Electronic Signature Information

Name	DOC1203771
Revision	4
Type	Controlled Document
Title	HC Perforce-Tower v2014.1 User Work Instructions
Originator	502150073_ravindar__yarramraju
Release Date	12/09/2014 05:45:47 PM
Obsolete Date	

Name	Reason For Change	File Size (Bytes)
HC_Perforce-Tower 2014 1_UWI_DOC1203771_User Work Instructions_Rev4.docx	HC Perforce - Tower v2014.1 - User Work Instructions	331710

Route	Signer	Function	Status	Comments	Completion Date
R-6058399	100011791_peter__van bloemendal		Review	Reviewed	9 Dec 2014 09:54:42 GMT
R-6058399	100026852_gustavo__hernandez		Review	the doc looks OK.	8 Dec 2014 15:50:16 GMT
R-6058399	212069664_mohammed__azmathullah		Approve	Approved	9 Dec 2014 17:45:37 GMT
R-6058399	212413056_balázs__nagy		Review	Approved	8 Dec 2014 15:29:58 GMT
R-6058399	502150073_ravindar__yarramraju		Review	Reviewed	9 Dec 2014 13:50:56 GMT
R-6058399	502314189_pavankumar__nagubandi		Review	approved	9 Dec 2014 14:19:24 GMT

Periodic Review

There are no signatures or routes related to this business object.

Obsolescence Approval

There are no signatures or routes related to this business object.

* Printed versions are For Reference Only *
+ Indicates a task was reassigned from an original assignee

GE Healthcare
Science and Technology Organization

User Work Instructions for HC Perforce – Tower v2014.1

Document ID: DOC1203771

Author:	Pavan kumar Nagubandi, Technical Representative, STO-Software
Reviewers:	Ravindar Yarramraju, Validation Manager, HCIT-NPI
	Balazs Nagy, Technical Representative, STO-Software
	Peter Van Bloemendal, Business System Owner, STO-Software-Platform Eng Excellence-Process Eng-ET&P
	Gustavo Hernandez, System User, STO-HCIT-Specialty Sol-Engineering-Imaging-IB (EU)
Approver:	Mohammed Azmathullah, QA Lead Engineer, Quality-HCIT QARA-Clinical Business

Note: Before using this document, make sure it is the latest revision. Access the MyWorkshop system to verify the current revision. If you do not have access to, or are unfamiliar with, MyWorkshop system, you should consult your quality representative.

Any printout of this document is considered an uncontrolled copy.

This document is proprietary to GE Healthcare.

Document Revision History

A record of changes made to this document by the project team.

Revision	*Date*	Author	Change Reference	Reason for Change
1	05-JUL-2013	Ankush Kala	All Pages	Initial Release
2	06-AUG-2014	Suman Dachepalli	Cover page Section 4.3.1 Section 4.3.4 Section 5 Section 7	Updated reviewers and approvers Updated Screenshots Updated workspace names updated instructions on how to connect to the server and login and version numbers updated perforce version numbers Updated the document as per 2014 .1 upgrade - FCCC 366867
3	17-OCT-2014	Peter van Bloemendal Suman Dachepalli	Cover Page Section 1 Section 4.3 Section 4.5 Section 4.4	Updated reviewers and approvers Updated definitions Renumbered section Perforce Client installation and configuration Added section File Review Process using Collaborator - FCCC#379033 Added section Using Git Client with Perforce - FCCC#380234
4	08-DEC-2014	Pavan kumar Nagubandi	Cover Page Section 2 Section 4.3 Section 4.3.5 Section 4.3.6 Section 4.4.1 Section 5	Updated System User reviewers Updated server details Commit server and Edge server Updated Perforce client details Added a new section for Workspace migration - FCCC#385684 Added a new section for Shelve promotion - FCCC#385684 Updated Git client details Updated proxy server details Updated shelve promotion FAQ

Contents

1 DEFINITIONS

The following definitions apply to specialized terms used in this document.

Definitions/Acronyms/Abbreviations	Meaning
Branch	A set of related files created by copying files, as opposed to adding files. A group of related files is often referred to as a codeline.
Branch View	A specification of the branching relationship between two codelines in the depot. Each branch view has a unique name and defines how files are mapped from the originating codeline to the target codeline.
Changelist	The changelist is the basic unit of work in Perforce. The basic file editing operations common to all SCM systems (such as editing files, adding files, deleting files, backing out changes and checking in files) are performed in changelists.
P4V (Perforce Visual Client)	Perforce's graphical user interface used to: Connects to the Perforce server to move files between the Perforce depots and the user's workspace.
Client Workspace View	A set of mappings that specifies the correspondence between file locations in the depot and the client workspace.
Collaborator	A product from vendor Smartbear used for creating review events of files including collaboration and issue tracking within the review.
Depot	A repository of files under Perforce control. It contains all existing versions of all files ever submitted to the server. There can be multiple depots on a single server. Files in a depot are referred to as depot files or versioned files.
Depot Pane	In the P4V Client the left Tree pane is the Depot pane which displays all files in the Depot
Git	A distributed version control system and source code management system
GUI	Graphical User Interface
Integrate	To create new files from existing files, preserving their ancestry (branching), or to propagate changes from one set of files to another (merging).
Merge	The process of combining the contents of two conflicting file revisions into a single file, typically using a merge tool like P4Merge.
OS	Operating System, software that manages hardware and software on computer
Perforce	Perforce Software Version Control system
Target File	The file that receives the changes from the donor file when you are integrating changes between a branched codeline and the original codeline.
Trigger	Perforce triggers are user-written programs or scripts that are called by a Perforce server whenever certain operations (such

Definitions/Acronyms/Abbreviations	Meaning
	as change list submissions, changes to forms, attempts by users to log in or change passwords) are performed. If the script returns a value of 0, the operation continues; if the script returns any other value, the operation fails.
URL	Uniform Resource Locator or web address
View	A description of the relationship between two sets of files. See client workspace view, label view, branch view.
Workspace	Folders or directories on the client computer where you work on revisions of files that are managed by Perforce.
Workspace Pane	In the P4V Client the right Tree pane is the Workspace pane which shows all files on your computer, including the ones not in the Depot.
Digital Assets	A digital asset can be any form of data stored on a computer including source code, text, graphics, sound, and video files.
P4D (Perforce Server)	P4D, the Perforce Server, is the engine that powers Perforce's shared versioning service and manages the shared file repository, or depot. P4D handles user requests and tracks development activity in its built-in database
P4P (Perforce Proxy)	P4P, the Perforce Proxy, is self-maintaining and caches versioned files for reuse on any local network with remote access to the Perforce shared versioning service. With P4P, remote users experience significantly quicker response times when downloading versions of files that have already been requested by other remote users.
P4	P4, the Perforce Command-Line Client, is a small, self-contained client program that offers access to all Perforce features.
P4Web	P4Web, the Perforce Web Client, provides convenient access to versioned files through popular web browsers, including Internet Explorer, Firefox, Safari, and Chrome.
FAQ’s	Frequently Asked Questions.

2 SERVER DETAILS

Commit Server's Location:
- Tower Avenue Data Center

Production commit server:
- prd.perforce.health.ge.com:1666
- ssl:prd.perforce.health.ge.com:1668

Edge server Location:
- Bangalore, India
- Buc, France

Edge servers:

- ssl:blr.perforce.health.ge.com:1668	Bangalore, India edge server
- ssl:buc.perforce.health.ge.com:1668	Buc, France edge server

Staging:
- stg.perforce.health.ge.com:1666
- ssl:stg.perforce.health.ge.com:1666

Development:
- dev.perforce.health.ge.com

3 USER ROLES

There are three main roles in our Perforce environment:

- User – Person who utilizes the Perforce application for the purpose of versioning and storing digital assets.
- Admin - A user with added permissions that are higher than a normal user allowing them to perform depot specific admin tasks and functions.
- Superuser – A user with elevated permissions, capable of doing server related operation like starting/stopping Perforce, create/delete/modify users, depots, triggers etc.

4 PERFORCE SETUP

4.1 Perforce System Architecture

Perforce is an enterprise version management system in which users connect to a shared file repository. Perforce applications are used to transfer files between the file repository and individual users' workstations. HC Perforce system architecture is detailed in Technical Architecture Description (TAD) [1].

4.2 System Usage

The Perforce server manages the master file repository, or depot. There can be more than one depot per server. The depots contain every revision of every file under Perforce control. Perforce organizes files in depots into directory trees, like a large hard drive. Files in a depot are referred to as depot files or versioned files. The server maintains a database (based on Berkeley DB) to track change logs, user permissions, and which users have which files checked out at any time. The information stored in this database is referred to as metadata.

Perforce servers use native operating system capabilities to manage the database and the versioned files, and require no dedicated file-systems or volumes.

The Perforce system is used by Software Engineers for software development purpose. Developers can use any of the standard client interfaces detailed above to connect to the central Perforce Server (P4D). For geographically distributed teams P4Proxy servers will be commissioned to enable fast access to the central server in Tower Avenue. P4Web will only support read-only viewer mode that allows read-only access to versioned files and their histories to users.

Perforce is highly scalable and can support more than 7500 users on the Tower Avenue system. Server can use an internal password DB or LDAP/SSO for password authentication, GEHC uses LDAP authentication. Perforce system provides extensive permission and access control system which can be referred in the MyWorkshop document [5].

Perforce Edge server:
An edge server contains a replicated copy of the commit server data and a unique, local copy of some workspace and work-in-progress information. It can process read-only operations and operations like **p4 edit** that only write to the local data. In working terms, it contains local workspace data and can handle more operations with no reliance on the commit server.

An edge server can handle most routine operations locally, a distributed Perforce service offloads a significant amount of processing work from the commit server and reduces data transmission between commit and edge servers.

From a user's perspective, most typical operations until the point of submit are handled by an edge server.

4.3 Perforce Client Application Installation and Configuration:

To access the Perforce server and depot files, you must install and configure a Perforce client on your local machine. There are a variety of client applications and application plug-ins that you can choose from based on your needs and the applications that you work with on a regular basis.

You can download the latest Perforce Client Applications from the local release area:
http://prd.perforce.health.ge.com:443/Admin/main/p4.clients/

In order to obtain and use the Perforce Visual Client on your local machine you first must submit a request on EAST Site (see Section USER SUPPORT) so a user account can be created for you. Perforce is licensed by the user and not by the machine. You will be assigned a license if you need to submit data to projects held in Perforce.

Generally your perforce username will be the same as your GE e-mail prefix similar to "firstname.lastname" although there are some exceptions. If you are unsure of your Perforce account username or status of your account please submit a support ticket as documented in section 6 User Support.

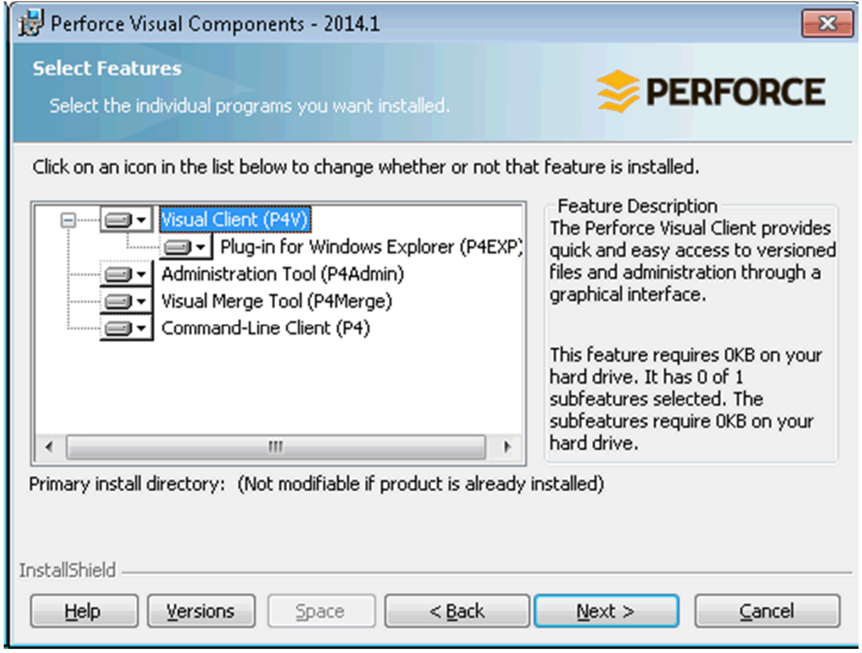
Your Perforce password will be your normal SSO ID password.

The Perforce Administrator will send you the link to the P4V client install package. Typically these URL's are as follows corresponding to the Operating System: - :
http://prd.perforce.health.ge.com:443/Admin/main/p4.clients/

Clicking on the link will list the available binaries w.r.t operation systems, download the p4vinst.exe package. Click "Save" to save the install to your desktop. Click "Run" to start the install and follow the instructions.

4.3.1 Installing and Configuring Perforce Visual Client (P4V):

Download system compatible P4V client from Perforce Website or local release area. The following screen shots illustrate the configuration choices that were presented when running the P4V version 2014.1 installer on Windows:



Perforce Visual Components - 2014.1
Select Features
Select the individual programs you want installed.
PERFORCE

Click on an icon in the list below to change whether or not that feature is installed.

Visual Client (P4V)
Plug-in for Windows Explorer (P4EXP)
Administration Tool (P4Admin)
Visual Merge Tool (P4Merge)
Command-Line Client (P4)

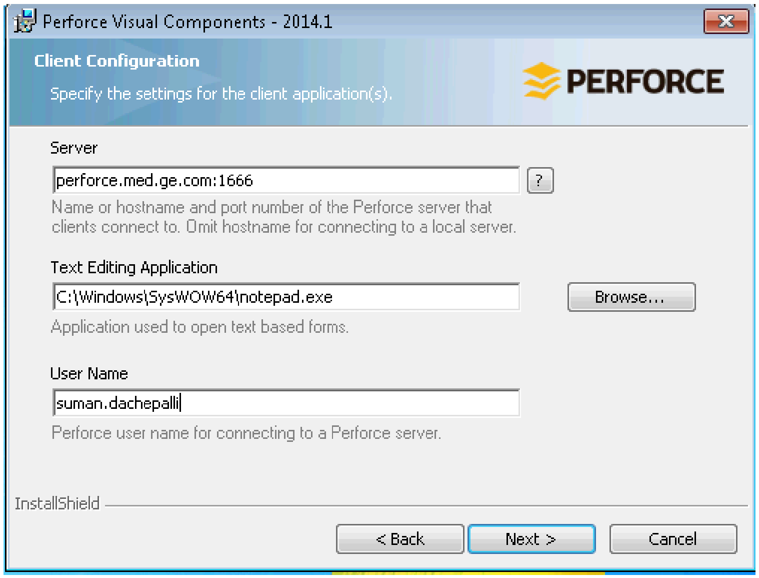
Feature Description

The Perforce Visual Client provides quick and easy access to versioned files and administration through a graphical interface.

This feature requires 0KB on your hard drive. It has 0 of 1 subfeatures selected. The subfeatures require 0KB on your hard drive.

Primary install directory: (Not modifiable if product is already installed)

Figure 1: P4V Installer Feature Selection



Perforce Visual Components - 2014.1
Client Configuration
Specify the settings for the client application(s).
PERFORCE

Server: perforce.med.ge.com:1666

Name or hostname and port number of the Perforce server that clients connect to. Omit hostname for connecting to a local server.

Text Editing Application: C:\Windows\SysWOW64\notepad.exe

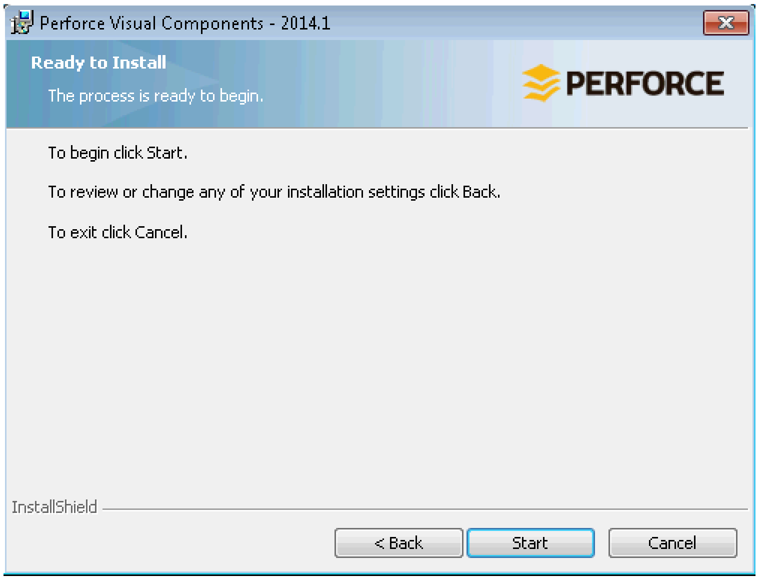
Application used to open text based forms.

User Name: suman.dachepalli

Perforce user name for connecting to a Perforce server.

?
Browse...
InstallShield
< Back
Next >
Cancel

Figure 2: P4V Installer Client Configuration



Ready to Install

The process is ready to begin.

To begin click Start.

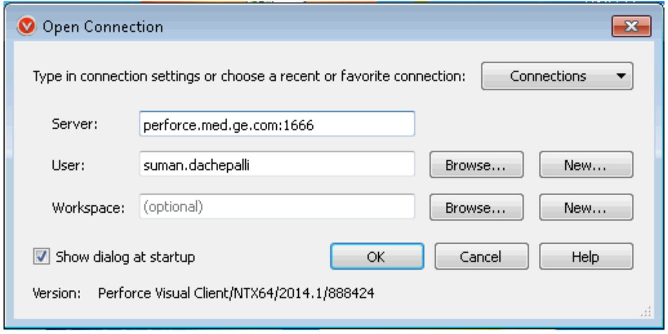
To review or change any of your installation settings click Back.

To exit click Cancel.

Figure 3: P4V Installer Ready Screen

4.3.2 Launch P4V

The connection dialog should look something like the image shown in Figure 4: Open Connection Dialog.



Type in connection settings or choose a recent or favorite connection:

Server: perforce.med.ge.com:1666

User: suman.dachepalli

Workspace: (optional)

☑ Show dialog at startup

Version: Perforce Visual Client/NTX64/2014.1/888424

Figure 4: Open Connection Dialog

- Once you connect to the server, look at the Depot pane inside of P4V and verify that you can see the Depot files.
- Switch to a workspace, Select **Connection > Switch to Workspace...**, select the available workspace and click OK.

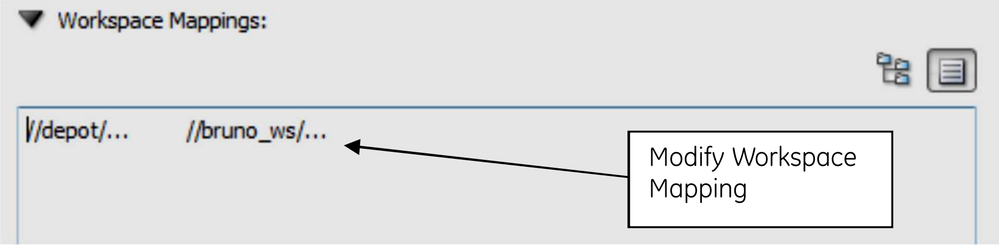
4.3.3 Configuring the Client Workspace

Perforce client programs manage files on your local hard drive in an area called your client workspace. You can have more than one client workspace on your computer. The top-level directory of any client workspace is the client workspace root.

To get files from the Perforce depot into a workspace on your local machine, you must configure a client view that maps files from the depot on the Perforce server to corresponding areas of a workspace on you r local machine. These mappings constitute your client workspace view. The client view for a given workspace identifies which files from the depot on the Perforce server get copied to a workspace on your local machine.

Client workspace views consist of one or more mappings. Each mapping has two elements, a "Depot view", that identifies a set of files in the depot, and a "workspace view", that defines the area in a workspace where the Perforce client places copies of the respective depot files.

To modify Workspace view, Select **Connection > Edit Current Workspace…**



//depot/… //bruno_ws/…

Modify Workspace Mapping

4.3.4 Creating a new Workspace

You can have multiple workspaces on your local machine. As a best practice, Perforce recommends that you create a separate workspace for each unique project that you work on to reduce the clutter in your workspace and reduce the overhead required to keep your workspace files synchronized with the depot.

To create a new workspace named **Task_1**

1. To open the P4V new workspace dialog, from the P4V menu select **Connection > New Workspace…**
2. Configure the dialog as follows:
   - Workspace name: **Task_1**
   - Workspace root: **leave as default**
   - Workspace Mappings:
     //depot/Talkhouse/… //Task_1/Talkhouse/…
   - Switch to new workspace immediately: **Checked**
   - Automatically get all revisions: Consider having this setting as **Unchecked**
   - It is also recommended to check the **"Compress"** Option

After reviewing the dialog configuration, click the **OK** button at the bottom of the dialog to create the **Task_1** workspace.

4.3.5 Migrating a workspace from a commit server or remote edge server to the local edge server

The process for moving a workspace from a commit server to an edge server is as follows:

Scenario: You have a workspace on a commit or remote edge server that you want to move to the local edge server.

1. Have all the workspace owners either submit or revert all of their current work and ensure that all shelved files are deleted.

2. **p4 unload -c <workspace_name>**
Execute this command against the Perforce service where the workspace is being migrated *from*. In this case, this would be the commit or remote edge server.

3. **p4 reload -c <workspace_name> -p <SERVER_URL>:<port>**
Execute this command against the local edge server, where the workspace is being migrated *to*. **<SERVER_URL>**:<**port>** refers to the commit or remote edge server the workspace is being migrated *from*.

4.3.6 Promoting shelved Changelists

Perforce allows changelists shelved on an edge server, which would normally be inaccessible from Commit server and other edge servers, to be *promoted* to the commit server.

For example, given two servers, edge and commit servers, the process works as follows:

1. Shelve and promote a changelist from edge.

p4 shelve -p -c 89

2. The shelved changelist is now available to commit server.

p4 describe -S 89

3. Promotion is only required once.

Subsequent **p4 shelve** commands automatically update the shelved changelist on the commit server, using server lock protection. For example, make changes on **edge** and refresh the shelved changelist:

p4 shelve -r -c 89

The updates can now be seen on **commit**:

p4 describe -S 89

Note: There is no mechanism to *unpromote* a shelved changelist; instead, delete the shelved files from the changelist.

4.4 Using Git Client With Perforce

In addition to traditional perforce GUI and command line client, support has been added to use Git client in conjunction with Git Fusion service to connect to Perforce depot. Main reasons for using Git as a client is that Git supports rapid branching and merging, and includes specific tools for visualizing and navigating a non-linear development history. Branches in Git are very lightweight: A branch in Git is only a reference to a single commit. With its parental commits, the full branch structure can be constructed

4.4.1 Git Client Installation

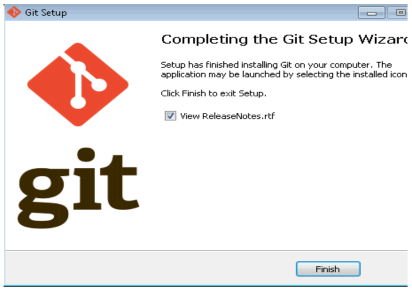
1. Download the Git client from below URL depending on your OS selection

   http://prd.perforce.health.ge.com:443/Admin/main/git.clients/

2. Double click on the installer to start the installation

   Specify the installation directory and the settings in the installer as per your preferences

Select the options from above screens depending on developer's opinion and it triggers the installation



Completing the Git Setup Wizarc

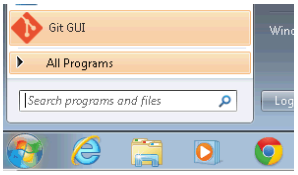
Setup has finished installing Git on your computer. The application may be launched by selecting the installed icon

Click Finish to exit Setup.

☑ View ReleaseNotes.rtf

Clicking on Finish completes the installation

3. To Invoke Git GUI, go to start button and locate "Git GUI". Alternatively Git can be accessed using git bash or command line.



4.4.2 Getting access Git Fusion

An ssh public-private key authentication is required to link Git account to existing Perforce User account. In order to generate such key-pair ssh-keygen must be used and is included in Git client installation.

1. Open command line Prompt on your system and navigate to Git installation folder

Ex: "C:\Program Files (x86)\Git\bin"

2. Execute "ssh-keygen" and generate an SSH Key pair
3. Once the key is generated, raise an SC request[refer Section 6] and attach the public key for gaining Git fusion access

4.4.3 Connecting to Git Fusion

Before users are able to use Git client with perforce server, their permissions need to be added in such a way that they are able to authenticate using their SSH keypair. In addition to this, the user will need to request perforce Admins to map specific parts of depot as a Git repository.

1. Make sure the email in Git configuration is of the format "SSO@ge.com" , you can change the email as follows

```
git config --global user.email sso@ge.com
```

2. To get the latest version of code from perforce repository, execute

```
git clone git@<host name>:<repo name>
```

4.5 Integration - File Review Process using Collaborator

Perforce supports integration with Collaborator for file review. Collaborator allows integration with Perforce Command Line and GUI clients in order to create new review with changelist or attach change list to existing review.

The workflow below describes a common review process

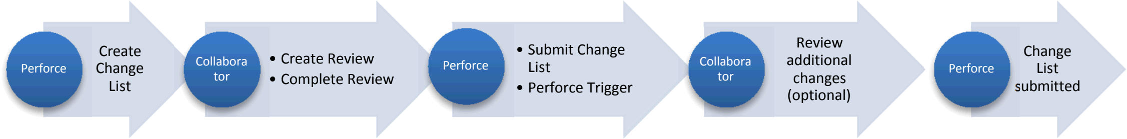


Figure 5: Review process

In addition Perforce can be configured to prevent code to be submitted before review completion using the following Collaborator Triggers for Perforce:

- **ccollab admin trigger ensure-review-started** - Changelist cannot be submitted until review of this changelist exists
- **ccollab admin trigger ensure-reviewed** - Changelist cannot be submitted until review of this changelist has been completed
- **ccollab admin trigger ensure-content-reviewed** - Changelist cannot be submitted until review completed & file list/content is unchanged
- **ccollab admin trigger update-changelist** - Automatically update Perforce changelist with information about the review of that changelist

Based on product team requirements the above trigger customization can be configured.. .

In order to utilize the above-mentioned triggers for creating a content review, installation of Collaborator Client is required.

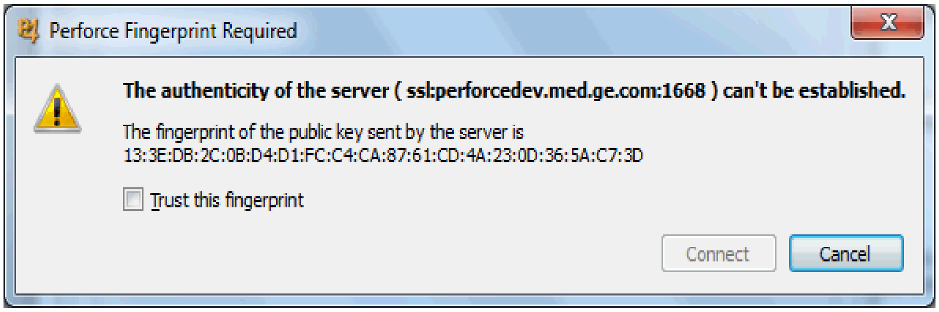
Please refer to section 6 to initiate Support Process for customized trigger configuration and Collaborator client installation.

Note: Change List will not be submitted before review completion therefore it is highly recommended to shelve a changelist to ensure data is not lost during the review cycle.

5 PERFORCE OPERATIONS/FAQ'S

Operation	Instructions
How can I request access to Perforce or request Perforce application support?	See USER SUPPORT section.
Do I need a Perforce Account if I just need to review files (read-only access)?	No, P4Web interface provides reads-only access to files stored in depots. If you need to add/modify files in perforce then you will need a Perforce Account. P4Web Interface: http://<Perforce Server FQDN>/
How to connect to the server and login	1. CLI method Windows CLI 2. `p4 set P4PORT=<Servername :Port no>` 3. `p4 set P4USER=<Username>` 4. `p4 set P4CLIENT=<workspace name>` 5. If the server is SSL enabled, run "`p4 trust`" command when connecting for the first time 6. p4 login Linux CLI 7. `export P4PORT =<Server name:Name>` 8. `export P4USER=<User Name>` 9. `export P4CLIENT=<Workspace Name>` 10. If the server is SSL enabled, run "`p4 trust`" command when connecting for the first time 11. p4 login GUI Method 1. Launch P4V Enter the following info in the connection dialog: 2. Server: **<Perforce Server FQDN:PORT# or Proxy Server FQDN:PORT#>** Example: prd.perforce.health.ge.com:1666 3. User: **Enter your Perforce username here (typically firstname.lastname)** Note: Drop the "**@ge.com**" from the email address as the @ char is an invalid char for Perforce.

4. If you are connecting to an SSL enabled server for the first time, verify the server fingerprint against the one provided to you by the admin and then accept the prompt to confirm and add it to your trusted list.



The authenticity of the server (ssl:perforcedev.med.ge.com:1668) can't be established.

The fingerprint of the public key sent by the server is
13:3E:DB:2C:0B:D4:D1:FC:C4:CA:87:61:CD:4A:23:0D:36:5A:C7:3D

□ Trust this fingerprint

5. Password: **Enter your SSO password here.**

Behind the scenes an authentication script matches your email address up to your numeric SSO user ID. Perforce does not allow all numeric chars for usernames as it conflicts with their changelist numbering scheme.

For geographically distributed teams it's wise to use the closest/local proxy server for faster operations, following proxy servers are available:

p4p-east.engops.health.ge.com:1666
Location: Tower Ave. datacenter, Milwaukee, WI

p4p-freiburg.engops.health.ge.com:1666
Location: Freiburg, DE

p4p-parkridge.engops.health.ge.com:1666
Location: Park Ridge, NJ

p4p-beijing.engops.health.ge.com:1666
Location: Beijing, China

p4p-hillsboro.engops.health.ge.com:1666
Location: Hillsboro, OR

p4p-plano.engops.health.ge.com:1666
Location: Plano, TX

p4p-dornstadt.engops.health.ge.com:1666
Location: Dornstadt, Germany

p4p-burlington.engops.health.ge.com:1666
Location: Burlington, VT

	p4p-chengdu.engops.health.ge.com:1666 Location: Chengdu, China p4p-hino.engops.health.ge.com:1666 Location: Hino, Japan p4p-sanramon.engops.health.ge.com:1666 Location: SanRamon, US ssl:p4p-uppsala.engops.health.ge.com:1668 Location: Uppsala, Sweden ssl:p4p-kuopio.engops.health.ge.com:1668 Location: Kuopio, Finland ssl:p4p-budapest.engops.health.ge.com:1668 Location: Budapest, Hungary
How to get latest version of a file	• CLI method 1. `p4 sync //depot/project/folder/filename` • GUI Method 1. Right click on the file, select **Get Latest Revision**
How to get a specific revision of a file	• CLI method 1. `p4 sync //depot/project/folder/filename#2` *Gets rev #2 of the file.* • GUI Method 1. Right Click on the File, select “**Get Revision**” 2. In the "Get Revision" dialog box, click on the radio button for "**Specify revision using:**" and leave "**Revisions**" in the drop down box, enter the revision number in the text box and Click “**Get Revision**”.
How to edit files	• GUI Method 1. In your Pending Change list, select the file and open it for edit.
How to add new files	• CLI method `p4 add -c X *` Opens all the files within the user's current directory for add, and links these files to

	changelist X. This assumes that the numbered changelist already exists. `p4 submit -c X` Submit Changelist X to take effect. • GUI Method 1. P4V: Drag and drop files into the changelist. Submit Change list to take effect.
How to specify a file type	• GUI Method 1. Context-click the desired file and choose **Check Out**. The file is checked out. 2. Context-click the file and choose **Change Filetype…** The **Change Filetype** dialog is displayed. 3. Set the desired type and attributes and click **OK** to dismiss the dialog. 4. Submit the changelist containing the file.

How to delete files	• CLI Method
	`p4 delete //depot/dev/main/docs/manuals/vendor.doc`
	`p4 submit`
	• GUI Method
	1. Select File, right click and select **Mark for Delete**. Will be deleted upon successful submission of Changelist.

How to revert a file	• CLI Method `p4 revert <filename>` • GUI Method 1. Context click on the file in the Changelist and select **Revert**. Note: you have two options (Revert, and Revert if Unchanged)
How to find which files you are working on	• CLI Method 1. `p4 opened -C <client name>` Displays files opened in specified Client workspace

	2. `p4 opened -u <username>` Displays files opened by a user. - GUI Method 1. Select Pending, or View-->Pending and see the change list where the files you are working on.
How to find what was changed in files worked on	- GUI Method 1. Right click on the file in your changelist, select Diff against have revision.
How to submit changed files (changelist)	- CLI method 1. `p4 submit -c X` Where X is the changelist #. - GUI Method 1. Open pending Changelist on your workspace and select **Submit**
How to locate a file	- GUI method 1. **Search > Find File...**, Type part or whole filename Note: You have the option where to search (All depots, Workspace, or selected tree)
How to remove files from your workspace	- GUI method 1. In the Workplace, Select files, right click and select **Remove from Workspace**
How to view a deleted file	- GUI method 1. Click on the filter symbol; Select the option "**Show Deleted Depot files**". Deleted file will show with an "X" symbol next to it.
How to recover a deleted file	- CLI method To restore a file that was deleted in the depot, sync to the last good revision of the file, add it to source control, and then submit it. For example, if `foo.c` was deleted at revision 26, you restore it with: 1. `p4 sync foo.c#25` 2. `p4 add foo.c`

	3. `p4 submit` Or, if you submitted changelist 300 and accidentally deleted all the files in the //depot/main/src directory tree you can restore them with: 1. `p4 sync //depot/main/src...@299` 2. `p4 add //depot/main/src/...` 3. `p4 submit` • GUI method 1. Ensure that the deleted files are shown in the Depot View pane (go to the View menu, select the Filter Depot menu and click on Show Deleted Depot Files) 2. Locate the file in the Depot View pane 3. Context-click on the file and select "**Get Revision**" 4. In the "Get Revision" dialog box, click on the radio button for "**Specify revision using:**" and leave "**Revisions**" in the drop down box 5. In the empty box to the right of the "**Revisions**" drop down item, type in the number of the last good revision of the file 6. Click "**Get Revision**" 7. In the Depot View pane, context-click on the file and select "**Mark for Add**" 8. In the Depot View pane, context-click on the file and select "**Submit**" 9. Fill in the changelist description and click "**Submit**"
How to view a files Properties and History	• GUI method 1. Right click on the file and select **File History**
How to compare files	• GUI method 1. Select two files, right click and select "**Diff Selected**".
How to compare folders	• GUI method 1. Select two folders in the depot or workspace pane. You can also select one folder and enter the second folder directly in the Diff dialog. **2.** Context-click and choose **Diff Against....** 3. In the **Diff** dialog, specify the paths and versions of the folders you want to compare. 4. Click **Diff** to launch the Folder Diff utility.

	5. The **Folder Diff** utility lists the subfolders and files in both diffed folders, side-by-side.
How to use Time Lapse view	• GUI method 1. Right click on the file, and select "**Time-Lapse View**".
How to create and use Bookmarks	• GUI method 1. **Tools > Bookmarks >Add Bookmark…** 2. Or right click on the file and select **Bookmark…**
How to switch between Depot view & Workspace view	• GUI method To toggle between Depot and Workspace view: **View > Depot Tree** **View > Workspace tree**
How to update your Workspace to get latest files	• CLI method 1. `p4 Sync` • GUI method Context Click on the file or tree, select **Get Latest Revision**.
How to resolve file differences	Upon submission, Perforce will notify you if you need to resolve the file before submitting (Rev has been changed in the depot since you checked it out) • CLI method 1. `p4 resolve` • GUI method 1. Right click on the file, select **Resolve** and choose the option best fits 2. Options are: Accept Source, Accept Target, Accept merge or run merge tools.
How to resolve file differences automatically	• GUI method 1. At the Resolve screen, select Accept Merge (that is the automatic result of the merge yours and theirs)
How to resolve file differences manually	• GUI method

	1. Run Merge tools from the Resolve screen and edit the result as needed.
How to lock files	- GUI method 1. Select a file that you checked out in the depot, right click and select **Lock**. By default most binary files are already locked by default when you check them out, this is defined in the server typemap. If you would like to make a specific filetype or even an individual file by locked by default when you edit it then contact the Perforce System Administrator.
How to backout a recently submitted changelist – edits only	- CLI method In this example changelist 1000 contains edits only -- no files were added or deleted. Also, in this scenario, you *just* submitted changelist 1000 -- nothing has changed in your workspace since you did the submit, and no one else submitted any subsequent changes to the files in changelist 1000. To back out changelist 1000: 1. `p4 sync @999` 2. `p4 edit //depot/foo //depot/bar //depot/ola` 3. `p4 sync` 4. `p4 resolve -ay` 5. `p4 submit`
How to backout an old submitted changelist – edits only.	- CLI method In this scenario, changelist 1000 was submitted *some time ago*. Since the Submit, other good edits have been submitted to the same files. Backing out changelist 1000 is more complicated in this case because you have to preserve any subsequent changes to the files. In this example changelist 1000 contained file edits only -- no files were *added or deleted.* When backing out an old changelist, use a workspace with no opened files. You can use p4 opened to see if there are any opened files in your current workspace. To back out changelist 1000 when subsequent changes have been made to the files you: 1. `p4 sync @999` 2. `p4 edit //depot/foo //depot/bar //depot/ola` 3. `p4 sync @1000` 4. `p4 resolve -ay` 5. `p4 sync` 6. `p4 resolve` 7. `p4 submit`
How to backout an old submitted changelist – with edits, additions and	CLI method What do you do if there were added or deleted files in the bad changelist? You submit a new

deletions.	changelist to do the reverse: if a file was added, you delete it; if a file was deleted, you add it. This final scenario assumes changelist 1000 included an add, a delete, and an edit: To back out changelist 1000 in this scenario, you use the following steps: 1. `p4 sync @999` 2. `p4 edit //depot/ola` 3. `p4 add //depot/bar` 4. `p4 sync @1000` 5. `p4 resolve -ay` 6. `p4 sync` 7. `p4 resolve` 8. `p4 delete //depot/foo` 9. `p4 submit`
How to create a new Workspace or Workspaces	• GUI method You can have multiple workspaces on your workstation. **1. Connection>New Workspace** 2. Or at the workspace drop box, select **New Workspace…**
How to edit an existing Workspace	• GUI method **1. Connection>Edit Current Workspace**
How to view Workspace mappings	• GUI method 1. In the Edit Workspace, select View tab and navigate through the Depot tree (Include or Exclude view mapping of the depot to your workspace)
How to work offline	• GUI method If you have inadvertently lost connectivity to the shared Perforce service (for example, if the server machine crashes), you can continue working on files under Perforce control by setting file permissions manually. When the server becomes accessible again, use P4V's Reconcile feature to create the changelists required to update the server with the changes you made in your workspace. 1. After connecting to the server in P4V, Select the folder you would like to find added, edited, or deleted files. Perforce compares this folder with your client workspace's "have list"; that is, what files you last had.

	2. Context-click on the folder and select "**Reconcile offline work**". Note the "**Folder Diff**" pane appears. 3. Click the cyan blue arrow icons at the top to scroll through file differences, and note your Reconcile Options in the Details tab in the lower pane. The darkest line shows you what file or directory File Diff is focused on. • A purple highlight indicates a file has been modified. Select this file and context-click, select "Check Out" if you want to edit or check in your changes to Perforce. • A yellow (or brown) highlight indicates a file is new. Select this file and context-click, select "Mark for Add" to add the file to Perforce, and the file is removed from view. • A strikethrough indicates the file was deleted from your workspace. Select this file and context-click, select "Mark for Delete" if you want to delete this file from Perforce. Note that you can make multiple selections using the control and shift keys. Close this screen when you are finished. Select **View > Pending Changelists**, and see the right pane to see your changes. Submit these changes when you are ready.
Labels in Perforce	Perforce users typically rely on changelist numbers to identify the state of a codeline at any point in time. As a user, you can sync your workspace to a changelist number to retrieve a snapshot of the codeline at the point when the changelist was committed. Although Perforce strongly encourages the use of Changelists to keep track of files associated with an atomic unit of change to a codeline, labels, in Perforce, provide some useful functionality to augment the traditional use case for Changelists. One of the primary use cases for labels in Perforce is to apply a meaningful name to a set of files in your codeline. Labels v/s Changelists: • A label can refer to any set of file revisions. A changelist number refers to the contents of all the files in the depot at the time the changelist was submitted. • You can change the list the files associated with a label. You cannot change the list of files associated with a submitted changelist. • You can assign your own names to labels. Changelist numbers are assigned by the Perforce server. Note that label names cannot be the same as client workspace, branch, or depot names.
How to rename a file?	1. You **cannot** have the file checked out. If you do, submit your changes first. 2. Right click the file in the depot, select "Rename". ✓ This will essentially check the files out and branch them. 3. If you're renaming multiple files, you can combine the auto-created change lists into one.

	✓ Highlight the files you want to move into another changelist, right click and select "Move to another Changelist…" ✓ Select the appropriate changelist as the destination. Then remove the now empty changelist by right clicking and selecting "Delete Empty Changelist". **IMPORTANT!** You cannot make any other edits to the files you're renaming! You must submit the rename changelist, and then check the files out again before you can edit their contents!
How to integrate file's	• GUI method **To open files for integrate**: context-click the file and choose **Integrate…** The **Integrate** dialog is displayed. Specify integration options as follows: ✓ **Don't copy target files to workspace**: Create a branch in the depot without retrieving the corresponding files from the depot to your workspace. ✓ **Enable baseless merges**: Perform the integration even if source and target share no common ancestor, using the most-recently-added revision of the source file as the base. ✓ **Disregard all integration history**: Force the integration on all revisions of source and target files, even if some revisions have already been integrated. Typically used with a specified revision range. ✓ **Disregard indirect integration history**: Restrict selection of the base file revision to direct (parent-to-child or child-to-parent) integration history only. ✓ **Reverse mappings in branch view**: For integrations that use a branch mapping as opposed to a file specification, integrate using the right side of the branch mapping as source and the left as target, which is the reverse of the default direction. ✓ **Enable integrations around deleted revisions**: If the target file has been deleted and the source file changed, this option re-branches the source file on top of the target file. If the source file has been deleted and the target file has changed, this option deletes the target file. By default, outstanding edits cannot be integrated with a deleted file. ✓ **Do not get latest revision of selected files**: Use the workspace revision of the target file. By default, the head revision of the target file is automatically retrieved into the workspace before integrating. ✓ **Propagate source filetypes to target files**: Assign the target file the same file type as the source file (when integrating to existing target files).
How to manage branch mappings?	• GUI method 1. **To create a branch mapping**, choose **File>New>Branch Mapping.** and enter the required information. To prevent the mapping from being changed by other users, check **locked.** To confine integration to closely related files, choose **direct**; to enable integration between distantly related files, choose **indirect**. 2. **To delete a branch mapping**, click the mapping you want to delete, then select **Edit>Delete Branch** *branchname*. 3. **To change a branch mapping**, double-click it and enter your changes in

	the **Branch** form. Modify the **View** field to reflect the relationship between source and target codelines.
What DIFF tools can I use?	CLI method Perforce has a built in diff tool **(p4diff)** but users can plug in any licensed (or OSS) diff tool you wish. Perforce allows different diff tools on a per file extension basis. (i.e. You can use p4diff for plain text files such as VB or C# code and use specialized diff tools for Word or Excel docs) GUI method 1. In the depot pane, select the two files you want to diff. 2. Context-click and choose **Diff...** The Diff dialog is displayed 3. Specify the revisions of the files you want to diff and click **Diff**. P4V launches P4Merge, displaying the differences between the files at the specified revision.
What MERGE tools can I use?	CLI method Perforce has a built in 3 way merge tool (**p4merge**) but users can plug in any licensed (or OSS) merge tool you wish. Perforce allows different merge tools on a per file extension basis. GUI method 1. In the depot pane, select the two files you want to diff. 2. Context-click and choose **Diff...** The Diff dialog is displayed 3. Specify the revisions of the files you want to diff and click **Diff**. P4V launches P4Merge, displaying the differences between the files at the specified revision
What is Shelving?	*Shelving* enables you to store copies of open files temporarily in the shared Perforce repository without checking them in. Shelving is useful for a variety of purposes, including taking and restoring snapshots of in-progress work and reviewing other users' code before it's checked in. When you shelve a file, a copy is placed in a pending changelist from which other users can unshelve it. Pending changelists that contain shelved files are displayed with the following icon and badge: . When the changelist is expanded, shelved files are listed under the **Shelved Files** node, indicated with the following icon: .
Can I use Perforce remotely over the VPN?	Yes, Perforce works no differently remotely than locally on the LAN, and will be somewhat slower depending on the speed of your VPN connection. Perforce will only commit atomically, either the commit is perfect or it is not. Perforce achieves this by taking an md5 hash of each file it is submitting on the client side and when all the data has been copied to the server the md5 hashes will be verified by the server for accuracy, if any single byte did not get transferred accurately then the entire submission will fail and the user must try again.
What is 3rdParty Depot?	The 3rdparty depot is intended for source and binary distributions from external vendors. The contents of this depot are read only to all Perforce users. All applicable product licensing requirements must be followed when using files from this depot. No license keys or equivalent should be added to this depot. If a license key is required for an installation of software originating in this depot then the license key should be stored in a separate depot location that is protected from global view.

	Automated installation scripts should not be stored in this depot. Software should be in an installable state, not ZIP or ISO files. An exception to this is bootable media such as Windows or Linux install ISOs. These should be stored as ISO files in this depot as extracting them to their component files would lose their boot files. OS X installers may be stored as DMG files as that is the standard distribution media on OS X.
Branching in Perforce	Branching is a method of maintaining the relationship between sets of related files. Branches can evolve separately from their ancestors and descendants, and you can propagate (integrate) changes from one branch to another as desired. Perforce's Inter-File Branching™ mechanism preserves the relationship between files and their ancestors while consuming minimal resources.
Using Service Account for build related operations.	A service account can be setup for automated tasks and processes such as those that create builds or sync files to a web/file server. Service account is not intended for interactive logon and access is provided using a ticket. Regular user tickets have a 12 hour window for usage. Services account ticket can have an longer expiration date to support login cycle for up to a year. Please refer to section 6 for User Support and requesting such service account to be setup.
Where can I find Perforce documentation?	Perforce guides/documentation can be located in MyWorkshop as listed in Section 7.
What is shelved changelist promotion?	Perforce allows changelists shelved on an edge server, which would normally be inaccessible from Commit server and other edge servers. These will be private to edge server and need to be *promoted* to the commit server, for global visibility. After promotion, shelve will be visible for others connected to commit/other edge server. Any change to shelve changelist from edge will now reflect in the commit server.

6 USER SUPPORT

Perforce Application support is managed thru EAST (Engineering Agility Support Team) website. Users can create a case on the EAST site for all perforce relates queries/requests. User will assign a Severity Level (Business Critical, Urgent, and Normal) to each case created on EAST Site; accordingly the Support Engineer will process the request. Severity Levels are associated with different Service Level Agreement (SLA), and overview is provided in the HCIT Tools Support Guide [8]

Steps to request Perforce Application Support:

1. Please file a Support Request in Support Central here: http://sc.ge.com/@HCIT+EAST
2. Select "SW Request and Tools info"
3. Select "Yes" for Is the Software Tool you need in the Strategic Tool List above?:*
4. Select "Perforce" from Which Tool in Toolbox do you need?:* drop-down and fill out the form

Sample Type of Requests:

1. User Administration: Addition/Removal/Account Modification/Account Reinstatement
2. Application Issues: Application Down/Slowness/Functionality Issues.
3. Others: IDE/Tools Integration, General Issues

For any immediate break-fix, a User can email Perforce Administrators at hcit.perforce.admins@ge.com as well.

7 REFERENCES

The table below lists the referenced documents and their My Workshop Identification numbers:

Table 7-1: Reference Documents

S. No	Document Description	Doc ID
1	Technical Architecture Description for HC Centralized Perforce	DOC1172013
2	HC Perforce 2014.1 - Introduction to Perforce	DOC1270733
3	HC Perforce 2014.1 - P4 Users Guide	DOC1270735
4	HC Perforce 2014.1 - Getting Started with P4V	DOC1270741
5	HC Perforce 2014.1 - System Administrators Guide	DOC1270753
6	HC Perforce 2014.1 - Guide for Command Reference	DOC1270757
7	HC Perforce 2014.1 – P4VS Plugin User Guide	DOC1410358
8	HC Engineering Tools - Support Guide	DOC1410371