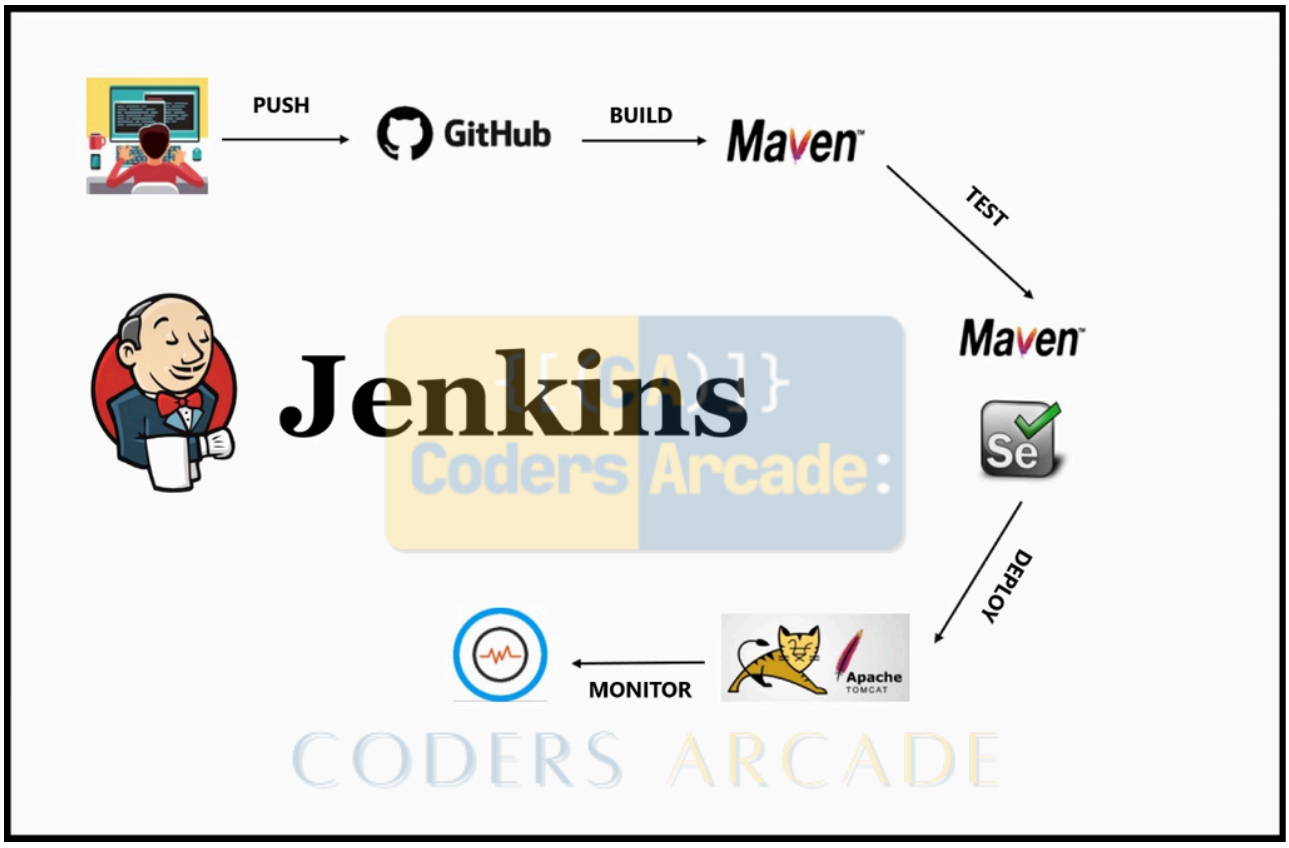
CA - Jenkins Notes & Documentation



What is Jenkins

- **Jenkins is a CI CD tool**

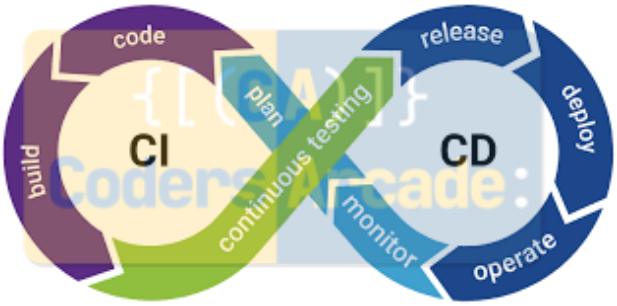
- **Free & Open Source**
- **Written in Java**

What is CI & CD

Watch This Video To Get Detailed Idea About CI/CD Pipeline : DevOps CI CD Pipeline || Simple & Detailed Explanation

Continuous Integration, Delivery & Deployment

CI/CD is **a method to frequently deliver apps to customers by introducing automation into the stages of app development**. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.



Installation

System Requirements

Memory 256 MB of RAM
Disk Space Depends on your projects
OS Windows, Mac, Ubuntu, Linux
Java 8 or 11 (JDK or JRE)

Installation on Windows

Watch This Video To Seamlessly Install Jenkins : Jenkins Installation - Step by Step Guide
Step 1 : Check Java is installed
Step 2 : Download Jenkins.war file
Step 3 : Goto cmd prompt and run command
java -jar jenkins.war --httpPort=8080
Step 4 : On browser goto http://localhost:8080
Step 5 : Provide admin password and complete the setup

Installation on Mac

Homebrew

Installation on Linux

How to install Jenkins on Amazon AWS EC2 Linux | 8 Steps

2. Jenkins Tutorials: How to install Jenkins on Ubuntu 22.04

Jenkins Configuration

How to change Home Directory

Step 1: Check your Jenkins Home > Manage Jenkins > Configure System
Step 2 : Create a new folder
Step 3 : Copy the data from old folder to new folder
Step 4 : Create/Update env variable JENKINS_HOME
Step 5 : Restart Jenkins
jenkins.xml
JENKINS_HOME

How to setup Git on Jenkins

Step 1 : Goto Manage Jenkins > Manage Plugins
Step 2 : Check if git is already installed in Installed tab
Step 3 : Else goto Available tab and search for git
Step 4 : Install Git
Step 5 : Check git option is present in Job Configuration

Create the first Job on Jenkins

How to connect to Git Remote Repository in Jenkins (GitHub)

Step 1 : Get the url of the remote repository
Step 2 : Add the git credentials on Jenkins
Step 3 : In the jobs configuration goto SCM and provide git repo url in git section
Step 4 : Add the credentials
Step 5 : Run job and check if the repository is cloned

How to use Command Line in Jenkins CLI

Faster, easier, integration
Step 1 : start Jenkins
Step 2 : goto Manage Jenkins - Configure Global Security - enable security
Step 3 : goto - http://localhost:8080/cli/
Step 4 : download jerkins-cli jar. Place at any location.
Step 5 : test the jenkins command line is working

java -jar jenkins-cli.jar -s http://localhost:8080 /help --username <userName> --password <password>

How to create Users + Manage + Assign Roles

How to create New Users
How to configure users

How to create new roles
How to assign users to roles
How to Control user access on projects

Step 1 : Create new users
Step 2 : Configure users
Step 3 : Create and manage user roles Role Based Authorization Strategy Plugin - download - restart jenkins
Step 4 : Manage Jenkins - Configure Global Security - Authorization - Role Based Strategy
Step 5 : Create Roles and Assign roles to users
Step 6 : Validate authorization and authentication are working properly

Jenkins Pipeline Beginner Tutorial

How to create Jenkinsfile

- **What is pipeline**
- **What is jenkins pipeline**
- **What is jenkinsfile**
- **How to create jenkinsfile**

Build > Deploy > Test > Release
Jenkinsfile : Pipeline as a code

Step 1 : Start Jenkins
Step 2 : Install Pipeline Plugin
Step 3 : Create a new job
Step 4 : Create or get Jenkinsfile in Pipeline section
Step 5 : Run and check the output

Jenkins Pipeline
How to get jenkinsfile from Git SCM

Step 1 : Create a new job or use existing job (type : Pipeline)
Step 2 : Create a repository or GitHub
Step 3 : Add Jenkinsfile in the repo
Step 4 : Under Jenkins job > Pipeline section > Select Definition Pipeline script from SCM
Step 5 : Add repo and jenkinsfile location in the job under Pipeline section
Step 6 : Save & Run

Jenkins Pipeline
How to clone a git repo using Jenkinsfile

Show Live Demonstration.

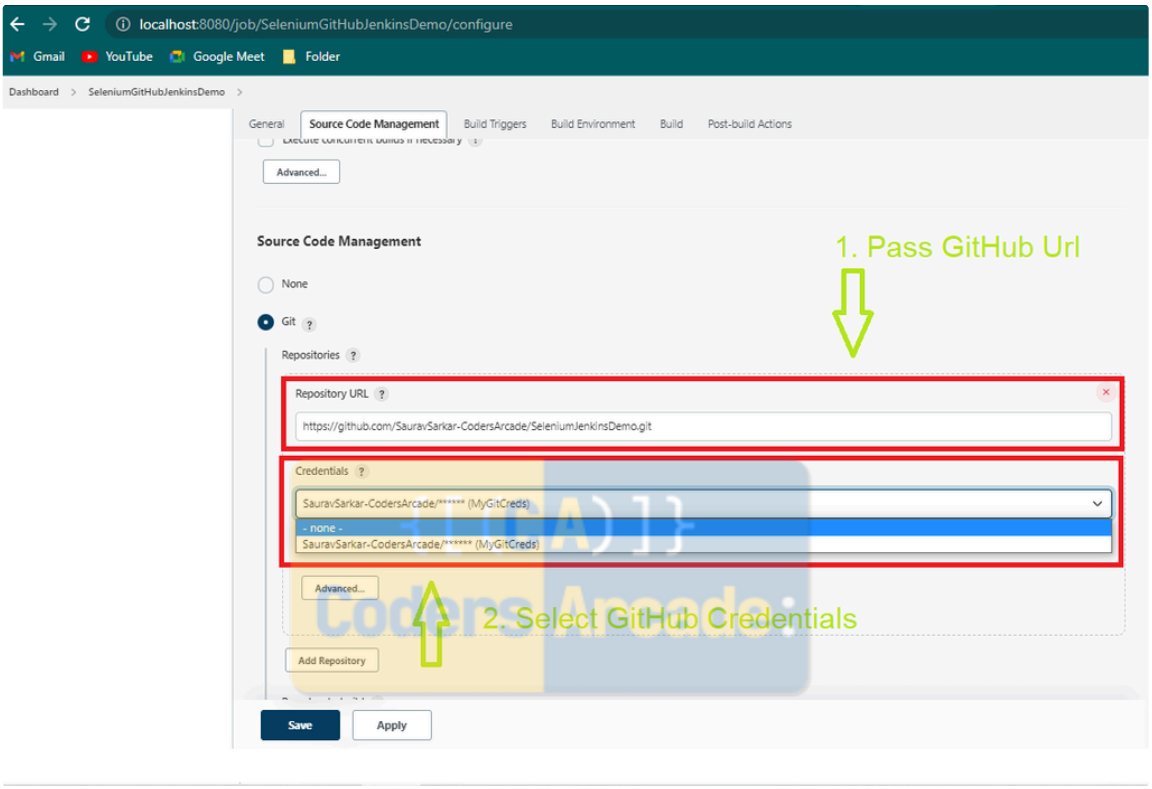
Running Selenium Automation Tests from GitHub via Jenkins CI Agent :

There are two ways in which we can perform Selenium Automation tests from GitHub via Jenkins

- Via **Git Credentials**
- Via **GitHub access token**
- The steps are shown with images.
- In the build, you have to execute the maven commands like: **mvn clean test, etc.**

Method 1:

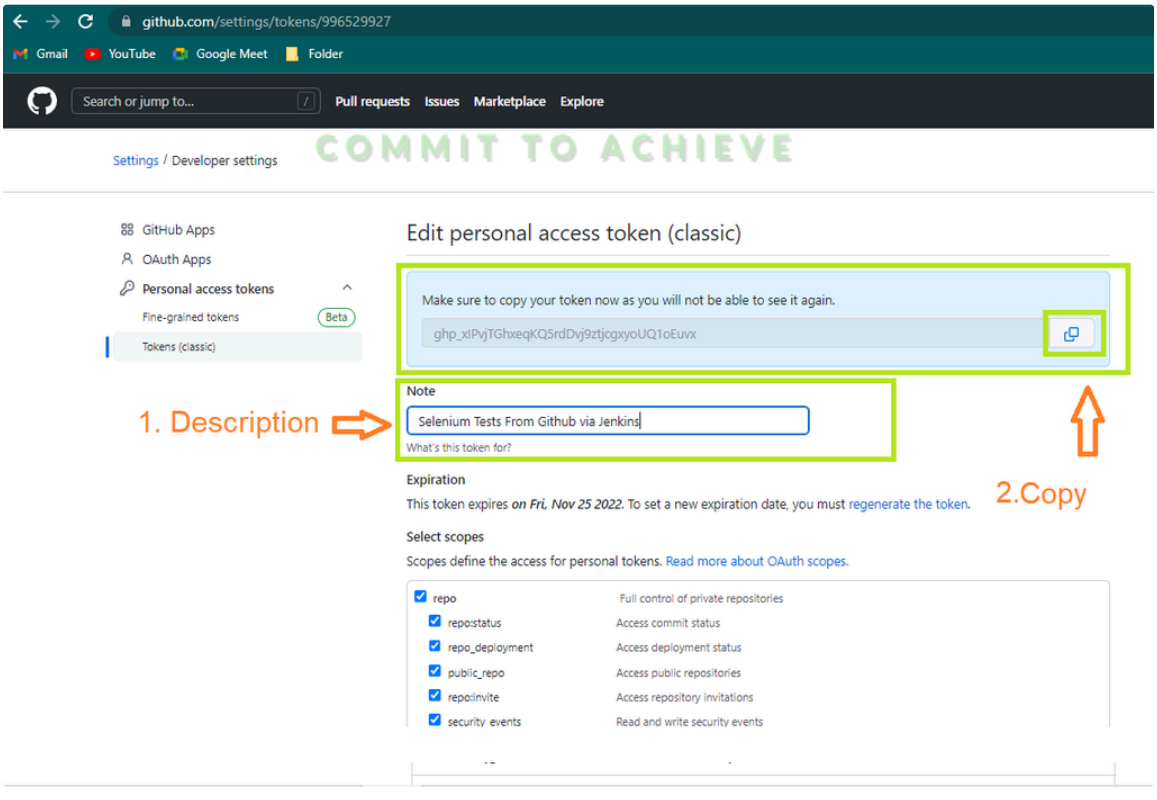
- Via **Git Credentials**



GitHub + Jenkins With Git Credentials

Method 2:

- Via **GitHub Access Token**



GitHub + Jenkins Via Access Token

CODERS ARCADE

COMMIT TO ACHIEVE

CA - Experiment 5 - Introduction to Jenkins: What is Jenkins?, Installing Jenkins on Local or Cloud Environment, Configuring Jenkins for First Use

Experiment 5: Introduction to Jenkins

🌟 Objective

To understand the fundamentals of **Jenkins**, install it on a local or cloud environment, and configure it for first-time use.

🔹 1. What is Jenkins?

Jenkins is an **open-source automation server** used for:
✅ **Continuous Integration (CI)** – Automatically testing and integrating code changes
✅ **Continuous Deployment (CD)** – Automating application deployment
✅ **Building Pipelines** – Managing end-to-end software development workflows
✅ **Plugin-Based Extensibility** – Supporting tools like Maven, Gradle, Ansible, Docker, and Azure DevOps

🚀 Why Use Jenkins?

✔ Automates builds and tests
✔ Reduces manual intervention
✔ Improves software quality
✔ Works with multiple tools and platforms

🔹 2. Installing Jenkins

Jenkins can be installed using multiple methods:
1️⃣ **Windows Installer (.msi) – Recommended for Windows**
2️⃣ **Linux Package Manager – Best for Linux Users**
3️⃣ **Jenkins WAR File – Universal method using Java**

We'll cover all three approaches.

🟢 2.1 Prerequisites

🔹 **Java 11 or 17** is required for Jenkins.
Check Java version:

```
java -version

```

If Java is not installed, download it from:
⛀ Download the Latest Java LTS Free

🟠 2.2 Installing Jenkins on Windows (MSI Installer) – Recommended

✅ **Step 1: Download Jenkins**

🔗 **Download from:** Download and deploy
Choose **Windows Installer (.msi)** for an easy setup.

✅ **Step 2: Install Jenkins**

1️⃣ Run the downloaded `.msi` file.
2️⃣ Follow the installation wizard.
3️⃣ Select **Run Jenkins as a Windows Service** (recommended).
4️⃣ Choose the **installation directory** (default: `C:\Program Files\Jenkins`).
5️⃣ Click **Install** and wait for the setup to complete.

✅ **Step 3: Start Jenkins**

1️⃣ Open **Services** (`services.msc`) and ensure **Jenkins** is running.
2️⃣ Open a web browser and go to:

```
http://localhost:8080

```


✅ **Step 4: Unlock Jenkins**

1️⃣ Find the initial **Admin Password** in:

```
C:\Program Files\Jenkins\secrets\initialAdminPassword

```

2️⃣ Copy the password and paste it into the Jenkins setup page.

✅ **Step 5: Install Recommended Plugins**

Jenkins will prompt you to install plugins. Click **"Install Suggested Plugins"**.

✅ **Step 6: Create Admin User**

1️⃣ Set up a **Username**, **Password**, and **Email**.
2️⃣ Click **Save and Finish**.

🔹 **Jenkins is now ready!** 🎉

Access it anytime at:

```
http://localhost:8080

```

🟢 2.3 Installing Jenkins on Linux (Ubuntu/Debian)

✅ **Step 1: Add Jenkins Repository**

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

```

✅ **Step 2: Install Jenkins**

```
sudo apt update
sudo apt install jenkins -y

```

✅ **Step 3: Start Jenkins**

```
sudo systemctl start jenkins
sudo systemctl enable jenkins

```

✅ **Step 4: Access Jenkins**

Find the **initial password** in:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

```

Then open Jenkins in a browser:

```
http://localhost:8080

```

🟠 2.4 Installing Jenkins Using WAR File (Works on Any OS)

This method allows you to run Jenkins without installing it as a service.

✅ **Step 1: Download the Jenkins WAR File**

🔗 Download from: Download and deploy
Choose **Generic Java Package (.war)**.

✅ **Step 2: Run Jenkins Using Java**

Navigate to the folder where the `.war` file is downloaded and run:

```
java -jar jenkins.war --httpPort=8080

```

🔹 This will start Jenkins on port **8080**.

✅ **Step 3: Open Jenkins in Browser**

Go to:

```
http://localhost:8080

```

✅ **Step 4: Unlock Jenkins & Setup**

Follow the **same steps** as the Windows/Linux installation:
✔ Find the initial password
✔ Install plugins
✔ Create an admin user

🔹 **Jenkins is now running without installation!**

To stop Jenkins, press **CTRL + C** in the terminal.

🔹 3. Configuring Jenkins for First Use

✅ 3.1 Understanding the Jenkins Dashboard

After logging in, you will see:
🔹 **New Item** → Create Jobs/Pipelines

- 🔹 **Manage Jenkins** → Configure System, Users, and Plugins
- 🔹 **Build History** → View previous builds
- 🔹 **Credentials** → Store secure authentication details

✅ 3.2 Installing Additional Plugins

Jenkins supports **plugins** for various tools like Maven, Gradle, Docker, and Azure DevOps.

🔹 To install a plugin:
1️⃣ Go to **Manage Jenkins → Manage Plugins**
2️⃣ Search for the required plugin
3️⃣ Click **Install without Restart**

✅ 3.3 Setting Up Global Tool Configuration

Configure Java, Maven, and Gradle in Jenkins:
1️⃣ Go to **Manage Jenkins → Global Tool Configuration**
2️⃣ Add paths for:

- **JDK** (`C:\Program Files\Java\jdk-17`)
- **Maven** (`C:\Maven\apache-maven-<version>`)
- **Gradle** (`C:\Gradle\gradle-<version>`)

3️⃣ Click **Save**

📌 Assessment Questions

1️⃣ What is **Jenkins** used for in DevOps?
2️⃣ Explain the **difference** between CI and CD in Jenkins.
3️⃣ How do you **install Jenkins** on Windows, Linux, and using the WAR file?
4️⃣ Where can you find the **initial Jenkins password** after installation?
5️⃣ What are some **essential Jenkins plugins** for automation?

📌 Important Note

You can use the below links to easily install **Jenkins** & understand **CI/CD Pipelines** in **DevOps**.

🔹 **Jenkins Installation Video** 🎥 – Follow this step-by-step guide for a **seamless Jenkins setup**: Jenkins Installation - Step by Step Guide

🔹 **Understanding CI/CD in DevOps** 🚀 – Learn how Jenkins fits into the **CI/CD pipeline**: DevOps CI CD Pipeline || Simple & Detailed Explanation

Ensure you have **Java installed** before setting up Jenkins. If you face any issues, check the **Jenkins logs** for troubleshooting. ✅

CA - Experiment 6 - Continuous Integration with Jenkins: Setting Up a CI Pipeline, Integrating Jenkins with Maven/Gradle, Running Automated Builds and Tests

---

Experiment 6: Continuous Integration with Jenkins

Objective

To set up a Continuous Integration (CI) pipeline in Jenkins, integrate it with Git, and run Selenium Java tests using Maven.

Prerequisites

Before proceeding, ensure the following:

✅ **Jenkins is installed and running** on your system. If not, refer to [Experiment 5].
✅ **Git is installed** and configured in Jenkins. (Verify with `git --version`).
✅ **Maven is installed** and configured in Jenkins. (Check with `mvn -version`).
✅ **Selenium Maven Project is ready** with test cases (`src/test/java`).
✅ **Project is stored in two places**:

- Locally on your system (e.g., `D:\Idea Projects\MVNGRDLDEMO`).
- Pushed to **GitHub** with a valid repository link.
  ✅ **Jenkins has access to the GitHub repository** (via credentials).

---

1. Configuring Jenkins & Git Integration

Step 1: Verify Git Installation in Jenkins

1. Open **Jenkins Dashboard** → **Manage Jenkins** → **Global Tool Configuration**.
2. Under **Git**, verify the installation path (e.g., `C:\Program Files\Git\bin\git.exe`).
3. Click **Save**.

Step 2: Add GitHub Credentials in Jenkins

1. Navigate to **Manage Jenkins** → **Manage Credentials**.
2. Select **Global credentials (unrestricted)** → Click **Add Credentials**.
3. Choose **Username with password** or **SSH Key**, provide details, and click **OK**.

---

2. Running a Selenium Java Test from a Local Maven Project

Step 1: Create a New Jenkins Job

1. Go to **Jenkins Dashboard** → Click **New Item**.
2. Enter a project name → Select **Freestyle Project**.
3. Click **OK**.

Step 2: Configure the Build Step

1. Scroll to **Build** → Click **Add build step** → **Execute Windows Batch Command**.
2. Enter the following commands (**ensure correct navigation to project directory**):

```
cd D:\Idea Projects\MVNGRDLDEMO
mvn test

```

3. Click **Save** → Click **Build Now** to execute the test.

---

3. Running Selenium Tests from a GitHub Repository via Jenkins

Step 1: Set Up a New Jenkins Job for GitHub Project

1. Go to **Jenkins Dashboard** → Click **New Item**.
2. Enter a project name → Select **Freestyle Project**.
3. Click **OK**.

Step 2: Configure Git Repository in Jenkins

1. Under **Source Code Management**, select **Git**.
2. Enter your GitHub repository URL (e.g., `https://github.com/your-repo-name.git`).
3. Select the **Git credentials** configured earlier.

Step 3: Add Build Step for Maven

1. Scroll to **Build** → Click **Add build step** → **Execute Windows Batch Command**.
2. Enter the Maven test command:

```
mvn test

```

3. Click **Save**.

Step 4: Trigger the Build

1. Click **Build Now** to fetch the code from GitHub and execute the Selenium tests.
2. Check the **Console Output** to verify test execution.

Important Notes

📌 **Prerequisites are crucial!** Make sure Jenkins, Git, Maven, and Selenium projects are set up correctly before proceeding.
📌 **Always navigate to the project directory** before running `mvn test` from a local system.
📌 **Use webhooks** in GitHub to automatically trigger builds when new code is pushed.
📌 **Configure email notifications** in Jenkins for build status updates.

🔗 **Jenkins & Git Integration Video:** [To Be Added - Version 1.2]
🔗 **Running Selenium Tests in Jenkins:** [To Be Added - Version 1.2]
