## **EXPERIMENT 4**

## **Objectives**

* Explain how to resolve the conflict during merge.

In this hands-on lab, you will learn how to:

* Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.

## **Prerequisites**

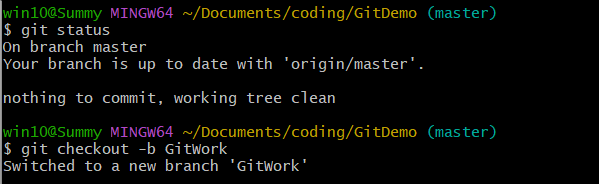
The following are the pre-requisites to complete this hands-on lab:

* Hands-on ID: **“Git-T03-HOL\_001”**

Estimated time to complete this lab: **30 minutes.**

Please follow the instructions to complete the hands-on. Each instruction expect a command for the Git Bash.

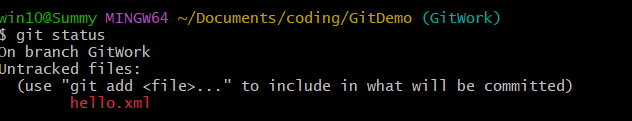
* Verify if master is in clean state.



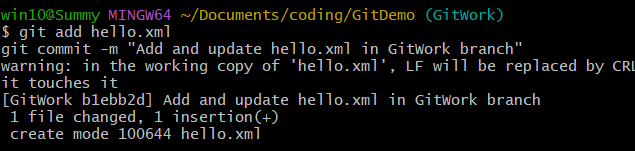
* Create a branch **“GitWork”.** Add a file “hello.xml”.



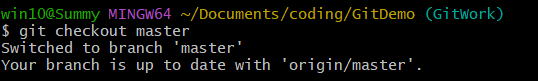
* Update the content of “hello.xml” and observe the status



* Commit the changes to reflect in the branch



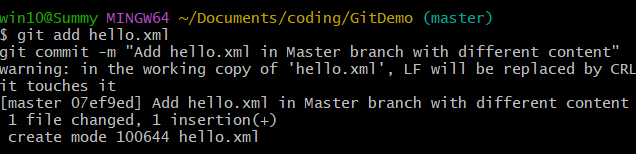
* Switch to master.



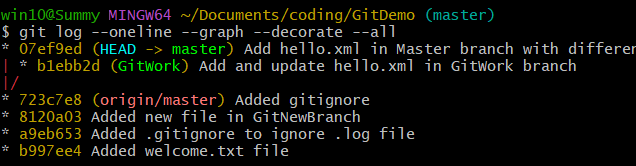
* Add a file **“hello.xml”** to the master and add some different content than previous.



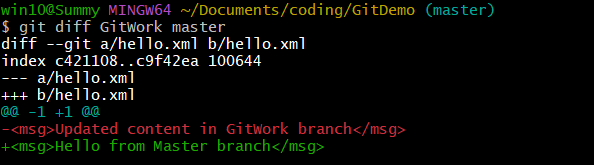
* Commit the changes to the master



* Observe the log by executing **“git log –online –graph –decorate –all”**

****

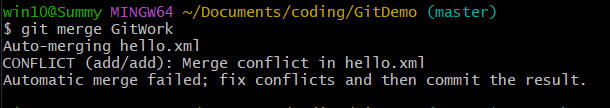
* Check the differences with Git diff tool

****

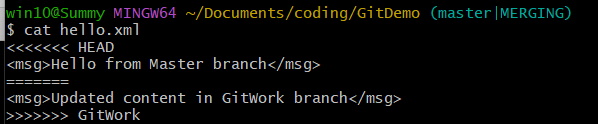
* For better visualization, use P4Merge tool to list out all the differences between master and branch

****

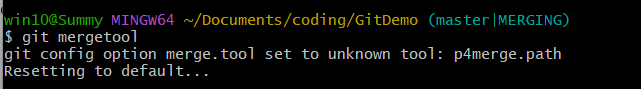
* Merge the bran to the master

****

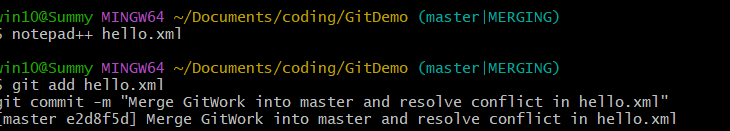
* Observe the git mark up.

****

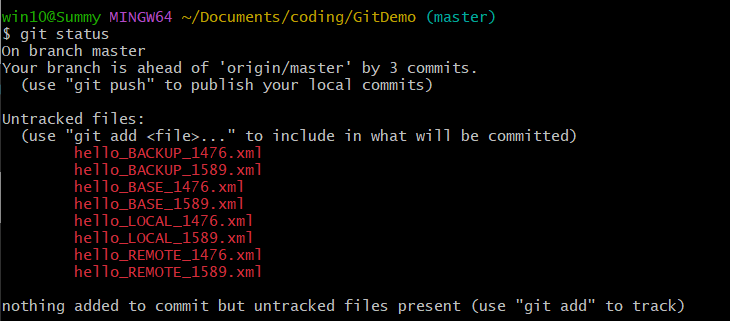
* Use 3-way merge tool to resolve the conflict

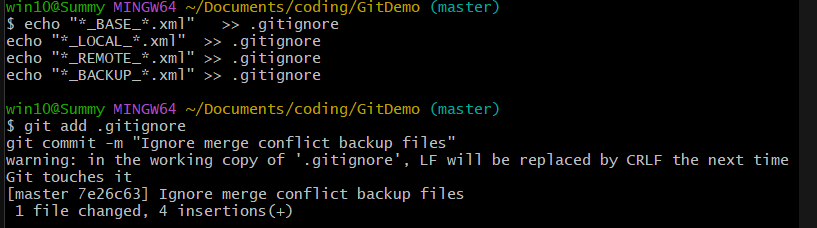
****

* Commit the changes to the master, once done with conflict

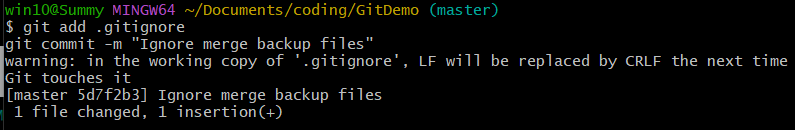
****

* Observe the git status and add backup file to the .gitignore file.

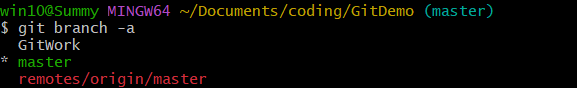
****

****

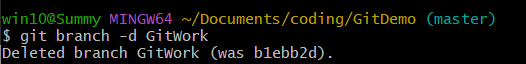
* Commit the changes to the .gitignore

****

* List out all the available branches

****

* Delete the branch, which merge to master.

****

* Observe the log by executing **“git log –oneline –graph –decorate”**

