

---

# **LibreCores CI Tool Documentation**

*Release 2017.1rc7.dev9+g74cd227*

**Stefan Wallentowitz, LibreCores.org team**

**Sep 09, 2017**



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	API Documentation . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



Welcome to the LibreCores CI tool (`lcci`). This tool is used to setup and control agents connected to the LibreCores Continuous Integration (CI) infrastructure. Agents can be set up and connected to run jobs for either your own projects, any other project or with projects matching certain criteria.

To ensure agents share a common setup and to guarantee reproducible builds we use [Environment Modules](#). Tools can be installed in different versions and the build environment selects the proper version. We provide a set of [Docker](#) containers to automatically build and/or install the tools. Along with the tool itself the configuration files for Environment Modules are created (so called `modulefiles`).

In the projects' `Jenkinsfile` the modules are then accordingly loaded:

```
node('lcci-2017.1') {
  lcci.load(["eda/fusesoc/1.6.1",
           "eda/verilator/3.902"])

  ...
}
```



## API Documentation

### `lcci.configuration`

```
class lcci.configuration.Configuration (args)  
    Bases: object  
    docstring for Configuration  
    get_tool_repository (filter=None)  
    read_file (file)
```

### `lcci.tools`

```
class lcci.tools.Tool (name, basepath)  
  
    basepath = None  
    install (version)  
    lcci_releases = {}  
    name = None  
    version_in_release (version)  
    versions = {}  
class lcci.tools.ToolRepository (token, path, filter)  
  
    get_releases ()  
    get_tool (name)
```

```
gh = None
list (release=None)
path = None
tools = []
```

## lcci.agent

```
class lcci.agent.Agent (name, config, volumes={})
    Bases: object
    docstring for Agent.
    start (no_daemon=False)
    stop ()
```

## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



---

## Python Module Index

---

|

`lcci.agent`, 4  
`lcci.configuration`, 3  
`lcci.tools`, 3



## A

Agent (class in lcci.agent), 4

## B

basepath (lcci.tools.Tool attribute), 3

## C

Configuration (class in lcci.configuration), 3

## G

get\_releases() (lcci.tools.ToolRepository method), 3

get\_tool() (lcci.tools.ToolRepository method), 3

get\_tool\_repository() (lcci.configuration.Configuration method), 3

gh (lcci.tools.ToolRepository attribute), 3

## I

install() (lcci.tools.Tool method), 3

## L

lcci.agent (module), 4

lcci.configuration (module), 3

lcci.tools (module), 3

lcci\_releases (lcci.tools.Tool attribute), 3

list() (lcci.tools.ToolRepository method), 4

## N

name (lcci.tools.Tool attribute), 3

## P

path (lcci.tools.ToolRepository attribute), 4

## R

read\_file() (lcci.configuration.Configuration method), 3

## S

start() (lcci.agent.Agent method), 4

stop() (lcci.agent.Agent method), 4

## T

Tool (class in lcci.tools), 3

ToolRepository (class in lcci.tools), 3

tools (lcci.tools.ToolRepository attribute), 4

## V

version\_in\_release() (lcci.tools.Tool method), 3

versions (lcci.tools.Tool attribute), 3