



# VM Workshop Presentation

Paul Flint

Barre Open Systems Institute  
[flint@flint.com](mailto:flint@flint.com)



# Ten Statement git

Next- MUSICAL OVERTURE

Aw...

We Love The git

Give Us Some git

# Hello World

I am Paul Flint, Director of the  
Barre Open Systems Institute  
(BOSI)

- Recovering Systems Security Consultant
- Disaster Recovery Specialist
- Vita <http://docbox.flint.com>

# Housekeeping

- In order to fully participate in this presentation you will need:
  - A quiet cell phone
  - A laptop with internet connection running Debian or a derivative. (I have a few with me...)
  - House rules are that If you win a prize you cannot consume it in the room.

# Talk Concept

In my life I have been fortunate to travel to non English speaking places. Despite language barriers I learned that if you know a few local language phrases you can get by.

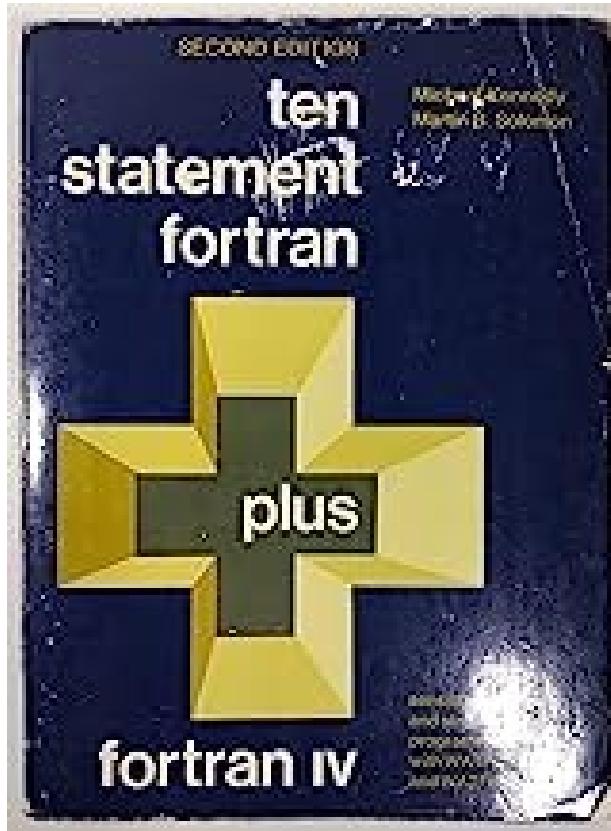
Examples include:

Hello

How much is this?

Where is the bathroom?

# Thus the basis of this Presentation is...



I took a FORTRAN course at Georgetown University, and this is the book they used...

Goal of this presentation  
is to show you how to  
use git to preserve your  
software work product.

# The Five Most Hated Words In ADP

1. DID
2. YOU
3. BACK
4. IT
5. UP ?

First and Foremost, GIT Is About  
Software Survival!

# About GIT...

- In 2005 Bitkeeper revoked its license for Linux kernel development.
- Under pressure, git is immediately developed by the open source community, this effort is spear-headed by Linus Torvalds for ongoing development of the Linux kernel.
- Released later in 2005, it is a bit of understated masterpiece...

# Scope Limit

- Git can be used for but not in this discussion:
  - Binary Backup
  - Code reconciliation
  - Project Control
  - Change and Configuration Management
  - World Domination...
- I just want to save your software...

# Git in a Nutshell

- There are two parts to git:
  - Git Commands
  - Git Repositories
- These are always used together!
- Your data is what git stores in repositories

# Git Repositories

- Your work is what git stores in repositories
- There are two types of repositories
  - Local (always)
  - Remote (cloud based)
- Local repository is always there
- There are various cloud repository hosts.  
These include...

# Cloud repositories for git

- Bitbucket
- GitLab
- Perforce
- Beanstalk
- Amazon AWS CodeCommit
- Codebase
- Codeburg
- SourceForge

# GitHub – 800 Pound Gorilla

## History

- Originally a 2007 startup
- 2014 Blacklisted by Russia
- Acquired by Micro\$oft in 2018

# github.com

- We will use github.com as an example
  - There are differences between various remote repositories
- Github.com has some unique security features
  - These are not open source
- The GitHub interface is not entirely standard, but the commands we use are...

# The Ten Statements

- clone
- init
- add
- commit
- status
- remote
- push
- pull
- revert
- log

# 1. Init

## git init

- Init creates a hidden local directory called “.git”
- Wise folks stay out of this local repository
- If cloning no need to init...
- It begins with init

## 2. Clone

```
git clone git@github.com:<userid>/<repository>
```

- Downloads remote repository
- You will learn faster by cloning
- If cloning no need to init.
- If all else fails download the zip

The Best Baby Step

### 3. Add

```
git add <filespec>
```

<filespec> can be:

- Individual File
- Range of Files
- Type of file
- Directory
- \*.sh
- (The Dreaded) \* .\*

## 4. commit

```
git commit -m "description"
```

A Commit records and summarizes the changes to your repository

“Description” can be:

- Version Number
- Description Text
- Criticism

“Description” should be:

- Process Related
- Constructive
- A Public Comment

# 5. Status

git status

- Shows you what is going on in both the local and remote repository.

# 8. Remote

git remote -v

- Tells you if you are in a repository
- Tells you what hosts your repository has been saved to
- A way to test git's capability to transfer to github
- If public private key stuff is working, use “gh auth login”

# 6. Push

git push

- Updates the remote repository
  - As a registered user of a repository

## 7. Pull

git pull

- Pull delivers code to your local repository and workplace from the remote cloud repository.

## 9. Restore

```
git restore <filespec>
```

- Recover from local backup.
- Alternative to renaming local repository and cloning again...
- Not a mainstream action.

# 10. Log

git log

- Shows changes in repositories
- Changes in reverse chronological order
  - Newest is first...

# Everyone runs a git repository.

- The Git Client Origin and Master files
- The Git repository server(s)
  - Local
  - Remote (Cloud(s))

# OPTIONS

- .gitignore
- git ls-files
- git branch
- git checkout
- gh – A useful GitHub extension

# gitignore

## .gitignore

- This is not a command
- A hidden text file in your directory that tells git what files to not involve with the git process.
- I recommend stringent use of this to avoid “git noise”

# git branch

git branch <new name>

- This creates a new branch with a new name. Edits, adds, commits, etc will only be in the branch that you're currently working with.
- If no <new name> given then git branch returns current branch.
- Good branch names could be:
  1. Development
  2. Production
  3. Test

Sound familiar?

*Everyone Has A Test System*  
*Some Are Lucky Enough To Have*  
*A Production System As Well*

...The Wit and Wisdom of Dave-the-Geek

# git checkout

git checkout <branch name>

- checkout switches which branch you are working on.

gh

- gh is an open source extension of the basic git command set.
- gh auth login – gets you on github.com easily after you are registered...

# git revert

```
git revert <file>
```

- Allows you to recover a file from the repository.
- Backup locally and use with caution.
- Alternatively, rename current directory and pull again.

# The Dreaded Demonstration

- I have not written a Red Hat version of this demonstration.
- If you have a Debian based system, proceed.
- I have a few loaner laptops with me for those who wish to try to git...

The site to access this demo is:

<https://github.com/flintiii/tsg>

Use your browser...

# Could There Be A git For VM?

- Write in REXX?
- Portability within IBM code space
- A tool for Change & Configuration Management?
- BOSI could help.

This could be a discussion point...

# BOSI Power Point Heros!

These folks >

Shared with me the  
Tedious Process of  
Turning Ten Statement  
Git into a presentation

- Judith Bramlage
- Kevin Cole
- Chris Roland
- Joseph Hart

With Special Thanks to:

Dave “The Geek” Lafrenier

# Thank you for your patience!

Paul Flint, Director  
Barre Open Systems Institute  
(BOSI)

- Vita <http://docbox.flint.com>
- [flint@flint.com](mailto:flint@flint.com)

Questions, Answers, Observations  
and/or Criticism Shall Be Rewarded

