

# Jenkins Windows Service - YAML Configuration Support

Google Summer of Code 2020

Buddhika Chathuranga

**DEVOPS  
WORLD**  
by CloudBees

# Hello World!

---

- I am Buddhika Chathuranga.
- From Sri Lanka.
- Undergraduate of University of Moratuwa.
- First time in Google summer of Code.
- First time in DevOps world conference.
- I love open source contributing.



# Windows Service Wrapper

## Current Implementation



# Windows Service Wrapper

---

- Windows service wrapper is a tool, which allows to run Jenkins as a service in a Windows machine.
- This feature is bundled into the Jenkins core.
- Windows service wrapper is configured by an XML file.
- Project Repository – [www.github.com/winsw/winsw](https://www.github.com/winsw/winsw)

# Configurations as an XML file

---

- XML file name should be same as the windows service wrapper executable name.
- XML file should be located inside the directory where windows service wrapper executable is located or in an ancestor directory.
- According to the current implementation user can't explicitly specify the XML file.
- Sample XML configurations file
  - <https://github.com/winsw/winsw/tree/master/examples>

# Sample XML configuration file

---

```
1 <service>
2   <id>jenkins</id>
3   <name>Jenkins</name>
4   <description>This service runs Jenkins automation server</description>
5   <env name="JENKINS_HOME" values="%LocalAppData%\Jenkins.jenkins"/>
6   <executable>Java</executable>
7   <arguments>-Xrs -Xmx256m -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle
8     --httpPort=8081 --webroot="%LocalAppData%\Jenkinswar"
9   </arguments>
10  <logmode>rotate</logmode>
11  <onfailure action="restart"/>
12
13  <extensions>
14    <extension
15      enabled="true"
16      className="winsw.Plugins.RunawayProcessKiller.RunawayProcessKillerExtension"
17      id="killOnStartup">
18      <pidfile>%LocalAppData%\Jenkinsjenkins.pid</pidfile>
19      <stopTimeOut>10000</stopTimeOut>
20      <StopParentFirst>>false</StopParentFirst>
21    </extension>
22  </extensions>
23 </service>
```

# Problems Identified

---

- No XML schema validation
- Configs are not validated on startup
- User cannot specify the configuration file explicitly
- Command line interface is less user friendly



# New features

**DEVOPS  
WORLD**  
*by CloudBees*



# New features

---

- Configuration as a YAML file
- New Command line interface
- Validate Configs on startup
  - XML schema validation

# Configuration as a YAML file

# Configuration as a YAML file

---

- Configuration will be provided by a YAML file.
- YAML is less verbose and more human readable.
- YAML is lightweight than XML and JSON.
- YAML is becoming more popular among configuration management tools.



# Sample YAML configuration file

---

```
1 id: jenkins
2 name: jenkins
3 description: This service runs Jenkins automation server
4 executable: Java
5 arguments: >-
6     -Xrs
7     -Xmx256m
8     -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle
9     --httpPort=8081
10    --webroot='%LocalAppData%\Jenkinswar'
11 env: - name: JENKINS_HOME
12     |   value: '%LocalAppData%\Jenkins.jenkins'
13 log:
14     |   logpath: '%LocalAppData%\Jenkins.log'
15     |   mode: rotate
16 onfailure: - action: restart
17             |   delay: 5 sec
18
```

# Current Status

---

- Parsing YAML config file
  - Merged
  - **YamlDotNet** library - <https://github.com/aaubry/YamlDotNet>
- Validate Configs on startup
  - Planning stage
- YAML schema validation
  - Planning stage
  - YAML config file will be validated with a JSON schema file

# New Command line interface



# New Command line interface

---

- According to the current implementation, there are only commands.
- There is no way to specify the configuration file explicitly.
- New CLI implemented in commands and options manner.
- User can specify the configuration file explicitly and skip schema validations.
- More user friendly.
- Command line parser library - <https://github.com/commandlineparser/commandline>

# Current Status

---

- Not merged yet.
- Will be merged with V3.
- Pull request - <https://github.com/winsw/winsw/pull/547>

# XML Schema validation



# XML Schema validation

---

- User can validate XML schema against an XSD file.
- XSD file will be shipped as an embedded resource.
- Status – Review stage
- Pull request - <https://github.com/winsw/winsw/pull/460>

# How to contribute

---

- Project Page
  - <https://www.jenkins.io/projects/gsoc/2020/projects/winsw-yaml-configs/>
- Project Repository
  - <https://github.com/winsw/winsw>
- Communication channel
  - <https://gitter.im/winsw/winsw>
- GSoC 2020 phase 1 demo
  - <https://www.youtube.com/watch?v=9qyo1f2rKQw&feature=youtu.be>
- GSoC 2020 phase 2 demo
  - <https://www.youtube.com/watch?v=9qyo1f2rKQw>

# Open pull requests

---

- YAML configuration unit test coverage
  - <https://github.com/winsw/winsw/pull/647>
- YAML support for extensions
  - <https://github.com/winsw/winsw/pull/638>
- New command line interface
  - <https://github.com/winsw/winsw/pull/547>
- XML schema validation
  - <https://github.com/winsw/winsw/pull/460>



# Special Thanks to

---

- Project Mentors
  - Oleg Nenashev
  - Next Turn
  - Michael Cirioli
- Jenkins Org admins

**Thank You**

**DEVOPS  
WORLD**  
by CloudBees