**Build and Release Engineer Interview Questions & Answers**

**What Is The Best Practice Configuration Usage For Files - Pom.xml Or Settings.xml ?**

The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.

**How Can I Change The Default Location Of The Generated Jar When I Command "mvn Package"?**

By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.

**What Is Maven's Order Of Inheritance?**

parent pom

project pom

settings

CLI parameters

**How Do I Determine Which Pom Contains Missing Transitive Dependency?**

run mvn -X

**What Is The Difference Between Compile And Install ?**

Compile compiles the source code of the project

whereas

Install installs the package into the local repository, for use as a dependency in other projects locally

**What Is The Use Of !! Command ? Can I Use It With Conjunction To Some Other String To Complete A Command ?**

Its used to execute last command. Yes this can be used with other string to execute new command. For eg - if ls was the last command, We can execute !! -l for having the long listing.

**Which Version Control (vc) Or Software Configuration Management (scm) Systems Work With Merge?**

You can use Tortoise SVN,which has Merge Utility embedded in it.

**What Is A Transitive Dependency ? Can We Override Transitive Dependency Version And If Yes, How ?**

Transitive dependency is the dependencies not defined directly in the current POM but the POM of the dependent projects.

Yes we can override transitive dependency version by specifying the dependency in the current POM.

**What Are The Benefits Of Transitive Dependency In Maven ?**

Transitive dependencies allows to avoid specifying the libraries that are required by the project which are specified in other dependent projects - Remote or Local.

**What Is A Cyclic Dependency ?**

A has dependency of B, B has dependency of C and C has dependency of A,architecture,technical lead

With Maven 2 , came transitive dependency wherein in above scenario, C will acts as a dependency of A as if this dependency has been defined directly in A but the negative side is that if it leads to cyclic dependency , it creates problems.

**What Technologies Have You Worked With For Build Management ?**

Ant and Maven

**Have You Created Any Build Script Yourself ?**

Yes, I have worked on many build scripts in last few years.

**Difference Between Jar , War And Ear ?**

Jar is Java Archive i.s compressed Class or Class / Java files.

War comprises of compressed Servlet class files,JSP FIles,supporting files, GIF and HTML files.

Ear comprise of compressed Java and web module files ( was files ).

**Which Version Control System You Are Using In Your Current Project ?**

We are using SVN and Git Hub.

**What Is Repository ?**

Repository is the heart of any version control system. It is central place where developers store all their work. Repository not only stores files but also history. Repository is accessed over a network, with repository acting as a server and version control tool acting as a client. Client can connect to repository, and then they can store/retrieve their changes to/from repository.

**What Is Jenkins ?**

It is a continuous integration tool written in Java.

**What Is The Difference Between Maven, Ant And Jenkins ?**

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

**Which Scm Tools Jenkins Supports ?**

AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC

**What Are The Various Ways In Which Build Can Be Scheduled In Jenkins ?**

Builds can be triggered by source code management  commits.

Can be triggered after completion of other builds.

Can be scheduled to run at specified time ( crons )

Manual Build Requests

**What Is A Transitive Dependency? Can We Override Transitive Dependency Version And If Yes, How?**

Transitive dependency is the dependencies not defined directly in the current POM but the POM of the dependent projects.

Yes, we can override transitive dependency version by specifying the dependency in the current POM.

**What Is The Use Of !! Command? Can I Use It With Conjunction To Some Other String To Complete A Command?**

It’s used to execute last command. Yes, this can be used with other string to execute new command. For eg – if ls was the last command, we can execute !! -l for having the long listing.

**What Factors Influence The Opening Of A Feature Branch?**

Typically, feature branches are created in cases where the new feature or enhancement has broad-sweeping changes to the code base such that introducing them in the trunk may be too disruptive. Also, feature branches may be used for prototyping or proof-of-concept for code that may never end up in trunk.

**What Is The Best Practice Configuration Usage For Files – Pom.xml Or Settings.xml?**

The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.

**How Can I Change The Default Location Of The Generated Jar When I Command “mvn Package”?**

By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.

**What Is Maven’s Order Of Inheritance?**

parent pom

project pom

settings

CLI parameters

**What Is The Purpose Of Continuous Integration For A Development Team?**

The primary purpose of CI is to provide regular, fast feedback to developers as they commit changes to the shared code repository (VCS).

The idea being that we’re always integrating our code on commit, so that when conflicts arise, they can be addressed more quickly and easily than if the changes had been made days, week, or even months ago.

**How Is Attenuation Happens From Jenkins To Remote Server?**

we have already like password less log in enable, so from Jenkins server any one login to the remote the Linux server and all without asking the password the key exchange is already done.

# **GIT Interview Questions**

**1. What is GIT ?**

GIT is a broadcasted adaptation administration structure and source code administration (SCM) frame including an accentuation to trade with small and valuable businesses by activity and proficiency.

**2. What is a repository in GIT?**

An archive contains a registry named git, where git keeps the greater part of its metadata for the storehouse. The substance of the git registry is private to git.

**3. How can we know if a branch is already merged into master in GIT ?**

We can utilize following orders for this reason:

**git branch** **– blended ace:** This prints the branches converged into an ace

**git branch – blended records**: This prints the branches converged into HEAD (i.e. the tip of the current branch)

**git branch – no-consolidated**: This prints the branches that have not been blended

As a matter, of course, this applies just to nearby offices.

We can utilize – a banner to demonstrate both nearby and remote branches.Or on the other hand, we can utilize – r banner to indicate just the remote branches.[Top 50 Devops Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-50-devops-interview-questions-and-answers-pdf)

**4. What is the purpose of git stash drop ?**

 On the off chance that we needn’t bother with a particular reserve, ([Salesforce Training](https://www.svrtechnologies.com/salesforce)) we utilize git stash drop charge to expel it from the rundown of supplies.

As a matter of course, this summon expels to most recent included reserve

To evaluate a particular reserve we indicate as a contention in the git stash drop charge.

**5. What is the HEAD in GIT ?**

AHEAD is a reference to the present looked at conferring.

It is a representative reference to the branch that we have looked at.

At any given time, one head is chosen as the ‘present head’ this head is otherwise called HEAD (dependably in capitalized).

**6. What is the most popular branching strategy in GIT ?**

There are numerous approaches to do stretching in GIT.One of the well-known routes is to keep up two branches:

**ace:** This branch is utilized for a generation. In this branch HEAD is a dependably underway prepared state.

**build up:** This branch is utilized for improvement. In this branch, we store the most recent code created in a venture.

This is work in advance code.Once the code is prepared for sending to creation, it is converted into the ace branch from creating a branch.

**7. What is SubGit ?**

SubGit is programming apparatus utilized for relocating SVN to Git. It is anything but difficult to utilize. By utilizing this we can make a writable Git reflection of a Subversion store.

It makes a bi-directional mirror that can be utilized for pushing to Git and in addition focusing on Subversion.

SubGit additionally deals with synchronization amongst Git and Subversion.

**8. What is the use of git instaweb ?**

Git-instaweb is a content by which we can peruse a git archive in a web program. It sets up the gitweb and a web-server that makes the working vault accessible on the web.

**9. What are git hooks ?**

Git snares are contents that can run consequently in the event of an occasion in a Git store. These are utilized for robotization of the work process in GIT.

Git snares likewise help in altering the inward conduct of GIT.

These are for the most part utilized for implementing a GIT confer arrangement.

**10. What are the main benefits of GIT ?**

There are following primary advantages of GIT:

⦁ **Distributed System:** GIT is a Distributed Version Control System (DVCS). So you can keep your private work in adaptation control yet totally escaped others. You can work disconnected too.

⦁ **Flexible Workflow:**GIT enables you to make your own work process. You can utilize the procedure that is appropriate for your venture. You can go for brought together or ace slave or some other work process.

⦁ **Fast:** GIT is quick when contrasted with other form control frameworks.

⦁ **Data Integrity:** Since GIT utilizes SHA1, information isn’t less demanding to degenerate.

⦁ **Free:** It is free for individual utilize. Such huge numbers of beginners utilize it for their underlying activities. It likewise works exceptionally well with substantial size task.

⦁ **Collaboration:** GIT is anything but difficult to use for ventures in which joint effort is required. Numerous prevalent open source programming over the globe utilize GIT.

**11. What are the disadvantages of GIT ?**

GIT has not very many weaknesses. These are the situations when GIT is hard to utilize. Some of these are:

**Binary Files:** If we have a considerable measure double records (non-content) in our venture, at that point GIT turns out to be moderate. E.g. Tasks with a lot of pictures or Word records.

**Steep Learning Curve:** It sets aside some time for a newcomer to learn GIT. A portion of the GIT summons is non-instinctive to a fresher.

**Slow remote speed:** Sometimes the utilization of remote stores in ease back because of system dormancy. Still, GIT is superior to different VCS in speed. [Top 50 DevOps Engineer Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-50-devops-engineer-interview-questions-and-answers-pdf)

**12. How will you start GIT for your project ?**

We utilize git init order in a current venture catalog to begin form control for our undertaking. After this, we can utilize git add and git confer orders to add records to our GIT archive.

**13. What is git clone in GIT ?**

In GIT, we utilize git clone summon to make a duplicate of a current GIT archive in our nearby.

This is the most prevalent approach to make a duplicate of the archive among designers.

It is like svn checkout. In any case, for this situation, the working duplicate is an undeniable archive.

**14. How will you create a repository in GIT ?**

To make another archive in GIT, first, we make an index for the venture. At that point, we run ‘git init’ charge. Presently, GIT makes the .git index in our venture catalog. This is the manner by which our new GIT store is .

**15. What are the different ways to start work in GIT ?**

We can begin work on GIT in following ways:

**New Project:** To make another storehouse we utilize git init order.

**Existing Project:** To chip away at a current storehouse we utilize git clone order.

**16. GIT is written in which language ?**

Most of the GIT circulations are composed in C dialect with Bourne shell. A portion of the charges is composed in Perl dialect.

**17. What does ‘git pull’ command in GIT do internally ?**

In GIT, git pull inside completes a git get first and after that completes a git blend.

So pull is a blend of two orders: bring and combine.

We utilize git pull order to convey our neighborhood office fully informed regarding its remote adaptation.

**18. What is git stash ?**

In GIT, now and again we would prefer not to submit our code yet we would prefer not to lose additionally the incomplete code. For this situation we utilize git stash summon to record the present condition of the working registry and list in a reserve. This stores the incomplete work in a reserve and cleans the present branch from uncommitted changes.

Presently we can chip away at a perfect working index.

Later we can utilize the reserve and apply those progressions back to our working index.

On occasion we are amidst some work and would prefer not to lose the incomplete work, we utilize git stash order.

**19. What is the meaning of ‘stage’ in GIT ?**

In GIT, arrange is a stage before confer. To arrange implies that the records are prepared for submission.

Let say, you are dealing with two highlights in GIT. One of the highlights is done and the other isn’t yet prepared. You need to confer and leave for home at night. Yet, you can confer. ([sap training](https://svrtechnologies.com/sap-training)) since the two are not completely prepared. For this situation, you can simply organize the element that is prepared and confer that part. The second element will stay as work in advance.

**20. What is the purpose of git config command ?**

We can set the design choices for GIT establishment by utilizing git config order.

**21. How can we see the configuration settings of GIT installation ?**

We can utilize ‘git config – list’ summon to print all the GIT arrangement settings in GIT installation.

**22. How will you write a message with commit command in GIT ?**

We call following command for commit with a message: $/ > git commit –m < message >

**23. What is stored inside a commit object in GIT ?**

GIT confer question contains following data: SHA1 name: A 40 character string to recognize a submit Files: List of documents that speak to the condition of a task at a particular purpose of time Reference: Any reference to parent submit objects.

**24. How many heads can you create in a GIT repository ?**

There can be any number of heads in a store. Of course, there is one head known as HEAD in every archive in GIT.

**25. Why do we create branches in GIT ?**

If we are all the while chipping away at various errands, tasks, deformities or highlights, we require numerous branches. In GIT we can make a different branch for each different reason.

Let say we are dealing with an element, we make a component branch for that. In the middle of we get a deformity to take a shot at then, we make another branch for imperfection and work on it. Once the deformity work is done, we combine that branch and return to chip away at include branch once more.

So taking a shot at numerous errands is the fundamental purpose of utilizing various branches

**26. What are the different kinds of branches that can be created in GIT ?**

**We can make various types of branches for following purposes in GIT:**

**Feature branches:** These are utilized for building up a component.

**Release branches:** These are utilized for discharging code to create.

**Hotfix branches:**These are utilized for discharging a hotfix to generation for an imperfection or crisis settle.

**27. How will you create a new branch in GIT ?**

We use following command to create a new branch in GIT: $/ > git checkout –b < branchname >

**28. How will you add a new feature to the main branch ?**

We do the advancement chip away at a component branch that is made from the ace branch. Once the advancement work is prepared we utilize git consolidate charge to blend it into an ace branch.

**29. What is a pull request in GIT ?**

A draw ask for in GIT is the rundown of changes that have been pushed to GIT store. For the most part, these progressions are pushed in a component branch or hotfix branch. In the wake of pushing these progressions, we make a force ask for that contains the progressions amongst ace and our element branch. This draw asks for is sent to analysts for investigating the code and afterward blending it in to create or discharge branch.

**30. What is merge conflict in GIT ?**

A consolidation struggle in GIT is the consequence of combining two confers. Once in a while, the resolve to be blended and current submit have changes in the same area. In this situation, GIT can’t choose which change is more critical. Because of this GIT reports a union clash. It implies blend isn’t effective. We may need to physically check and resolve the consolidation struggle. [Top 30 GIT Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-30-git-interview-questions-and-answers-pdf)

**31. What is the command you can use to write a commit message ?**

 The order that is utilized to compose a submit message is “git confer – a”. The – an on the order line trains git to confer the new substance of every single followed document that has been adjusted. You can utilize “git add< file >” before git commit –a if new files need to be committed for the first time.

**32. What are the main differences between GIT and SVN ?**

The distinction amongst GIT and SVN is

a) Git is less favored for dealing with to a great degree expansive documents or as often as possible changing paired records while SVN can deal with different activities put away in a similar store.

b) GIT does not bolster ‘confers’ over various branches or labels. Subversion permits the formation of envelopes at any area in the storehouse design.

c)under a label root Subversion enables committers to regard a tag as a branch and to make numerous amendments while Gits are unchangeable.

**33. What language is used in GIT ?**

GIT is quick, and ‘C’ dialect makes this conceivable by decreasing the overhead of runtimes related with higher dialects.

**34. What is the function of ‘GIT PUSH’ in GIT ?**

‘GIT PUSH’ refreshes remote refs alongside related objects.  [Top 50 Maven Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-50-maven-interview-questions-and-answers-pdf)

**35. Why GIT better than Subversion ?**

GIT is an open source rendition control framework; it will enable you to run ‘variants’ of a venture, which demonstrate the progressions that were made to the code additional time likewise it permits you to keep the backtrack if vital and fix those progressions. Various engineers ([pivotal training](https://svrtechnologies.com/pivotal-training)) can check out, and transfer changes and each change would then be able to be credited to a particular designer.

**36. What is “Staging Area” or “Index” in GIT ?**

Before finishing the submits, it can be organized and inspected in a middle of the road zone known as ‘Arranging Area’ or ‘Index’.[Top 40 JUnit Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-40-junit-interview-questions-and-answers-pdf)

**37. What is GIT stash drop ?**

When you are finished with the reserved thing or need to expel it from the rundown, run the git ‘stash drop’ summon. It will evacuate the last included reserve thing as a matter of course, and it can likewise expel a particular thing on the off chance that you incorporate as a contention.

**38. How will you know in GIT if a branch has been already merged into master ?**

**Git branch**—blended records the branches that have been converted into the present branch

**Git branch—** no consolidated records the branches that have not been blended.

**39. What is the function of git clone ?**

The git clone summon makes a duplicate of a current Git archive. To get the duplicate of a focal archive, ‘cloning’ is the most widely recognized route utilized by developers.

**40. What is the function of ‘git config’ ?**

The ‘git config’ summon is a helpful method to set setup choices for your Git establishment. Conduct of a vault, client information, inclinations and so forth can be characterized through this command.

**41. What does commit object contain ?**

An arrangement of records, speaking to the condition of an undertaking for a given purpose of time

Reference to parent confer objects

An SHAI name, a 40 character string that extraordinarily distinguishes the confer protest.

**42. How can you create a repository in Git ?**

: In Git, to make a storehouse, make an index for the undertaking on the off chance that it doesn’t exist, and after that run order “git init”. By running this order .git index will be made in the task registry, the catalog does not should be empty

**43.In git, what is ‘head’ and how many heads can be made in a repository ?**

To a confer question**,** a ‘head’ is basically a reference. In each storehouse, there is a default head alluded as “Ace”. An archive can consist any number of heads.

**44. What is the purpose of branching in GIT ?**

The motivation behind spreading in GIT is that you can make your own branch and hop between those branches. It will enable you to go to your past work keeping your current work intact

**45. What is the common branching pattern in GIT ?**

The basic method for making the branch in GIT is to keep up one as “Fundamental” branch and make another branch to actualize new highlights. This example is especially valuable when there are numerous designers chipping away at a solitary venture. [Top 40 Jira Interview Questions and Answers](https://svrtechnologies.com/devops-interview-questions/top-40-jira-interview-questions-and-answers-pdf)

**46. What is a ‘conflict’ in git ?**

A ‘contention’ emerges when the confer that must be consolidated has some adjustment in one place, and the current submit likewise has a change at a similar place. Git won’t have the capacity to anticipate which change should come first.

**47. How can conflict in git resolved ?**

To determine the contention in git, alter the documents to settle the clashing changes and after that include the settled records by running “git include” after that to confer the repaired combine, run “git submit”. Git recollects that you are amidst a merger, so it sets the guardians of the confer correctly. [Devops Video Training](https://svrtechnologies.com/video-training-courses)

**48. To delete a branch what is the command that is used ?**

Once your improvement branch is converted into the primary branch, you needn’t bother with

Advancement branch. To erase a branch utilize, the summon “git branch–d [head]”

**49. What is another option for merging in git ?**

“Rebasing” is the other option to converging in git.[Devops Training](https://svrtechnologies.com/devops-training)

**50. What is the syntax for “Rebasing” in Git ?**

The syntax used for rebase is “git rebase [new-commit]” [Company](https://devops.com/)

**What Is Github?**

GitHub is industry-standard version control and publishing platform for web developers. GitHub is the place where developers store the code. It is the best place to share code with friends, co-workers, classmates, and total strangers. For understanding GitHub, we should understand about “Git.” “Git” is version control system, which means that whenever the developer creates something and make changes to the code or release new versions, anyone can keep track of all the modifications in a central repository.

**How Does Github Help In Collaborating With Other Developers?**

Whenever any programmer uploads their code to Git, other developers can access and edit it, isolated from each other and with all versions kept intact. Git takes away any confusion with having multiple people working on the same file at the same time. There are no chances of your hard work.

**How To Use Github?**

**The use of GitHub can be explained in the following steps:**

Install git and create a GitHub account.

Create a local git repository.

Add a new file to the repo.

Add a file to the staging environment.

Create a commit.

Create a new branch.

**What Is Github Link?**

GitHub is a web-based Git or version control repository on the Internet with a hosting service. It offers the distributed version control and source code management (SCM) functionality of Git as well as adding its own new features.

**Is Github Open Source Software?**

GitHub is not open-source software because they have a commercial offer based known as "GitHub for Enterprise.” Nevertheless, anyone can apply their ideas just like open-source software on GitHub look alike, known as GitLab which is a ruby application with its source code here.

**What Is A Branch On Github?**

Branch in Git means a lightweight movable pointer to one of these commits. The default branch name in Git is the master. As we initially make commits, we are given a master branch that points to the last commit being made. Every time we commit, it moves forward automatically.

**What Is A Git Pull?**

Git pull is shorthand for Git fetch, followed by Git merge FETCH\_HEAD. More specifically, Git pull runs Git fetch with the given parameters and calls Git merge to merge the retrieved branch heads into the current branch. Should be the name of a remote repository as forwarded to Git-fetch.

**What Is A Git Pull Request?**

Pull requests let us inform others about the changes that have been pushed to a GitHub repository. Once a pull request is sent, interested developers can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

**What Is Github Handle?**

GitHub handle is the username, which is displayed, on the top right corner in GitHub. This is the one we use to login to GitHub when we enter the site and commit to HTTPS and the one that appears in the URLs of our GitHub repositories.

**What Is Repo?**

GitHub users shorten this to “repo.” It can be local to a folder on the computer, or it can be a storage space on GitHub or another online host. We can keep code files, text files, image files; our name it, inside a repository.

**How Can We Create A Branch In Github?**

GitHub has a simple UI for creating branches. Once opened the branch drop-down and it prompts "Find or create a branch.” We need to Type the name of your new branch, then click the "create" button that appears. For retrieving the new branch from GitHub, we can use the standard Git fetch command.

**What Is A Github Fork?**

Fork is a copy of a repository. Forking a repository allows us to freely experiment with the changes without affecting the original project. Most commonly, forks are used to propose changes to someone else's project or to use someone else's project as a starting point of our own idea.

**Can We Change Our Github Username?**

We can change our GitHub account name at any time. For this, we need to click the profile picture > Settings > Account Settings > Change Username. Links to the repositories will redirect to the new URLs, but they should be updated on all other sites because someone who chooses the abandoned username can override the links.

**How Much Space Do We Get On Github?**

We get a space of 1 GB but if it exceeds 1GB, we receive a polite email from GitHub Support requesting to reduce the size of the repository and scale it down. In addition, we place a strict limit of files exceeding 100 MB in size.

**Can We Delete The Github Account?**

We can delete the GitHub user account any time. Before we do so, we should hand over the reins of any organizations we might own. Deleting the user account removes all repositories, forks of private repositories, wikis, issues, pull requests, and pages owned by the account.

**What Is The Difference Between Bit Bucket And Github?**

Bit bucket and GitHub are very similar in terms of features. If the VCS is Mercurial, then Bit bucket is for us. Each has its own features not found in the other, such as GitHub pages for small web-hosting projects, or Bit bucket’s JIRA integration.

**How To Upload Files To Github?**

**Following are the step required for uploading files:**

Navigate to the main page of the GitHub repository.

Under your repository name, click Upload files.

Drag and drop the file or folder you would like to upload to your repository onto the file tree.

At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file.

**What Is A Commit?**

Commit is a record of the files that have been modified, since the last time we made a commit. We make changes to our repo (for example, adding a file or modifying one) and then tell Git to put those files into a commit. Commits make up the core of our project and allow us to go back to the state of a project at any point.

**What Is Staging Environment?**

This is a process through which, we request Git, to put files into a commit? This is where the staging environment or index comes in handy. When we make changes to the repo, Git notices that a file has changed but will not do anything with it (like adding it in a commit). To add a file to a commit, we need to add it to the staging environment. To do this, we can use the Git add command. Once we have used the Git, add the command to add all the files we want to the staging environment, we can then request Git to package them into a commit using the Git commit command.

**Explain Github Workflow?**

**Git provides three key areas that are uniquely designed, to give developers lots of control over workflow:**

A working directory, which contains all the current states of files. Numerous developers can access directory when they are logged in, so collaboration is extremely easy.

Staging area, which indexes everything for the next commit, and any files that have been added or edited since the previous save.

Git repository is a dedicated space where new commits are added. A Git repository maintains all the metadata, the files, and a dedicated database that tracks versions of the project.

With proper importance given on speed, data integrity, and excellent support for distributed, non-linear workflows, GitHub provides excellent standards for remote digital teams. In addition, Git is convenient for working offline or without a VPN (virtual private network), making it easy for the developers to work on the move and stay productive at will. Small & medium enterprises that are involved in building sites rapidly and prototypes for their customers, GitHub is the most efficient, safe and seamless way to get projects reviewed, approved, and signed off or on. GitHub provides the most cost-effective module in any business.

**What Is Git?**

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

**What Is A Repository In Git?**

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**What Is The Command You Can Use To Write A Commit Message?**

The command that is used to write a commit message is “git commit –a”.  The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

**What Is The Difference Between Git And Svn?**

**The difference between GIT and SVN is:**

a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

b) GIT does not support ‘commits’ across multiple branches or tags.  Subversion allows the creation of folders at any location in the repository layout.

c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

**What Are The Advantages Of Using Git?**

a)      Data redundancy and replication  
b)      High availability   
c)       Only one.git directory per repository  
d)      Superior disk utilization and network performance  
e)      Collaboration friendly  
f)       Any sort of projects can use GIT

**What Language Is Used In Git?**

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

**What Is The Function Of ‘git Push’ In Git?**

‘GIT PUSH’ updates remote refs along with associated objects.

**Why Git Better Than Subversion?**

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes.  Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

**What Is “staging Area” Or “index” In Git?**

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**What Is Git Stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

**What Is Git Stash Drop?**

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command.  It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

**How Will You Know In Git If A Branch Has Been Already Merged Into Master?**

Git branch merged lists the branches that have been merged into the current branch

Git branch no merged lists the branches that have not been merged

**What Is The Function Of Git Clone?**

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’  is the most common way used by programmers.

**What Is The Function Of ‘git Config’?**

The ‘git config’ command is a convenient way to set configuration options for your Git installation.  Behaviour of a repository, user info, preferences etc. can be defined through this command.

**What Does Commit Object Contain?**

a) A set of files, representing the state of a project at a given point of time  
b) Reference to parent commit objects  
c) An SHAI name, a 40 character string that uniquely identifies the commit object.

**How Can You Create A Repository In Git?**

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

**What Is ‘head’ In Git And How Many Heads Can Be Created In A Repository?**

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”.  A repository can contain any number of heads.

**What Is The Purpose Of Branching In Git?**

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

**What Is The Common Branching Pattern In Git?**

The common way of creating branch in GIT is to maintain one as “Main“ branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

**How Can You Bring A New Feature In The Main Branch?**

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

**What Is A ‘conflict’ In Git?**

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**How Can Conflict In Git Resolved?**

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge,  run “git commit”.  Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**To Delete A Branch What Is The Command That Is Used?**

Once your development branch is merged into the main branch, you don’t need development branch.  To delete a branch use, the command “git branch –d [head]”.

**What Is Another Option For Merging In Git?**

“Rebasing” is an alternative to merging in git.

**What Is The Syntax For “rebasing” In Git?**

The syntax used for rebase is “git rebase [new-commit] “

**What Is The Difference Between ‘git Remote’ And ‘git Clone’?**

‘git remote add’  just creates an entry in your git config that specifies a name for a particular URL.  While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

**What Is Git Version Control?**

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version.  Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

**Mention Some Of The Best Graphical Git Client For Linux?**

**Some of the best GIT client for LINUX is:**

a)      Git Cola  
b)      Git-g  
c)      Smart git  
d)      Giggle  
e)      Git GUI  
f)       qGit

**What Is Subgit? Why To Use Subgit?**

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration.  Subgit is a solution for a company -wide migration from SVN to Git that is:

a)      It is much better than git-svn  
b)      No requirement to change the infrastructure that is already placed  
c)       Allows to use all git and all sub-version features  
d)      Provides genuine stress –free migration experience.

**What Is The Function Of ‘git Diff ’ In Git?**

‘git diff ’ shows the changes between commits, commit and working tree etc.

**What Is ‘git Status’ Is Used For?**

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

**What Is The Difference Between The ‘git Diff ’and ‘git Status’?**

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

**What Is The Function Of ‘git Checkout’ In Git?**

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

**What Is The Function Of ‘git Rm’?**

To remove the file from the staging area and also off your disk ‘git rm’ is used.

**What Is The Function Of ‘git Stash Apply’?**

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

**What Is The Use Of ‘git Log’?**

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

**What Is ‘git Add’ Is Used For?**

‘git add’ adds file changes in your existing directory to your index.

**What Is The Function Of ‘git Reset’?**

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

**What Is Git Is-tree?**

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

**How Git Instaweb Is Used?**

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

**What Does ‘hooks’ Consist Of In Git?**

This directory consists of Shell scripts which are activated after running the corresponding Git commands.  For example, git will try to execute the post-commit script after you run a commit.

**Explain What Is Commit Message?**

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

**How Can You Fix A Broken Commit?**

To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

**Why Is It Advisable To Create An Additional Commit Rather Than Amending An Existing Commit?**

There are couple of reason

a) The amend operation will destroy the state that was previously saved in a commit.  If it’s just the commit message being changed then that’s not an issue.  But if the contents are being amended then chances of eliminating something important remains more.

b) Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

**What Is ‘bare Repository’ In Git?**

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers ‘Bare Repository’ is used. A bare repository comprises of a version history of your code.

**Name A Few Git Repository Hosting Services:**

Pikacode

Visual Studio Online

GitHub

GitEnterprise

SourceForge.net

Svn Subversion Interview Questions & Answers

Svn Subversion Interview Questions & Answers

**What Exactly Is Svn?**

Open source version control system (VCS) which is in the short referred to as SVN, specifically designed for tracing all the modifications that have taken place to the source code of your design or files. SVN is a source which is utilised to direct all the files and folders and all the recent modifications which have happened in these files. SVN helps us to keep a complete track of the changes done by any person or by yourself onto the files.

**List Out What All Things Should Be Stored In Svn Repository?**

Source Code

Mysql Queries

Database Updates

Project regarding important files

Product Documents

Minutes of Meting and Imp Email

**Difference Between Svn Commit And Svn Update?**

**SVN commit:** Push (upload) the local changes to Repository.

**SVN Update:** Get (download) the Repository changes files to local system.

**What Is Use Of Revert In Svn?**

Revert your local changes.

**It have two types:**

**Local Revert:** It will delete all changes from files which you made after updates and before commit.

**Repo Revert:** Upload the changes to previous Repo.

**List Out All The Command Prompts Used In The Svn?**

SVN has a quite long list of commands prompt out of which few are considered to be most effective and are regularly used in performing different operations in this system.

**Some of the most commonly used common prompts in SVN are:**

Revert

Import

Checkout

Commit

Update

Copy

Beside these command prompts SVN has also the presence of other set of commands which are handy for different set of operations.

**How To Delete File From Svn Repo?**

svn delete filename

**What Is The Use Of Tortoisesvn?**

TortoiseSVN is a Subversion client, implemented as a Microsoft Windows shell extension, that helps programmers manage different versions of the source code for their programs. It is a free software released under the GNU General Public License.

**How To List The Entire Cos By A User In Different Jobs Through A Single Command?**

ct lsco -me

this gives all checked out files by a user

ct lsco -avobs -cview -me

List all files checked out to all views by a user

**Describe About Different Best Practices For Svn?**

**Some of the best practical approaches which are the best fit for SVN if effectively followed include:**

The very first thing to be taken care of while working with the SVN is that make use of your own local space in order to perform any function in the SVN.

Secondly while working on SVN platform it is highly advisable to commit to all the autonomous changes no matter how small the changes may be.

Making use of the comment option will be of a great help for the team to know what are the changes are done and why by just going through the comment section.

The best approach in SVN is to validate all the set of files you are committing, you actually changed and it is always advisable to keep a copy of the update whenever you proceed to commit to the report.

**Tell Us About All The Elements That Should Be Stored In The Svn Repository?**

SVN repository can be used for storing all the existing changes and files which are present in the branches of the existing system.

**Here below is the list of all the things which are stored in the SVN repository:**

MySQL Queries

Project regarding important files

Product Documents

Database Updates

Minutes of Meting and Imp Email

Source Code

**Explain About The Concept Of Synchronizing With The Repository & Tell Us About The Disparity Between Synchronizing And Update?**

The term synchronizing with the repository simply relates to the procedure of updating your local workspace with all the changes which have been committed by the other members of your team.

This is totally different from the concept of Update as Merge is a function that can be performed manually for the merge conflicts by simply using this.

**What Is The Command To Create A New Directory Under Version Control?**

Command to create a new directory under version control includes

Svn mkdir directory

Svn mkdir http://url/directory

**Mention How You Can Import Your Existing Directory Into The New Repository?**

The command you will use to import your existing directory into the new repository you have to write

Svn import/home/my surface/programming

file:///home/mysurface/repo/programing\_repo-m “initial import”.

**State The Procedure Of Creating A Patch Is Svn?**

For the purpose of applying a patch in SVN the first thing you are required to do is to “Create Patch” by stimulating changes and creating the .dff file. Then, this existing .diff file can be implemented to the latest code base by using “Apply Patch”.

**Differentiate Between The Git & Svn Repository?**

**The major differentiating aspects of GIT & SVN concepts are:**

The command “commit” is not supported by GIT across multiple branches or tags & whereas SVN facilitates the creation of new folders t any location of your repository.

GIT is considered to be unvarying where as SVN supports the function of multiple revisions.

In the case where your presented with larger files and frequently varying binary files, the GIT is less preferred & where as SVN is best preferred for handling a number of projects which are stored in the same repository.

**Differentiate The Concepts Of Commit And Update?**

In SVN repository the word update relates to the process of updating the local workspace in relation to all the changes which are done within the repository by the members of the team.

While the word commit relates to the simple process of making changes within the local to repository. In much easier terminology it can stated as uploading a file into the repository.

**What Can You Say Regarding The G & R Result Codes In Svn?**

**The SVN result codes are:**

**G Code:** G Code indicates all the changes which are that are consolidated into working copy of the report automatically.

**R Code:** R Code specifies that particular context or item which is replaced in the present working directory copy. It indicates that a particular file has underwent some changes are is arranged for deleting and some other new file is scheduled to get added in its area.

**State The Process Involved To Revert To A Previous Version?**

For the purpose of retrieving the previously existing version, the main command on which you are needed to focus is the “revert” command. The action performed by the revert command is that it simply erase all the local edits. But the main command which is needed to be used for this operation is the “merge” command.

 This can be explained with a simple example.  For instance, the present version of your file [xyz.txt] is version 201 and the version which you currently require is 301.

**In such an instance the command code should be executed in this manner**

**svn merge –r 201:301 xyz.txt**

**svn commit –m “Reverted to revision 301” xyz.txt**

**What Do You Mean By "synchronizing With Repository" ? How Is It Different From "update”?**

Synchronizing with Repository is the process of updating the local workspace with the changes committed by others. This is different from Update as we can do Merge Manually for the Merge conflicts using this.

**What Is The Best Practice For Svn?**

**Best practices for SVN:**

Update and Test before commit

Work from your own local workspace

Commit small autonomous changes

Validate the files you are committing, you actually changed

Keep in touch with repository

Watch for conflicts

Always group your check-in logically

Use comment

**What Is “branch” , “tag” And “trunk” In Svn ?**

Trunk is the main body of development, originating from the start of the project till end.

Branch is a copy of code derived from a certain point in the trunk that is used for applying major changes to the code while preserving the integrity of the code in the trunk.

Trunk is a point in time on the trunk or a branch that you wish to preserve. This is like base lining the code after a major release.

# Base ClearCase CheatSheet

Use this "cheat sheet" as a quick reference and practical aid to efficent use of IBM Rational® ClearCase®. Tips, reminders, and expert advice from one of Rational's in-house ClearCase pros answer the most common user needs.

## Views

### How many Views do I need?

Generally, one per discrete task is a good idea, though this is not a requirement. Views should be regarded as "cheap" entities that are easily created and destroyed. Create one for each task you need to do and when that task is done, delete the view.

### Dynamic versus Snapshot Views

ClearCase and ClearCase LT offer snapshot views as workspaces. Snapshot views work by copying versions of elements from VOBs to your computer. To see the latest versions of elements, you need to update your snapshot view periodically. An update operation copies the latest versions of elements from the VOB to your view.

ClearCase (but not ClearCase LT) also offers dynamic views. A dynamic view uses the Multiversion File System (MVFS) to provide immediate, transparent access to data stored in VOBs. When you work in a dynamic view, you do not need to copy data fromVOBs to your view; you always see the latest versions of elements. Dynamic views also provide advanced functionality such as build auditing and binary sharing.

### How to create a Dynamic View

|  |  |
| --- | --- |
| 1 | cleartool mkview -tag atag  storage-location |

Where atag is the view-tag (the "name" of the view which should be a name that you can easily remember) and storage-location is the location of a place on a disk where the view can store information.

Alternatively, there is a simplified command:

|  |  |
| --- | --- |
| 1 | cleartool mkview -tag atag -stgloc -auto |

In this example, you don't need to specify a storage location, but rather this command enables you to use a storage location that has been set up by your ClearCase admin.

### How to delete a Dynamic View

|  |  |
| --- | --- |
| 1 | cleartool rmview -tag atag |

Where atag is the view-tag of the view you want to remove.

### How to create a Snapshot View

|  |  |
| --- | --- |
| 1 | cleartool mkview -snapshot -tag atag -stgloc -auto apathname |

Where apathname is the path to which you want your source code (work area) to reside. The -tag atag is optional - if you omit it, ClearCase will make one up based on your username and the workspace you specify in apathname.

### How to delete a Snapshot View

|  |  |
| --- | --- |
| 1 | cleartool rmview apathname |

Where apathname is the path to where your source code (work area) resides.

## Modifying files

### Checkout

To modify an element you need to check it out. Do this with the following command:

|  |  |
| --- | --- |
| 1 | cleartool co anelement |

where anelement is the name of the file or directory you want to check out.

### Checkin

Once you are happy with the changes you have made to a checked-out element, you can check it in with the following command:

|  |  |
| --- | --- |
| 1 | cleartool ci anelement |

where anelement is the name of the file or directory you want to check in. Once you check-in an element, the changes you made to that element are usable by everyone on your project.

### Uncheckout

If you want to cancel a checkout, you can do so with the following command:

|  |  |
| --- | --- |
| 1 | cleartool unco anelement |

where anelement is the name of the file or directory you want to check in.

## Creating new files

### Creation

To create a new file, you can use whatever editor or command you would normally use to create the file. The new file will be a view-private entity until you take the next step to put it under ClearCase control.

### Putting a file under version control

To put a file under ClearCase control you must first check-out the directory that will be containing the file, issue the command to tell ClearCase to put the file under version control, and then check-in the directory containing the new file. Here are commands to perform these steps:

|  |  |
| --- | --- |
| 1  2  3 | cleartool co thedirectory  cleartool mkelem thefile  cleartool ci thedirectory |

After this sequence is performed, the file thefile is still checked-out and can be modified further before you check it in.

## Deleting elements

To remove an element you should use the cleartool rmname command. This command removes the name of the specified element from the directory in which it is contained. Similar to placing an element under ClearCase control, you must first check-out the directory containing the element to be removed, do the cleartool rmname of the element, and then check-in the new version of the directory. Here is an example:

|  |  |
| --- | --- |
| 1  2  3 | cleartool co thedirectory  cleartool rmname anelement  cleartool ci thedirectory |

After this sequence, anelement is no longer listed in subsequent versions of thedirectory. It is, however, still listed in previous versions of the directory. If, at a later date you want to re-add anelement to a new version of thedirectory (or any other directory), you can do so easily.

## Renaming elements

Renaming an element is similar to deleting it. First you check-out the old directory containing the element, and you also check-out the new directory to which the element is to be moved. Then you use the cleartool mv command to perform the move operation. Be sure to check-in the old and new directories when you are done.

|  |  |
| --- | --- |
| 1  2  3  4  5 | cleartool co .  cleartool co newdirectory  cleartool mv anelement newdirectory  cleartool ci .  cleartool ci newdirectory |

Renaming an element is also accomplished using the cleartool mv command except the old directory and the new directory are the same.

|  |  |
| --- | --- |
| 1  2  3 | cleartool co .  cleartool mv anelement anewname  cleartool ci . |

In subsequent versions of the directory, the name anelement will not appear, having been replaced by anewname but the name anelement will still appear in prior versions of the directory. Both the names anewname and anelement will still refer to the same element; only the name associated with the element in the directory will have changed.

## Getting information

### More about Views

What View am I in?

At any given time you can tell what view you are in by issuing the following command:

|  |  |
| --- | --- |
| 1 | cleartool pwv |

pwv means "print working view" in the tradition of the UNIX command pwd for "print working directory"

### What is my "config-spec"?

Recall that a configuration specification (config-spec) is the set of rules that tells ClearCase what configuration of elements you want to work with. You can see that set of rules by issuing the following command:

|  |  |
| --- | --- |
| 1 | cleartool  catcs |

### How do I change my config-spec?

You can edit your config-spec by issuing the following command:

|  |  |
| --- | --- |
| 1 | cleartool edcs |

Your favorite editor , as specified by the environment variable WINEDITOR (first choice), VISUAL (second choice), or EDITOR (third choice) will be invoked for the editing operation. If none of these environment variables is set then vi will be invoked for the editing operation on UNIX systems and Notepad will be invoked for the editing operation on Windows systems.

### About files/directories

### What is checked-out and to whom?

If you want to see who has what checked-out, you can use the cleartool lsco command. This command has a number of possible arguments to use depending on the specific information you want. Here are some common ones:

|  |  |
| --- | --- |
| 1 | cleartool lsco |

Lists all checkouts of elements in the current directory.

|  |  |
| --- | --- |
| 1 | cleartool lsco anelement |

List checkout information about the element anelement.

|  |  |
| --- | --- |
| 1 | cleartool lsco -recurse |

Lists all checkouts of elements in the current directory and below.

|  |  |
| --- | --- |
| 1 | cleartool lsco -cview |

Lists checkouts of elements in the current directory that are checked-out to the current view.

|  |  |
| --- | --- |
| 1 | cleartool lsco -cview -recurse |

Lists checkouts of elements in the current directory and below that are checked-out to the current view.

|  |  |
| --- | --- |
| 1 | cleartool lsco -me |

Lists checkouts of elements in the current directory that the current user has checked-out.

|  |  |
| --- | --- |
| 1 | cleartool lsco -user fred |

Lists checkouts of elements in the current directory that the user with login id "fred" has checked-out.

Please refer to the ClearCase Reference Manual for more detailed information about the cleartool lsco command.

### What is the change history of this file/directory?

If you want to see the change history associated with a particular element, you can use the cleartool lsh command.

|  |  |
| --- | --- |
| 1 | cleartool lsh anelement |

Where anelement is the name of an element under ClearCase control

### What versions of what files/directories am I looking at?

To see exactly what versions of elements are being projected to you by your view and why, you can use the cleartool ls command.

|  |  |
| --- | --- |
| 1 | cleartool ls |

### What are the differences between this version of a file/directory and a different version?

To view the differences between the versions of an element visible in your view and another version of the element you can use the cleartool diff command.

|  |  |
| --- | --- |
| 1 | cleartool diff -g -pred anelement |

Graphically presents the differences between the version of anelement selected by your view and it's immediate predecessor version.

|  |  |
| --- | --- |
| 1 | cleartool diff -g anelement anelement@@/main/5 |

Graphically presents the differences between the version of anelement selected by your view and version 5 on the main branch of anelement.

You can also use this command without the -g option to present the difference information to you textually rather than graphically.

See the cleartool diff entry in the ClearCase Reference Manual for more detailed information on these commands.

### What does the version tree look like for this file/directory?

To view a graphical representation of the version tree associated with an element you can use the xlsvtree command.

|  |  |
| --- | --- |
| 1 | cleartool lsvtree -g anelement |

You can view a textual representation of the version tree associated with an element with the cleartool lsvtree command.

|  |  |
| --- | --- |
| 1 | cleartool lsvtree anelement |

### I want more information about this element?

You can get more detailed information about a particular element with the cleartool describe command.

|  |  |
| --- | --- |
| 1 | cleartool describe anelement |

Lists information about anelement such as the version selected by your view, who created it and when, the comment associated with the version, the element type and the predecessor version.