColorMap
globalColorMapName	String

ansiColorMap CONFIGURATOR

	Colorinap	
F	black	String
F	blackB	String
F	blue	String
F	blueB	String
F	cyan	String
F	cyanB	String
F	defaultBackground	Integer
L	defaultForeground	Integer
L	green	String
L	greenB	String
L	magenta	String
L	magentaB	String
L	name	String
L	red ·····	String

```
String
      redB
                                           String
      white
      whiteB
                                           String
      yellow
                                           String
      yellowB
                                           String
gradle ROOT ELEMENT
      installations
                                           list of GradleInstallation
gradle CONFIGURATOR
Implementation of ToolInstallation
Gradle
                                           String
      home
                                           String
      name
      properties
                                           list of ToolProperty
                                           Possible values:
                                              • installSource
                                           list of ToolProperty
      properties
                                           Possible values:
                                              • installSource
jdk © ROOT ELEMENT
      installations
                                           list of JDK
git ROOT ELEMENT
      installations
                                           list of GitTool
    CONFIGURATOR
git
Ğit
                                           String
                                           String
      home
                                           Set path to git executable. This can be just "git" or complete path.
      properties
                                           list of ToolProperty
      ROOT ELEMENT
ant
      installations
                    ----- list of AntInstallation
     CONFIGURATOR
Implementation of ToolInstallation
Ant
                                           String
      home
      name
                                           String
```

12/21/2021		nttps://ei.my+050.m/comiguration as coac/reference
proper	ties	list of ToolProperty
		Possible values:
		• installSource
DWODOW	ties	list of Tool Proporty
proper	ues	Possible values:
		rossible values:
		• installSource
.ella.	DOOT EL EN ENTE	
maven 🐠	ROOT ELEMENT	
– installa	tions	list of MavenInstallation
1		
maven COI	NFIGURATOR	
Implementati	on of <u>ToolInstallation</u>	
Maven		Okuina
- name		String
– proper	ties	list of ToolProperty
		Possible values:
		Locality and
		• installSource
— nroner	ties	list of ToolProperty
proper	ues	Possible values:
		1 double values.
		• <u>installSource</u>
• .• . 🛱 ROOT	ELEMENT	
Jare		
– installa	tions	list of ToolInstallation
		Possible values:
		• <u>ant</u>
		• dockerTool
		• <u>git</u>
		• g <u>radle</u>
		• <u>jdk</u>
		• <u>jgit</u>
		• jgitapache
		 maven mercurialInstallation
		• nodejs
		<u></u>
dockerTo	ol CONFIGURATOR	
Implementati	on of <u>ToolInstallation</u>	
Docker		
- home		
		A directory containing bin/docker. Or leave blank to use docker from \$PATH.
— name		String

When checked, Hudson will pass --debug when running most Mercurial commands. This can be useful if something is going wrong but you are having a hard time tracing what it is.

executable String Location of actual Mercurial executable. INSTALLATION will be substituted with installation directory. Normally INSTALLATION/bin/hg but you could specify another name such as INSTALLATION/my-mercurial.py or even look for Mercurial in the default path (leaving installation directory blank) using just hg. If you are configuring per-node tool home locations, you may leave the installation directory undefined on some nodes, in which case just hg (i.e., from \$PATH) is used. String home masterCacheRoot String Location of cached repositories on the controller node. Default: \$JENKINS_HOME/hgcache String name properties list of ToolProperty Possible values: • installSource list of ToolProperty properties **Possible values:** • installSource boolean useCaches When checked, Hudson will maintain a cache of the Mercurial repository in use by a project. The cache will be kept on the master node, updated on demand, and additional caches may be kept on agent nodes running this job. All jobs pointing to the same repository location will share a single cache (even if they use different named branches). If you have several jobs using the same repository location, or one job using multiple workspaces, enabling caching can substantially reduce the load on the Mercurial server and speed up Hudson builds and polling. boolean useSharing When checked, different clones of the same repository will share the history portion of the Mercurial. Enabling this option also automatically enables the caching option. If you have several jobs using the same repository location, or one job using multiple workspaces, enabling sharing should substantially reduce the disk usage on your nodes. The share extension is distributed with Mercurial 1.3 and later.

nodejs CONFIGURATOR

Implementation of <u>ToolInstallation</u> NodeJS

home String
name String
properties list of ToolProperty
Possible values:

installSource
properties list of ToolProperty

Possible values:

• installSource

mercurialInstallation	ROOT ELEMENT
-----------------------	--------------

installations -----list of MercurialInstallation

$maven Global Config \ {}^{\hbox{\scriptsize CONFIGURATOR}}$

globalSettingsProvider GlobalSettingsProvider
Possible values:

- standard
- filePath
- <u>mvn</u>

installations list of MavenInstallation

Possible values:

- standard
- <u>mvn</u>
- filePath

standard CONFIGURATOR

Implementation of <u>SettingsProvider</u> Use default maven settings

mvn CONFIGURATOR

Implementation of **SettingsProvider**

Provided settings.xml

settingsConfigId ----- String

filePath CONFIGURATOR

Implementation of <u>SettingsProvider</u> Settings file in filesystem

pathString

Path to settings.xml file, relative to project workspace or absolute (variables are supported).

standard CONFIGURATOR

Implementation of <u>GlobalSettingsProvider</u> Use default maven global settings

filePath CONFIGURATOR

Implementation of <u>GlobalSettingsProvider</u> Global settings file on filesystem

path ----- String

Path to settings.xml file, relative to project workspace or absolute (variables are supported).

mvn CONFIGURATOR

Implementation of <u>GlobalSettingsProvider</u> provided global settings.xml

settingsConfigId String

dockerTool ® ROOT ELEMENT

installations list of DockerTool

jgitapache ROOT ELEMENT

installations list of ToolInstallation

Possible values:

- <u>ant</u>
- dockerTool
- <u>git</u>
- gradle
- <u>jdk</u>
- jg<u>it</u>
- <u>jgitapache</u>
- maven
- mercurialInstallation
- <u>nodejs</u>

nodejs ROOT ELEMENT

installations ----- list of NodeJSInstallation