

**Bart  
Trojanowski,  
bart@jukie.net**

# Git the basics

Bart Trojanowski, bart@jukie.net

Jukie Networks Inc.

July 9th, 2008

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends

-

### Concepts

- SCM components
- SCM operations
- Decentralization

### GIT History

#### Repository

- Structure
- Objects

#### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends

-

Next...

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

### Repository

Structure

Objects

### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

### Repository

Structure

Objects

### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Concepts

- ▶ Source Control Management
  - ▶ track changes to files
  - ▶ repository / database of changes
  - ▶ working directory / current state
- ▶ Centralized SCM
  - ▶ server: single database
  - ▶ client: working directory & state
- ▶ Decentralized SCM
  - ▶ anyone can be a server
  - ▶ repository coupled with working directory
  - ▶ complete history
  - ▶ disconnected operation

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Concepts

- ▶ Source Control Management
  - ▶ track changes to files
  - ▶ repository / database of changes
  - ▶ working directory / current state
- ▶ Centralized SCM
  - ▶ server: single database
  - ▶ client: working directory & state
- ▶ Decentralized SCM
  - ▶ anyone can be a server
  - ▶ repository coupled with working directory
  - ▶ complete history
  - ▶ disconnected operation

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

- ▶ Source Control Management
  - ▶ track changes to files
  - ▶ repository / database of changes
  - ▶ working directory / current state
- ▶ Centralized SCM
  - ▶ server: single database
  - ▶ client: working directory & state
- ▶ Decentralized SCM
  - ▶ anyone can be a server
  - ▶ repository coupled with working directory
  - ▶ complete history
  - ▶ disconnected operation

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

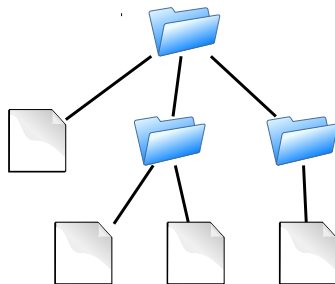
Loose ends

-

# SCM components

## Working tree

- ▶ directories
- ▶ files



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-



# SCM components

## Repository contents

- ▶ files



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

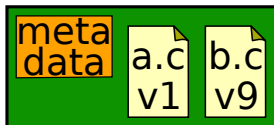
Loose ends

-

# SCM components

## Repository contents

- ▶ files
- ▶ commits



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

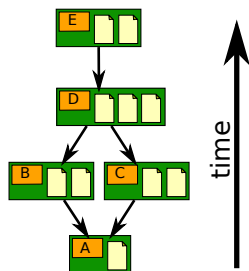
Loose ends

-

# SCM components

## Repository contents

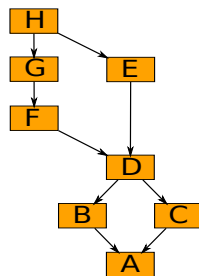
- ▶ files
- ▶ commits
- ▶ ancestry



# SCM components

directed acyclic graph

“DAG”



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# SCM components

## References

- ▶ tags

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

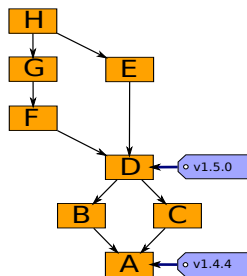
Rebasing

Remotes

GUI tools

Loose ends

-



# SCM components

## References

- ▶ tags
- ▶ branches

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

**SCM components**

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

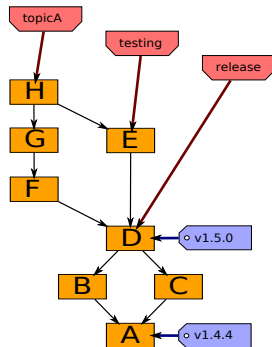
Rebasing

Remotes

GUI tools

Loose ends

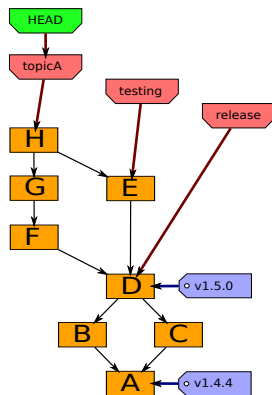
-



# SCM components

## HEAD

- ▶ current checkout
- ▶ points to branch



### Concepts

#### SCM components

SCM operations

Decentralization

### GIT History

#### Repository

Structure

Objects

#### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

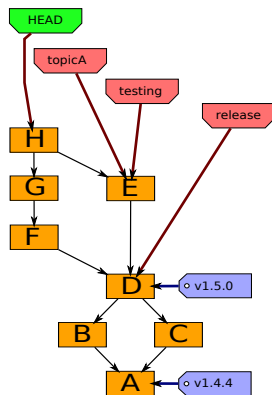
Loose ends

-

# SCM components

## HEAD

- ▶ current checkout
- ▶ points to branch
- ▶ sometimes detached

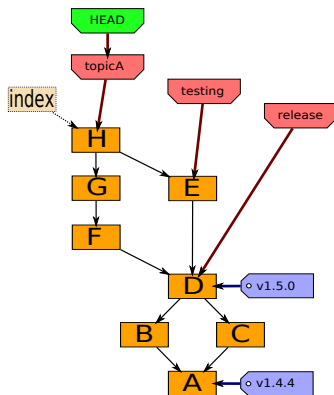




# SCM components

## Index

- ▶ “staging area”
- ▶ what is to be committed



## Git the basics

Bart  
Trojanowski,  
bart@jukie.net

### Concepts

SCM components

**SCM operations**

Decentralization

### GIT History

#### Repository

Structure

Objects

#### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# SCM operations

## Bootstrap

- ▶ init
- ▶ checkout
- ▶ switch branch

## Modify

- ▶ add, delete, rename
- ▶ commit

## Information

- ▶ status
- ▶ diff
- ▶ log

## Reference

- ▶ tag
- ▶ branch

# SCM operations

## Bootstrap

- ▶ init
- ▶ checkout
- ▶ switch branch

## Modify

- ▶ add, delete, rename
- ▶ commit

## Information

- ▶ status
- ▶ diff
- ▶ log

## Reference

- ▶ tag
- ▶ branch

# SCM operations

## Bootstrap

- ▶ init
- ▶ checkout
- ▶ switch branch

## Modify

- ▶ add, delete, rename
- ▶ commit

## Information

- ▶ status
- ▶ diff
- ▶ log

## Reference

- ▶ tag
- ▶ branch

# SCM operations

## Bootstrap

- ▶ init
- ▶ checkout
- ▶ switch branch

## Modify

- ▶ add, delete, rename
- ▶ commit

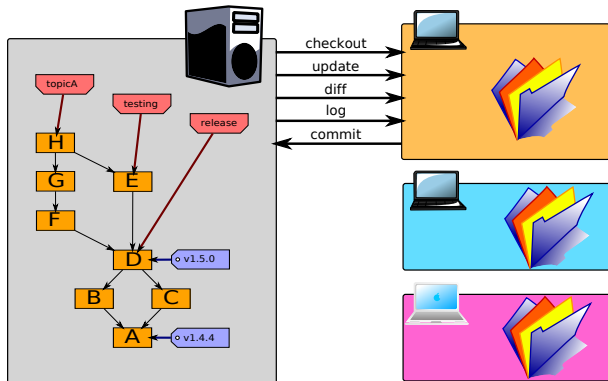
## Information

- ▶ status
- ▶ diff
- ▶ log

## Reference

- ▶ tag
- ▶ branch

# Centralized SCM



- ▶ operations require **server**
  - ▶ single point of failure
  - ▶ bottleneck

## Concepts

SCM components

**SCM operations**

Decentralization

## GIT History

### Repository

Structure

Objects

### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# more SCM operations

## Decentralized

- ▶ clone
- ▶ pull, fetch
- ▶ push

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

**SCM operations**

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

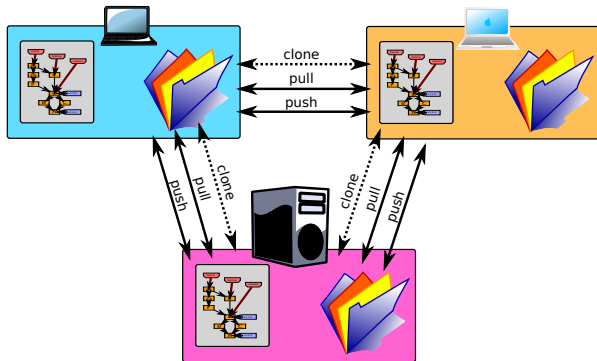
GUI tools

Loose ends

-



# Decentralized SCM



- ▶ anyone can be a server

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

**SCM operations**

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

**Decentralization**

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

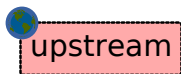
Remotes

GUI tools

Loose ends

-

# Decentralization



- ▶ public repository

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

**Decentralization**

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

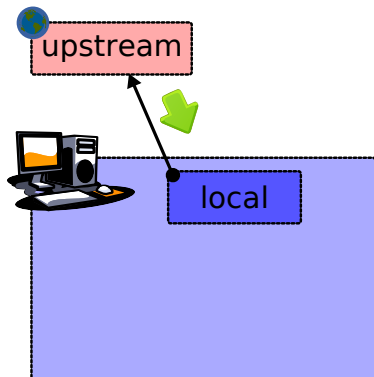
Remotes

GUI tools

Loose ends

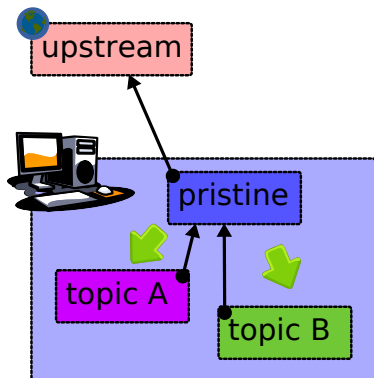
-

# Decentralization



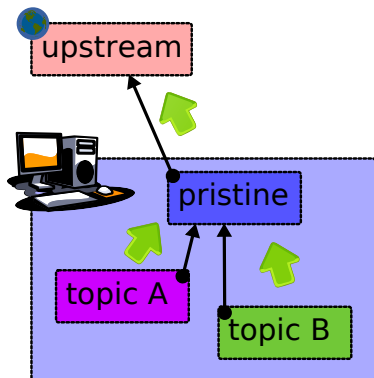
- ▶ make a local clone

# Decentralization



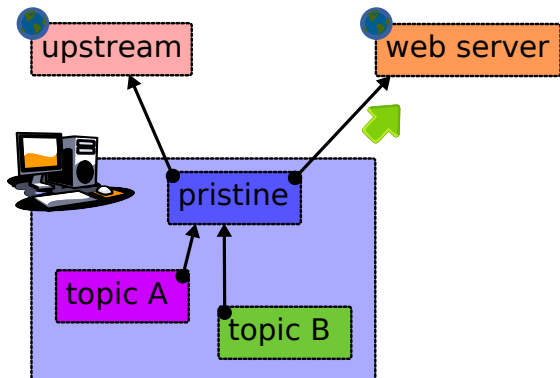
- ▶ local cloning is lightweight

# Decentralization



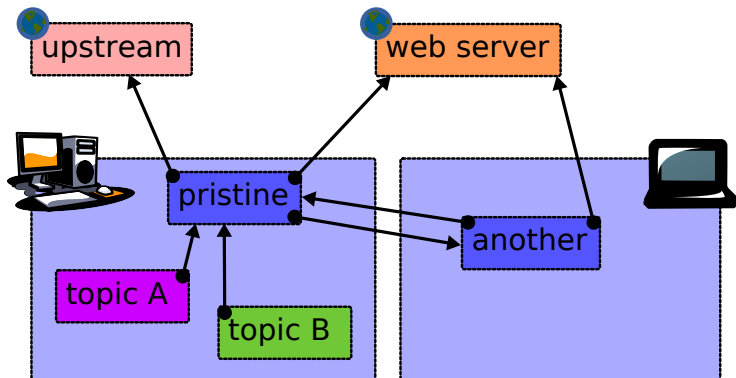
- ▶ push changes between any repositories

# Decentralization



- ▶ publish changes to public server

# Decentralization



- ▶ share changes with trusted peers



# Is Decentralization any good?

- ▶ non-intrusive micro-commits
- ▶ detached operation
- ▶ no single point of failure
- ▶ backups are trivial

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

**Decentralization**

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Next...

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

### Repository

Structure

Objects

### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

### Repository

Structure

Objects

### Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Birth of GIT

- ▶ 2002
  - ▶ Linus uses BitKeeper for tracking Linux
  - ▶ BK gets better
  - ▶ Linux development scales better
- ▶ April 6, 2005
  - ▶ BitMover drops free license
  - ▶ Linus writes his own SCM, GIT
- ▶ April 18, 2005
  - ▶ GIT can merge
- ▶ June 16, 2005
  - ▶ GIT is officially used to track Linux
- ▶ Feb 14, 2007
  - ▶ GIT 1.5.0 is released
  - ▶ major usability effort

# Birth of GIT

- ▶ 2002
  - ▶ Linus uses BitKeeper for tracking Linux
  - ▶ BK gets better
  - ▶ Linux development scales better
- ▶ April 6, 2005
  - ▶ BitMover drops free license
  - ▶ Linus writes his own SCM, GIT
- ▶ April 18, 2005
  - ▶ GIT can merge
- ▶ June 16, 2005
  - ▶ GIT is officially used to track Linux
- ▶ Feb 14, 2007
  - ▶ GIT 1.5.0 is released
  - ▶ major usability effort

# Birth of GIT

- ▶ 2002
  - ▶ Linus uses BitKeeper for tracking Linux
  - ▶ BK gets better
  - ▶ Linux development scales better
- ▶ April 6, 2005
  - ▶ BitMover drops free license
  - ▶ Linus writes his own SCM, GIT
- ▶ April 18, 2005
  - ▶ GIT can merge
- ▶ June 16, 2005
  - ▶ GIT is officially used to track Linux
- ▶ Feb 14, 2007
  - ▶ GIT 1.5.0 is released
  - ▶ major usability effort

# Birth of GIT

- ▶ 2002
  - ▶ Linus uses BitKeeper for tracking Linux
  - ▶ BK gets better
  - ▶ Linux development scales better
- ▶ April 6, 2005
  - ▶ BitMover drops free license
  - ▶ Linus writes his own SCM, GIT
- ▶ April 18, 2005
  - ▶ GIT can merge
- ▶ June 16, 2005
  - ▶ GIT is officially used to track Linux
- ▶ Feb 14, 2007
  - ▶ GIT 1.5.0 is released
  - ▶ major usability effort

# Birth of GIT

- ▶ 2002
  - ▶ Linus uses BitKeeper for tracking Linux
  - ▶ BK gets better
  - ▶ Linux development scales better
- ▶ April 6, 2005
  - ▶ BitMover drops free license
  - ▶ Linus writes his own SCM, GIT
- ▶ April 18, 2005
  - ▶ GIT can merge
- ▶ June 16, 2005
  - ▶ GIT is officially used to track Linux
- ▶ Feb 14, 2007
  - ▶ GIT 1.5.0 is released
  - ▶ major usability effort

# GIT gets better

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

*And then realize that nothing is perfect. Git is just  
\*closer\* to perfect than any other SCM out there.  
-Linus*



# Next...

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

## Repository

Structure

Objects

## Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

**Repository**

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

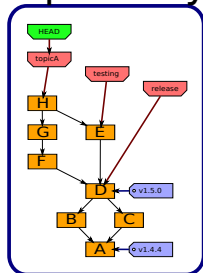
**Structure**  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Structure

## repository



the history

### Concepts

SCM components  
SCM operations  
Decentralization

### GIT History

### Repository

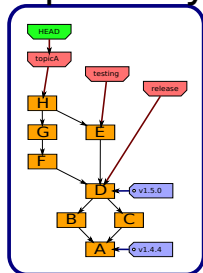
**Structure**  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Structure

## repository



## index

```
100644 20b024 0 bar
100644 1d52a6 0 baz
100644 20b024 0 sub/ffi
100644 43dbe0 0 sub/foo
```

staging area

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

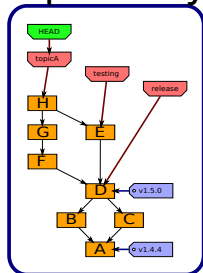
**Structure**  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Structure

## repository



## index

```
100644 20b024 0 bar
100644 1d52a6 0 baz
100644 20b024 0 sub/foo
100644 43dbe0 0 sub/foo
```

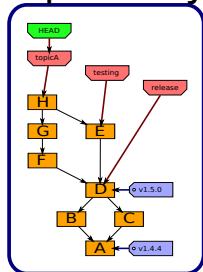
## work tree

```
|-- bar
|-- baz
|-- sub
|   |-- fi
|   |-- foo
```

files you edit

# Structure

## repository



## index

```
100644 20b024 0 bar
100644 1d52a6 0 baz
100644 20b024 0 sub/foo
100644 43dbe0 0 sub/foo
```

## work tree

```
|-- bar
|-- baz
|-- sub
|   |-- fi
|   |-- foo
```



“staging”

add  
remove  
rename

### Concepts

SCM components  
SCM operations  
Decentralization

### GIT History

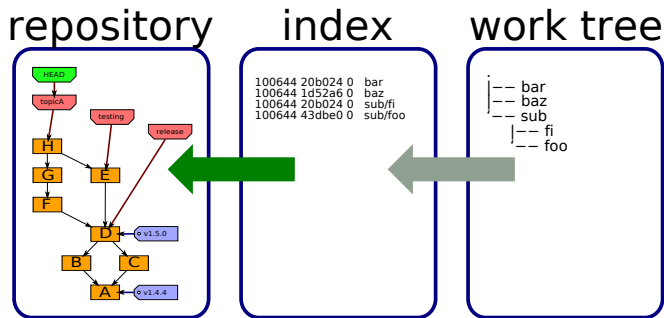
### Repository

**Structure**  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Structure



“committing”

commit

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

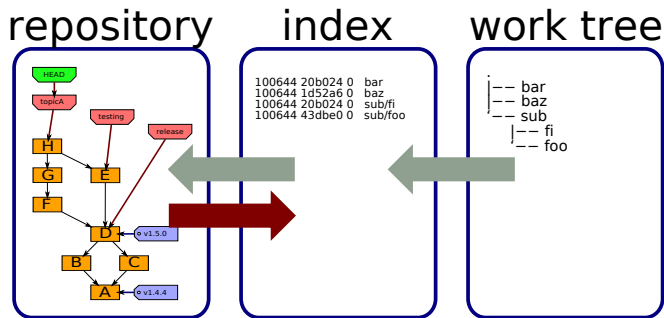
## Repository

**Structure**  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Structure



“reading tree”

checkout  
read-tree  
reset

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

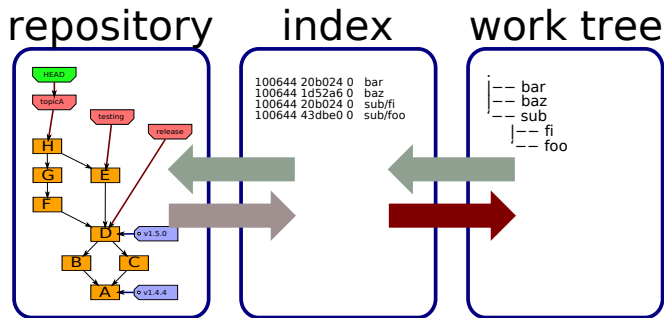
**Structure**  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Structure



“checking out”

checkout  
checkout-index  
reset

# The repository

```
.git
|-- HEAD           current checkout reference
|-- config        repo private config
|-- description   repo description
|-- hooks
|   '-- ...       hooking scripts
|-- index         changes to commit
|-- info
|   |-- exclude   repo private
|   '-- refs      refs?
|-- logs
|   '-- ...       "reflog" data
|-- objects
|   |-- XX
|   |   '-- ...   loose objects
|   |-- info
|   |   '-- packs info about packs
|   '-- pack
|       '-- ...   packs and indexes
'-- refs
    |-- heads
    |   '-- master master branch
    '-- tags
        '-- ...   tags
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure**
- Objects

## Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# The repository

`.git/config`

- ▶ repository config

`.git/description`

- ▶ describes the repository  
useful for gitweb

`.git/info/exclude`

- ▶ patterns to ignore

# The repository

`.git/config`

- ▶ repository config

`.git/description`

- ▶ describes the repository  
useful for gitweb

`.git/info/exclude`

- ▶ patterns to ignore

# The repository

`.git/config`

- ▶ repository config

`.git/description`

- ▶ describes the repository  
useful for gitweb

`.git/info/exclude`

- ▶ patterns to ignore

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Objects

```
.git/objects
```

```
| -- 23
```

```
| ' -- d4bd826aba9e29aaace9411cc175b784edc399
```

```
| -- 76
```

```
| ' -- 49f82d40a98b1ba59057798e47aab2a99a11d3
```

```
| -- c4
```

```
| ' -- aaefaa8a48ad4ad379dc1002b78f1a3e4ceabc
```

```
| -- e7
```

```
| ' -- 4be61128eef713459ca4e32398d689fe80864e
```

```
| -- info
```

```
| ' -- packs
```

```
' -- pack
```

```
| -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.idx
```

```
' -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.pack
```

# Objects

```
.git/objects
```

```
| -- 23
```

```
| ' -- d4bd826aba9e29aaace9411cc175b784edc399
```

```
| -- 76
```

```
| ' -- 49f82d40a98b1ba59057798e47aab2a99a11d3
```

```
| -- c4
```

```
| ' -- aaefaa8a48ad4ad379dc1002b78f1a3e4ceabc
```

```
| -- e7
```

```
| ' -- 4be61128eef713459ca4e32398d689fe80864e
```

```
| -- info
```

```
| ' -- packs
```

```
' -- pack
```

```
| -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.idx
```

```
' -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.pack
```

“loose objects”



# Objects

```
.git/objects
```

```
| -- 23
```

```
| ' -- d4bd826aba9e29aaace9411cc175b784edc399
```

```
| -- 76
```

```
| ' -- 49f82d40a98b1ba59057798e47aab2a99a11d3
```

```
| -- c4
```

```
| ' -- aaefaa8a48ad4ad379dc1002b78f1a3e4ceabc
```

```
| -- e7
```

```
| ' -- 4be61128eef713459ca4e32398d689fe80864e
```

```
| -- info
```

```
| ' -- packs
```

```
' -- pack
```

```
| -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.idx
```

```
' -- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.pack
```

“pack file”

# Objects

content addressable

type	size
data	

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

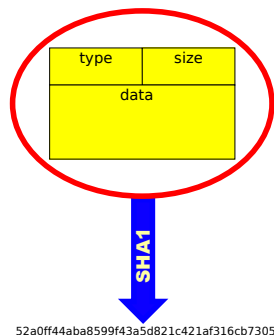
GUI tools

Loose ends

-

# Objects

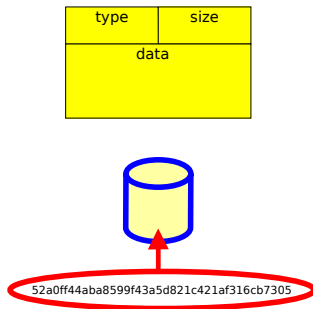
content addressable



52a0ff44aba8599f43a5d821c421af316cb7305

# Objects

content addressable



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

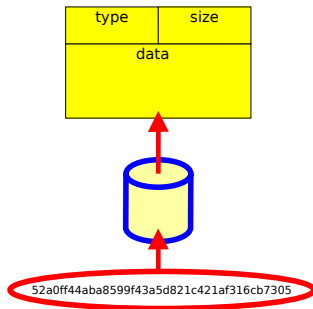
GUI tools

Loose ends

-

# Objects

content addressable



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
**Objects**

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

4 types

- ▶ blobs

<b>"blob"</b>	size
file data	

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

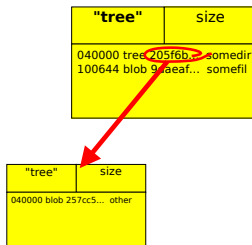
Loose ends

-

# Objects

## 4 types

- ▶ blobs
- ▶ trees



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
**Objects**

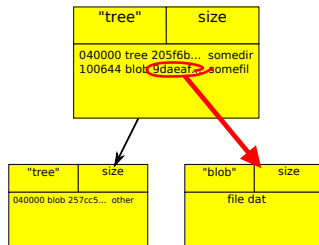
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

## 4 types

- ▶ blobs
- ▶ trees



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
**Objects**

## Using GIT

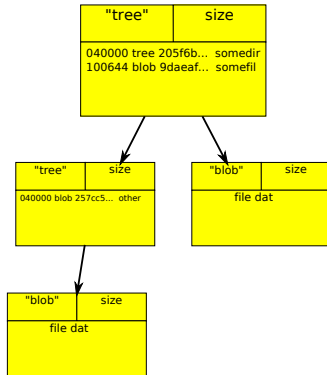
Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Objects

4 types

- ▶ blobs
- ▶ trees



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
**Objects**

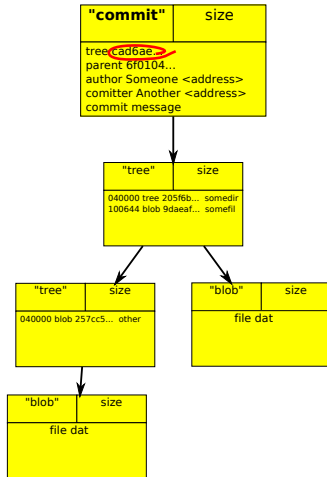
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

## 4 types

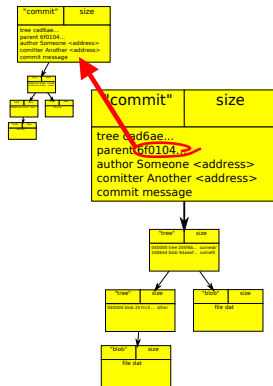
- ▶ blobs
- ▶ trees
- ▶ commits



# Objects

## 4 types

- ▶ blobs
- ▶ trees
- ▶ commits



### Concepts

SCM components  
SCM operations  
Decentralization

### GIT History

### Repository

Structure  
**Objects**

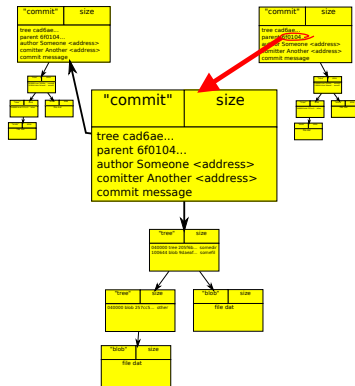
### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

## 4 types

- ▶ blobs
- ▶ trees
- ▶ commits



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
**Objects**

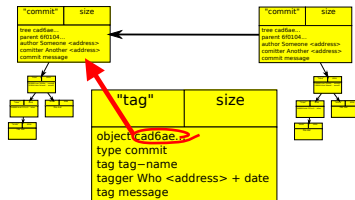
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

## 4 types

- ▶ blobs
- ▶ trees
- ▶ commits
- ▶ tags



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

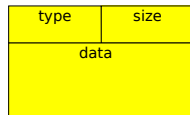
Structure  
**Objects**

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Objects

immutable



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

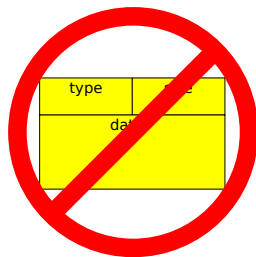
GUI tools

Loose ends

-

# Objects

immutable



type	size
data	

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

**Objects**

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

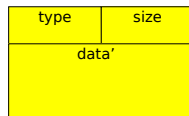
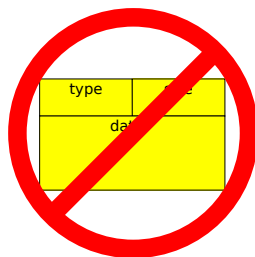
GUI tools

Loose ends

-

# Objects

immutable



8a9fd66b8bb939638564ebfdecc23b5d58070fc8



# Next...

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

## Repository

Structure

Objects

## Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

**Commands**

Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Git commands

```
$ git <options> <command> <options>
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

**Commands**

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Git commands (137)

add	fast-export	merge-one-file	revert
am	fast-import	merge-resolve	rm
annotate	fetch	merge-subtree	send-email
apply	fetch-pack	merge-tree	send-pack
archimport	filter-branch	mergetool	sh-setup
archive	fmt-merge-msg	mktag	shell
bisect	for-each-ref	mtree	shortlog
blame	format-patch	mv	show
branch	fsck	name-rev	show-branch
bundle	fsck-objects	pack-objects	show-index
cat-file	gc	pack-redundant	show-ref
check-attr	get-tar-commit-id	pack-refs	stash
check-ref-format	grep	parse-remote	status
checkout	gui	patch-id	strip-space
checkout-index	hash-object	peek-remote	submodule
cherry	http-fetch	prune	svn
cherry-pick	http-push	prune-packed	symbolic-ref
citool	imap-send	pull	tag
clean	index-pack	push	tar-tree
clone	init	quiltimport	unpack-file
commit	init-db	read-tree	unpack-objects
commit-tree	instaweb	rebase	update-index
config	log	receive-pack	update-ref
count-objects	lost-found	reflog	update-server-info
cvsexportcommit	ls-files	relink	upload-archive
cvsimport	ls-remote	remote	upload-pack
cvsserver	ls-tree	repack	var
daemon	mailinfo	repo-config	verify-pack
describe	mailsplit	request-pull	verify-tag
diff	merge	rerere	what-changed
diff-files	merge-base	reset	write-tree
diff-index	merge-file	rev-list	
diff-tree	merge-index	rev-parse	gitk

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Every day use. . .

`add`

`fetch`

`rm`

`branch`

`mv`

`show`

`checkout`

`gc`

`grep`

`stash`  
`status`

`clone`  
`commit`

`init`

`pull`  
`push`

`tag`

`config`

`log`

`rebase`

`remote`

`diff`

`merge`

`reset`

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

**Commands**  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Some GUI tools...

add

fetch

rm

mergetool

shell

branch

mv

show

gc

stash

checkout

grep  
gui

status

citool

pull  
push

tag

clone  
commit

init

config

log

rebase

remote

diff

merge

reset

gitk

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

**Commands**

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Occasional use...

add			revert
am			rm
annotate	fetch		send-email
apply			
		mergetool	
archive			shell
bisect			shortlog
blame	format-patch	mv	show
branch			show-branch
bundle			
	gc		
	grep		stash
checkout	gui		status
			submodule
cherry			svn
cherry-pick			
citool		pull	tag
clean		push	
clone	init	quiltimport	
commit			
	instaweb	rebase	
config	log		
		reflog	update-server-info
		remote	
describe			
diff	merge	reset	what changed
			gitk

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

**Commands**

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# And the plumbing...

<b>add</b>	fast-export	merge-one-file	<b>revert</b>
<b>am</b>	fast-import	merge-resolve	<b>rm</b>
<b>annotate</b>	<b>fetch</b>	merge-subtree	<b>send-email</b>
<b>apply</b>	fetch-pack	merge-tree	send-pack
archimport	filter-branch	<b>mergetool</b>	sh-setup
<b>archive</b>	fmt-merge-msg	mktag	<b>shell</b>
<b>bisect</b>	for-each-ref	mtree	<b>shortlog</b>
<b>blame</b>	<b>format-patch</b>	<b>mv</b>	<b>show</b>
<b>branch</b>	fsck	name-rev	<b>show-branch</b>
<b>bundle</b>	fsck-objects	pack-objects	show-index
cat-file	<b>gc</b>	pack-redundant	show-ref
check-attr	get-tar-commit-id	pack-refs	<b>stash</b>
check-ref-format	<b>grep</b>	parse-remote	<b>status</b>
<b>checkout</b>	<b>gui</b>	patch-id	strip-space
checkout-index	hash-object	peek-remote	<b>submodule</b>
<b>cherry</b>	http-fetch	prune	<b>svn</b>
<b>cherry-pick</b>	http-push	prune-packed	symbolic-ref
<b>citool</b>	imap-send	<b>pull</b>	<b>tag</b>
<b>clean</b>	index-pack	<b>push</b>	tar-tree
<b>clone</b>	<b>init</b>	<b>quiltimport</b>	unpack-file
<b>commit</b>	init-db	read-tree	unpack-objects
commit-tree	<b>instaweb</b>	<b>rebase</b>	update-index
<b>config</b>	<b>log</b>	receive-pack	update-ref
count-objects	lost-found	<b>reflog</b>	<b>update-server-info</b>
cvsexportcommit	ls-files	relink	upload-archive
cvsimpport	ls-remote	<b>remote</b>	upload-pack
cvsserver	ls-tree	repack	var
daemon	mailinfo	repo-config	verify-pack
<b>describe</b>	mailsplit	request-pull	verify-tag
<b>diff</b>	<b>merge</b>	rerere	<b>what-changed</b>
diff-files	merge-base	<b>reset</b>	write-tree
diff-index	merge-file	rev-list	
diff-tree	merge-index	rev-parse	

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-



# Help

```
git help
```

- ▶ list of common commands

```
git <command> -h
```

- ▶ brief help output

```
man git-<command>
```

```
git help <command>
```

```
git <command> --help
```

- ▶ manual page

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

### Commands

Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Help

```
git help
```

- ▶ list of common commands

```
git <command> -h
```

- ▶ brief help output

```
man git-<command>
```

```
git help <command>
```

```
git <command> --help
```

- ▶ manual page

# Help

```
git help
```

- ▶ list of common commands

```
git <command> -h
```

- ▶ brief help output

```
man git-<command>
```

```
git help <command>
```

```
git <command> --help
```

- ▶ manual page

# Configuration

`$HOME/.gitconfig`

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email you@domain.tld
```

```
$ git config --global color.pager true
```

```
$ git config --global color.ui auto
```

# Configuration

```
$HOME/.gitconfig
```

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email you@domain.tld
```

```
$ git config --global color.pager true
```

```
$ git config --global color.ui auto
```

# Configuration

```
$HOME/.gitconfig
```

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email you@domain.tld
```

```
$ git config --global color.pager true
```

```
$ git config --global color.ui auto
```

# Configuration

```
$ cat .gitconfig
```

```
[user]
```

```
name = "Bart Trojanowski"  
email = "bart@jukie.net"  
signingkey = 2289688F
```

```
[core]
```

```
pager = less -FRSX  
editor = vim
```

```
[color]
```

```
ui = auto
```

```
[merge]
```

```
tool = vimdiff
```

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
**Commit**  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Bootstrapping

```
$ git init
```

- ▶ ran in project workspace
- ▶ creates `.git` directory

# Staging

## What to commit?

- ▶ additions

```
$ git add file  
$ git add .
```

- ▶ removal

```
$ git rm file
```

- ▶ renames

```
$ git mv old new
```

# Staging

## What to commit?

- ▶ additions

```
$ git add file
```

```
$ git add .
```

- ▶ removal

```
$ git rm file
```

- ▶ renames

```
$ git mv old new
```

# Staging

## What to commit?

- ▶ additions

```
$ git add file
```

```
$ git add .
```

- ▶ removal

```
$ git rm file
```

- ▶ renames

```
$ git mv old new
```

# Staging

What to commit?

▶ additions

```
$ git add file
```

```
$ git add .
```

▶ removal

```
$ git rm file
```

▶ renames

```
$ git mv old new
```

# Staging

```
$ cat .gitignore
```

```
*.O
```

```
*~
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

**Commit**

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

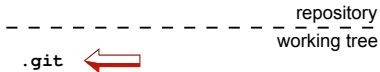
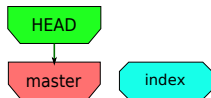
# Committing

```
$ git commit -a -m'some comment'
```

- ▶ will create a commit of all or only staged items

# Bootstrapping

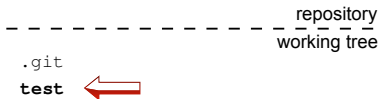
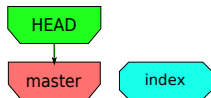
```
$ mkdir project  
$ cd project  
$ git init
```





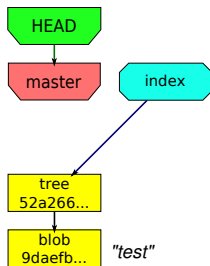
# First commit

```
$ echo test > test
```



# Stage

```
$ echo test > test  
$ git add test
```



.git  
test

repository  
working tree

# Commit

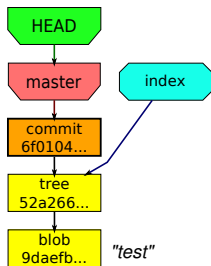
```
$ echo test > test
```

```
$ git add test
```

```
$ git commit -m'test'
```

```
Created initial commit 6f01040: test  
1 files changed, 1 insertions(+),  
0 deletions(-)
```

```
create mode 100644 test
```



.git  
test

repository  
working tree

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
**Commit**  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

## Stage another

```
$ echo test > test
```

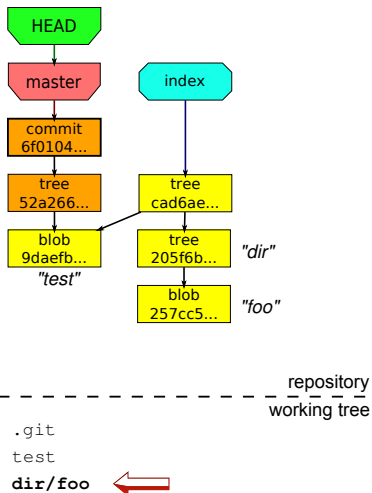
```
$ git add test
```

```
$ git commit -m"test"
```

```
$ mkdir dir
```

```
$ echo foo > dir/foo
```

```
$ git add dir/foo
```



## Commit another

```
$ echo test > test
```

```
$ git add test
```

```
$ git commit -m"test"
```

```
$ mkdir dir
```

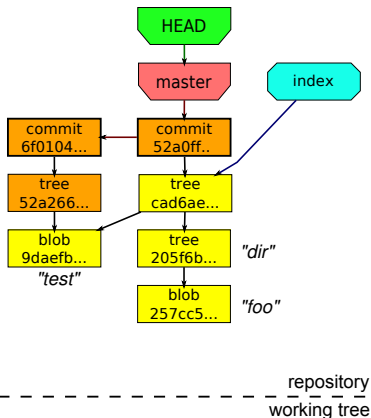
```
$ echo foo > dir/foo
```

```
$ git add dir/foo
```

```
$ git commit -m"foo"
```

```
Created commit 52a0ff4: foo
1 files changed, 1 insertions(+),
0 deletions(-)
```

```
create mode 100644 dir/foo
```



```

.git
test
dir/foo

```

```

repository
working tree

```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

- Commands
- Commit**
- Inspection
- Branching
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
**Inspection**  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Repository status

```
$ git status
```

shows...

- ▶ staged
- ▶ unstaged
- ▶ untracked

# Diffs

```
$ git diff
```

- ▶ changes between index and working files

```
$ git diff --staged
```

- ▶ changes between HEAD and index

```
$ git diff HEAD
```

- ▶ changes between HEAD and working files

```
$ git diff $commit $commit
```

- ▶ changes between two commits



# Diffs

```
$ git diff
```

- ▶ changes between index and working files

```
$ git diff --staged
```

- ▶ changes between HEAD and index

```
$ git diff HEAD
```

- ▶ changes between HEAD and working files

```
$ git diff $commit $commit
```

- ▶ changes between two commits

# Diffs

```
$ git diff
```

- ▶ changes between index and working files

```
$ git diff --staged
```

- ▶ changes between HEAD and index

```
$ git diff HEAD
```

- ▶ changes between HEAD and working files

```
$ git diff $commit $commit
```

- ▶ changes between two commits

# Diffs

```
$ git diff
```

- ▶ changes between index and working files

```
$ git diff --staged
```

- ▶ changes between HEAD and index

```
$ git diff HEAD
```

- ▶ changes between HEAD and working files

```
$ git diff $commit $commit
```

- ▶ changes between two commits

# Object references

specific commit ID...

full hash            6bb1270ffb60cbfef87266d2d4b4abe4218d9c68

short hash           6bb127

tag                    **v1.5.6.1**

local branch         **master**

remote branch       **origin/master**

by message          “:/some text”

checkout             HEAD

last fetch            FETCH\_HEAD

previous head        ORIG\_HEAD

...

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Object references

a commit before HEAD

HEAD^ == HEAD~1

Git the basics

Bart  
Trojanowski,  
bart@juki.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Object references

few commits before HEAD

HEAD^^^ == HEAD~3

Git the basics

Bart  
Trojanowski,  
bart@juki.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Object references

few commits before master

`master^^^ == master~3`

Git the basics

Bart  
Trojanowski,  
bart@juki.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Object references

what was it yesterday?

`@{yesterday}` == `HEAD@{yesterday}`

Git the basics

Bart  
Trojanowski,  
bart@juki.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-



# Object references

how about my-other-branch on June 1st?

```
my-other-branch@{June.1}
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Object references

master a few changes ago?

master@{3}

Git the basics

Bart  
Trojanowski,  
bart@juki.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Show me!

review last commit...

```
$ git show
```

```
commit 83b2d051814e884a8e264127ed47552a5dcf6c1d
```

```
Author: Bart Trojanowski <bart@jukie.net>
```

```
Date: Thu Jul 3 21:44:39 2008 -0400
```

```
changed one line
```

```
diff --git a/test b/test
```

```
index 808a2c4..99810fa 100644
```

```
--- a/test
```

```
+++ b/test
```

```
@@ -1,3 +1,3 @@
```

```
Some old text before the change.
```

```
-Some text for removal.
```

```
+Replacement line.
```

```
Some old text after the change.
```

# Show me!

just the stats...

```
$ git show --stat
```

```
commit 83b2d051814e884a8e264127ed47552a5dcf6c1d
```

```
Author: Bart Trojanowski <bart@jukie.net>
```

```
Date: Thu Jul 3 21:44:39 2008 -0400
```

```
changed one line
```

```
test | 2 +-
```

```
1 files changed, 1 insertions(+), 1 deletions(-)
```

# Show me!

SVN'esq status information...

```
$ git show --name-status
```

```
commit 3d3d2989b817af3fd4fa6d63f200113bd6c94bdb
```

```
Author: Bart Trojanowski <bart@jukie.net>
```

```
Date: Thu Jul 3 22:59:13 2008 -0400
```

```
something more interesting
```

```
A      sub/bar
```

```
D      sub/foo
```

```
M      test
```

# Show me!

review any other commit...

```
$ git show HEAD
$ git show HEAD^^^
$ git show master~10
$ git show master@{May.16}
...
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Show me!

show a file (or tree) in history...

```
$ git show HEAD:file
```

contents...

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Logs

See commit history...

```
$ git log
```

```
commit 3d3d2989b817af3fd4fa6d63f200113bd6c94bdb
Author: Bart Trojanowski <bart@jukie.net>
Date: Thu Jul 3 22:59:13 2008 -0400
```

most recent commit

```
commit 83b2d051814e884a8e264127ed47552a5dcf6c1d
Author: Bart Trojanowski <bart@jukie.net>
Date: Thu Jul 3 21:44:39 2008 -0400
```

second most recent

```
commit 1cc1b35a611c39f49842e2ca28d40886c1ae9b7c
Author: Bart Trojanowski <bart@jukie.net>
Date: Thu Jul 3 21:44:05 2008 -0400
```

middle commit

```
commit 411515f51a78d66a27a7d56ebe9f70dbd2bff008
Author: Bart Trojanowski <bart@jukie.net>
Date: Thu Jul 3 21:43:36 2008 -0400
```

second oldest

```
commit 52a0ff44aba8599f43a5d821c421af316cb73051
Author: Bart Trojanowski <bart@jukie.net>
Date: Mon Jun 30 21:44:55 2008 -0400
```

oldest commit

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-



# Logs

`git log` is awesome!

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

**Inspection**

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# Logs

limit by range...

```
$ git log tag..branch
```

```
$ git log HEAD~10..
```

```
$ git log branch1 branch2 ^common
```

```
$ git log -10
```

```
$ git log -10 master@{yesterday}
```

```
$ git log --since="May 1" --until="June 1"
```

# Logs

limit by commit attributes. . .

```
$ git log --author=fred
$ git log --committer=joe
$ git log --grep="commit.*message.*text"
```

# Logs

search for a change...

```
$ git log -S"some code change"
```

```
$ git log --pickaxe-regex -S"some.*code.*change"
```

# Logs

limit by changes to specific path...

```
$ git log -- some/file
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection**
- Branching
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Faster grep

another way to search...

```
$ git grep -e "pattern" -- some/file
```

```
$ git grep -e "pattern" branch -- some/file
```

# Faster grep

another way to search...

```
$ git grep -e "pattern" -- some/file
```

```
$ git grep -e "pattern" branch -- some/file
```

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# References

things that point to commits

lightweight

mutable

disposable

3 basic types

Git the basics

**Bart  
Trojanowski,  
bart@juki.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# References

things that point to commits

lightweight

mutable

disposable

3 basic types

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# References

things that point to commits

lightweight

mutable

disposable

3 basic types

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# References

things that point to commits

lightweight

mutable

disposable

3 basic types

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# References

things that point to commits

lightweight

mutable

disposable

3 basic types

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

# References

## local branches

- ▶ `$ git branch -l`  
branch1  
branch2  
\* master
- ▶ `.git/refs/heads/<branch>`

# References

tags

▶ `$ git tag -l`

tag1

tag2

tag3

▶ `.git/refs/tags/<tag>`

# References

## remote branches

▶ `$ git branch -r`

`fred/master`

`joe/master`

`joe/another-branch`

▶ `.git/refs/remotes/<remote>/<branch>`



# Creating branches

```
$ git branch name commit
```

- ▶ new branch “name” on HEAD or specified commit

# Switching branches

```
$ git checkout -f name
```

- ▶ checkout files from “name” branch
- ▶ optionally force overwriting changed files

# Create and switch

```
$ git checkout -b name commit
```

- ▶ checkout files from “name” branch

# Switching with changes

```
$ git checkout name
```

```
error: You have local changes to 'filename';  
cannot switch branches.
```

```
$ git checkout -m name
```

- ▶ merge outstanding diff onto branch “name”
- ▶ can result in **conflict**

# Switching with changes

```
$ git checkout name
```

```
error: You have local changes to 'filename';  
cannot switch branches.
```

```
$ git checkout -m name
```

- ▶ merge outstanding diff onto branch “name”
- ▶ can result in **conflict**

# Working on branches

start at some tree

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

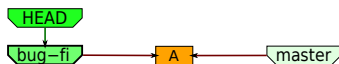
Loose ends

-



# Working on branches

```
$ git checkout -b bug-fix
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

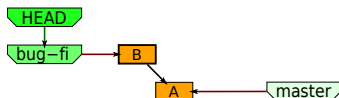
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git commit -a -m'B'
```



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

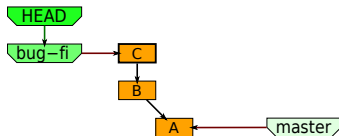
## Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Working on branches

```
$ git commit -a -m'C'
```



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

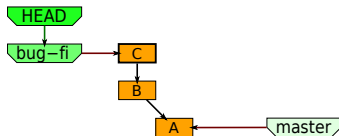
Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Working on branches

you have a *wicked* idea



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

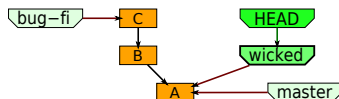
### Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Working on branches

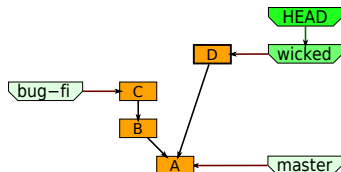
you have a *wicked* idea

```
$ git checkout -b wicked master
```



# Working on branches

```
$ git commit -a -m'D'
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

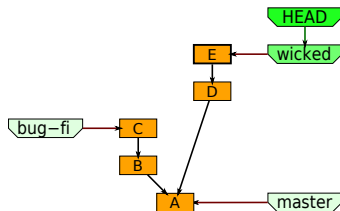
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git commit -a -m'E'
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

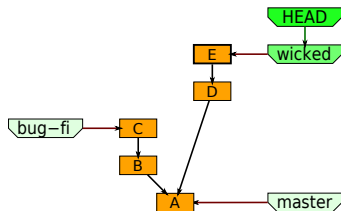
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

you're getting somewhere



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

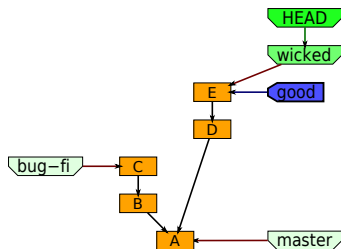
### Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Working on branches

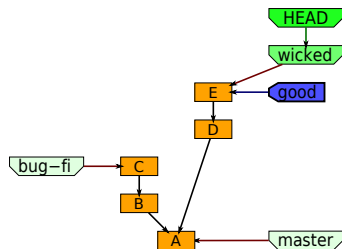
you're getting somewhere

```
$ git tag -a -m 'got somewhere' good
```



# Working on branches

manager asks about the bug



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

### Using GIT

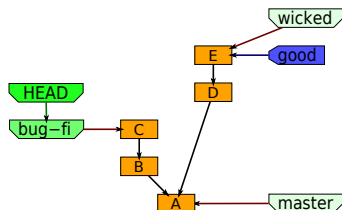
Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Working on branches

manager asks about the bug

```
$ git checkout bug-fix
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

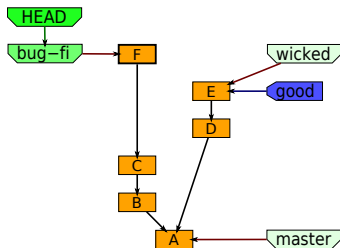
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git commit -a -m'F'
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

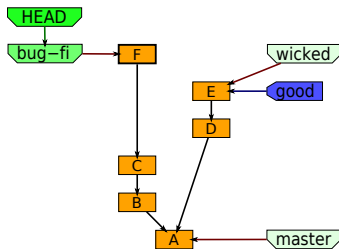
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

your mind is elsewhere...



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

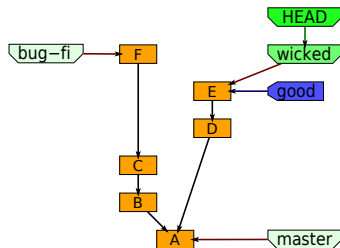
## Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Working on branches

your mind is elsewhere...

```
$ git checkout wicked
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

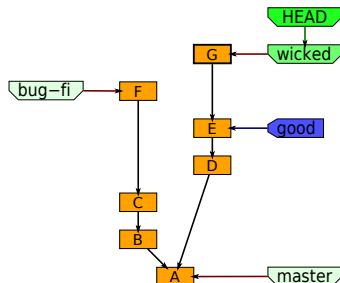
### Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

...so you finish off the *wicked* feature

```
$ git commit -a -m"G"
```

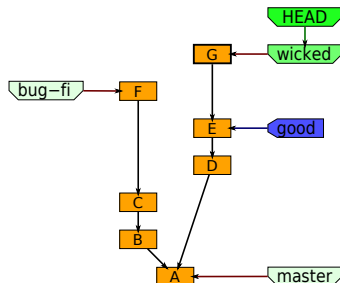


# Working on branches

feature's done

bug is fixed

...time to merge



Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

**Branching**

Merging

Rebasing

Remotes

GUI tools

Loose ends

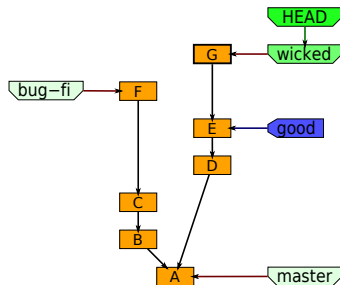
-

# Working on branches

feature's done

bug is fixed

...time to merge



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

### Using GIT

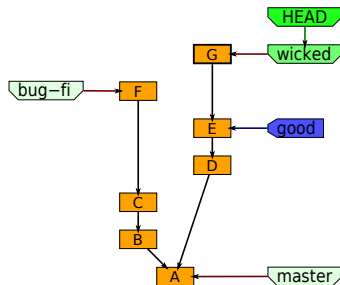
Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Working on branches

feature's done

bug is fixed

... time to merge



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

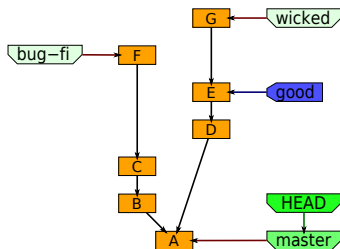
## Using GIT

Commands  
Commit  
Inspection  
**Branching**  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Working on branches

```
$ git checkout master
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

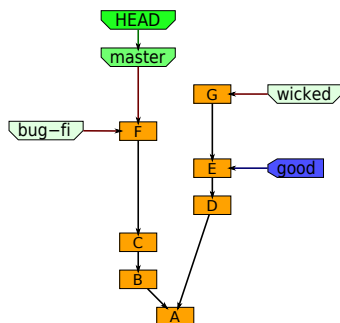
- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git checkout master  
$ git merge bug-fix
```

or

```
$ git checkout master  
$ git reset --hard bug-fix
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

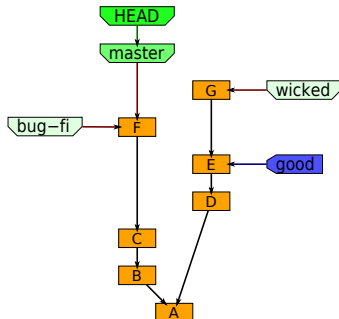
- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git checkout master  
$ git merge bug-fix
```

or

```
$ git checkout master  
$ git reset --hard bug-fix
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

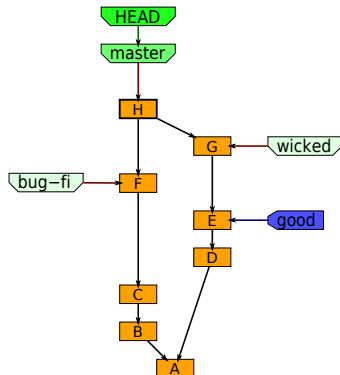
- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Working on branches

```
$ git merge wicked
```

...in case of conflict run

```
$ git mergetool
```

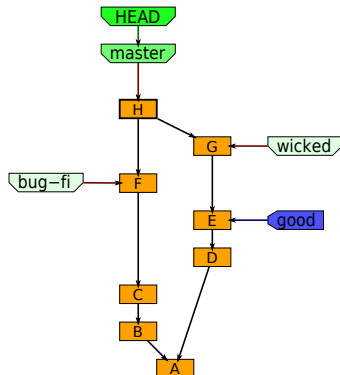


# Working on branches

```
$ git merge wicked
```

...in case of conflict run

```
$ git mergetool
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

- Commands
- Commit
- Inspection
- Branching**
- Merging
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
**Merging**  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Merging

```
$ git merge <branch> ...
```

- ▶ merge multiple branches
- ▶ creates commit with 2+ parents
- ▶ can cause **conflicts**  
ie: require user intervention

# Merging

one more example



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

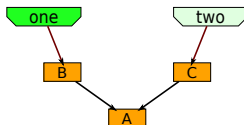
## Using GIT

Commands  
Commit  
Inspection  
Branching  
**Merging**  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



# Merging

two topic branches



## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

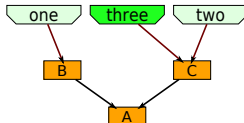
Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
**Merging**  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

# Merging

```
$ git checkout -b three two
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

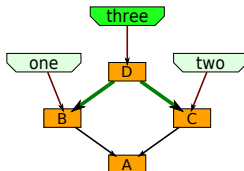
- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging**
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Merging

```
$ git checkout -b three two  
$ git merge one
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging**
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Merging

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

**Merging**

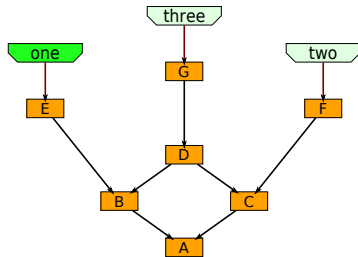
Rebasing

Remotes

GUI tools

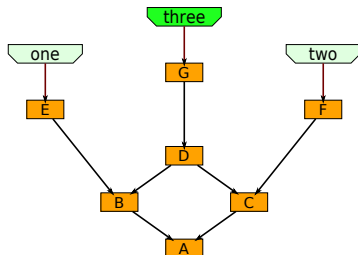
Loose ends

-



# Merging

```
$ git checkout three
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

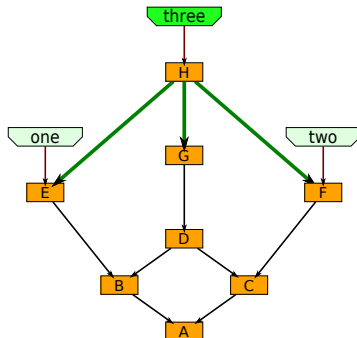
### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging**
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Merging

```
$ git checkout three
```

```
$ git merge one two
```



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

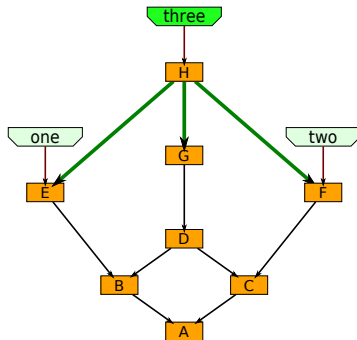
### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging**
- Rebasing
- Remotes
- GUI tools
- Loose ends
-

# Merging

```
$ git checkout three  
$ git merge one two
```

“octopus”



Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
**Rebasing**  
Remotes  
GUI tools  
Loose ends  
-



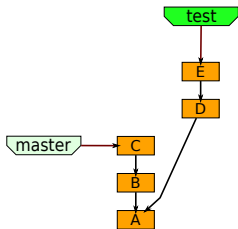
# Rebasing

```
$ git rebase <branch>
```

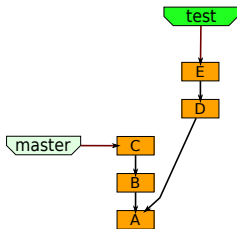
- ▶ moves new work onto a new baseline
- ▶ can cause **conflicts**  
ie: require user intervention

# Rebasing

merge



rebase



take two identical trees

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging

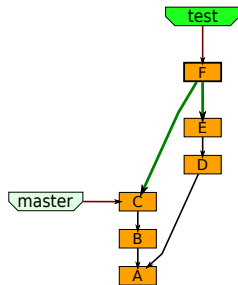
## Rebasing

- Remotes
- GUI tools
- Loose ends

-

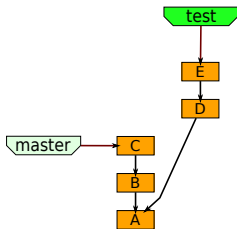
# Rebasing

merge



```
$ git merge master
```

rebase



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

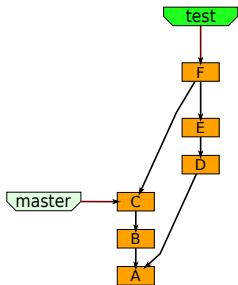
- Structure
- Objects

## Using GIT

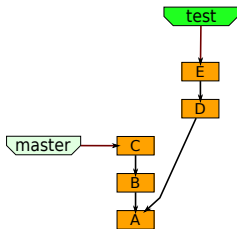
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing**
- Remotes
- GUI tools
- Loose ends
-

# Rebasing

merge



rebase



that's easy

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

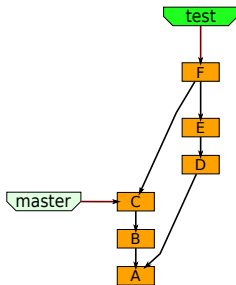
- Structure
- Objects

## Using GIT

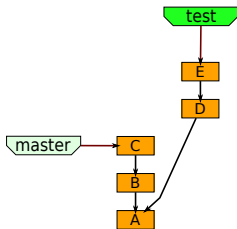
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing**
- Remotes
- GUI tools
- Loose ends
-

# Rebasing

merge



rebase



```
$ git rebase master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

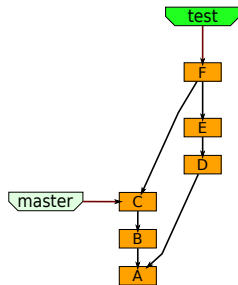
- Structure
- Objects

## Using GIT

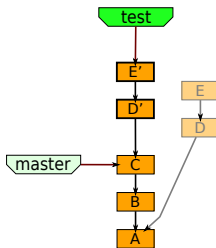
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing**
- Remotes
- GUI tools
- Loose ends
-

# Rebasing

merge



rebase



```
$ git rebase master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

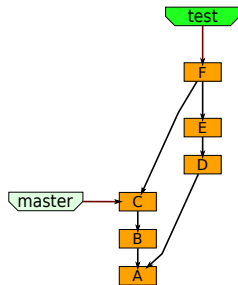
- Commands
- Commit
- Inspection
- Branching
- Merging

## Rebasing

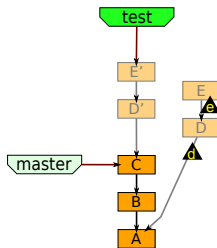
- Remotes
- GUI tools
- Loose ends

# Rebasing

merge



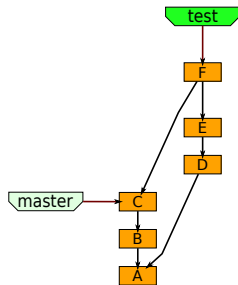
rebase



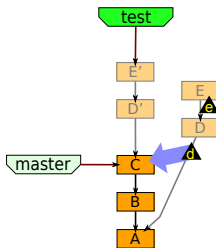
```
$ git rebase master
```

# Rebasing

merge



rebase



```
$ git rebase master
```

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging

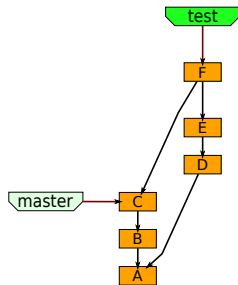
**Rebasing**

Remotes  
GUI tools  
Loose ends  
-

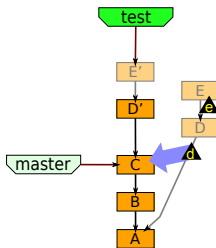


# Rebasing

merge



rebase



```
$ git rebase master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging

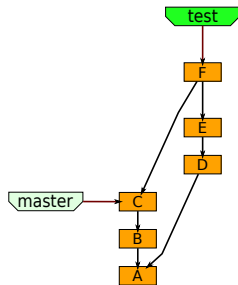
## Rebasing

- Remotes
- GUI tools
- Loose ends

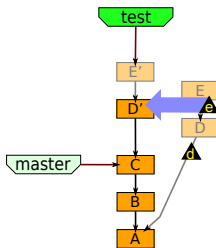
-

# Rebasing

merge



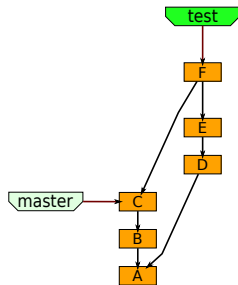
rebase



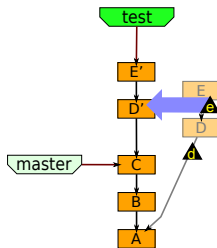
```
$ git rebase master
```

# Rebasing

merge



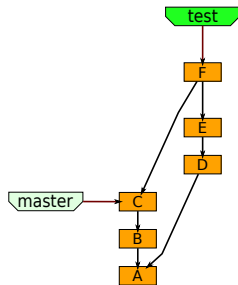
rebase



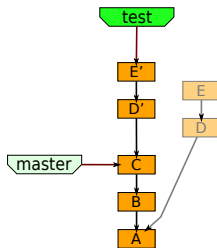
```
$ git rebase master
```

# Rebasing

merge



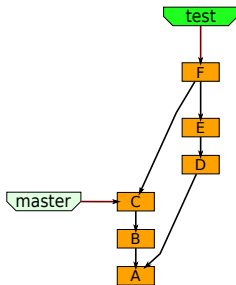
rebase



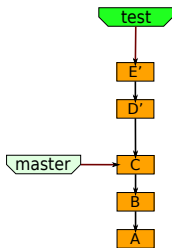
```
$ git rebase master
```

# Rebasing

merge



rebase



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing**
- Remotes
- GUI tools
- Loose ends
-

## Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

### Concepts

SCM components  
SCM operations  
Decentralization

### GIT History

### Repository

Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
**Remotes**  
GUI tools  
Loose ends  
-

# Cloning

```
$ git clone <remote>
```

- ▶ replicates `remote` repository
- ▶ populates new repository
- ▶ checkout new working tree

# Git URLs

- ▶ local repo

```
/home/git/project.git/  
file:///home/git/project.git/
```

- ▶ http protocol

```
http://repo.or.cz/r/git.git
```

- ▶ native git protocol

```
git://repo.or.cz/git.git
```

- ▶ ssh protocol

```
ssh://bart@jukie.net/~git/project.git/  
bart@jukie.net/~git/project.git/
```



# Git URLs

- ▶ local repo

```
/home/git/project.git/  
file:///home/git/project.git/
```

- ▶ http protocol

```
http://repo.or.cz/r/git.git
```

- ▶ native git protocol

```
git://repo.or.cz/git.git
```

- ▶ ssh protocol

```
ssh://bart@jukie.net/~git/project.git/  
bart@jukie.net/~git/project.git/
```

# Git URLs

- ▶ local repo

```
/home/git/project.git/  
file:///home/git/project.git/
```

- ▶ http protocol

```
http://repo.or.cz/r/git.git
```

- ▶ native git protocol

```
git://repo.or.cz/git.git
```

- ▶ ssh protocol

```
ssh://bart@jukie.net/~git/project.git/  
bart@jukie.net/~git/project.git/
```

# Git URLs

- ▶ local repo

```
/home/git/project.git/  
file:///home/git/project.git/
```

- ▶ http protocol

```
http://repo.or.cz/r/git.git
```

- ▶ native git protocol

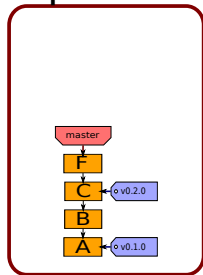
```
git://repo.or.cz/git.git
```

- ▶ ssh protocol

```
ssh://bart@jukie.net/~git/project.git/  
bart@jukie.net/~git/project.git/
```

# Working with remotes

repo.or.cz



```
git://repo.or.cz/project.git
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

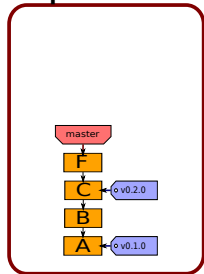
### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing
- Remotes**
- GUI tools
- Loose ends

-

# Working with remotes

repo.or.cz



```
$ git clone ssh+git://repo.or.cz/project.git
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

**Remotes**

GUI tools

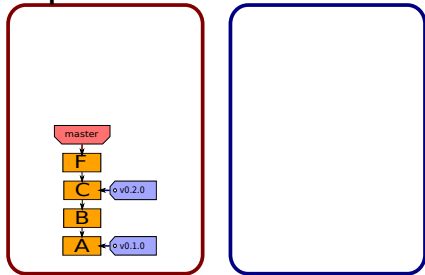
Loose ends

-

# Working with remotes

repo.or.cz

local

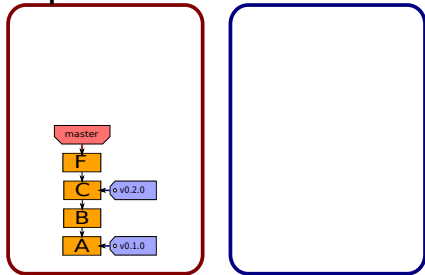


```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

# Working with remotes

repo.or.cz

local



```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

**Remotes**

GUI tools

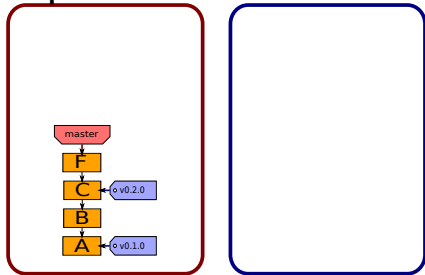
Loose ends

-

# Working with remotes

repo.or.cz

local



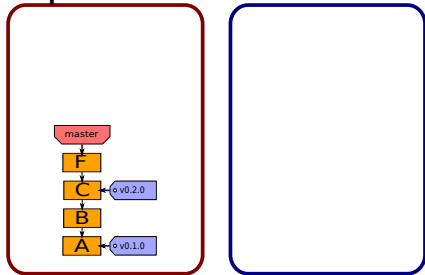
```
$ git clone ssh+git://repo.or.cz/project.git build-tree
Initialize build-tree/.git
Initialized empty Git repository in /tmp/build-tree/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```



# Working with remotes

repo.or.cz

local



```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

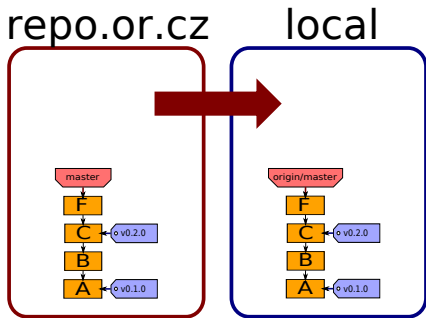
**Remotes**

GUI tools

Loose ends

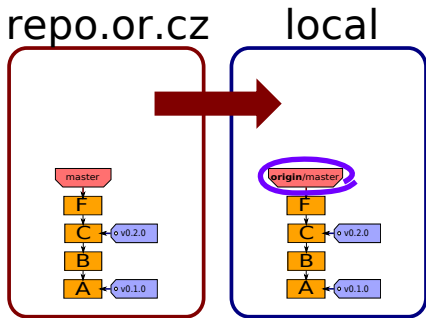
-

# Working with remotes



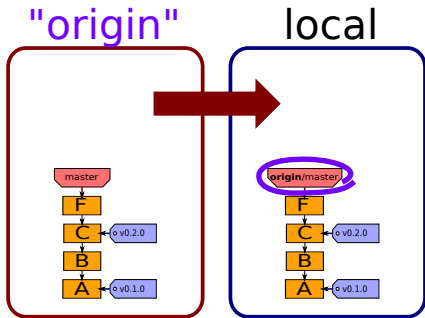
```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

# Working with remotes



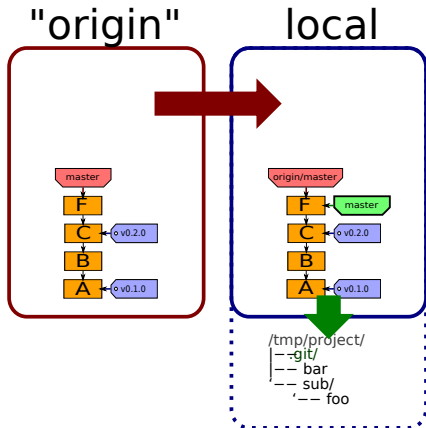
```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

# Working with remotes



```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

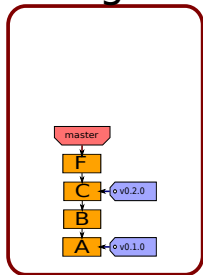
# Working with remotes



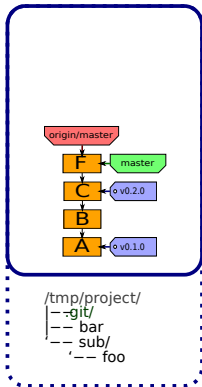
```
$ git clone ssh+git://repo.or.cz/project.git
Initialize project/.git
Initialized empty Git repository in /tmp/project/.git/
remote: Counting objects: 77575, done.
remote: Compressing objects: 100% (26407/26407), done.
remote: Total 77575 (delta 55750), reused 71007 (delta 49775)
Receiving objects: 100% (77575/77575), 22.87 MiB | 798 KiB/s, done.
Resolving deltas: 100% (55750/55750), done.
Checking out files: 100% (1396/1396), done.
```

# Working with remotes

## "origin"



## local



```
$ git branch -a  
* master  
  origin/master
```

```
$ git tag -l  
v0.1.0  
v0.2.0
```

### Concepts

- SCM components
- SCM operations
- Decentralization

### GIT History

### Repository

- Structure
- Objects

### Using GIT

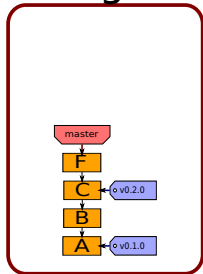
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

### Remotes

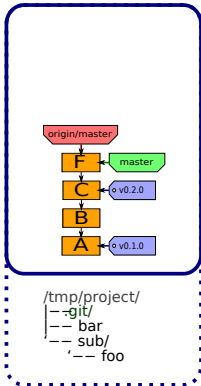
- GUI tools
- Loose ends
-

# Working with remotes

## "origin"



## local

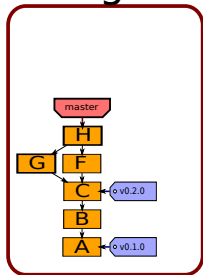


```
$ git branch -a  
* master  
origin/master
```

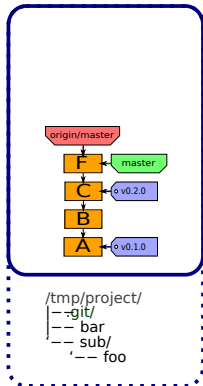
```
$ git tag -l  
v0.1.0  
v0.2.0
```

# Working with remotes

## "origin"



## local



### Concepts

- SCM components
- SCM operations
- Decentralization

### GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

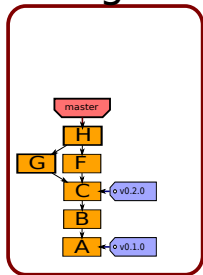
### Remotes

- GUI tools
- Loose ends
-

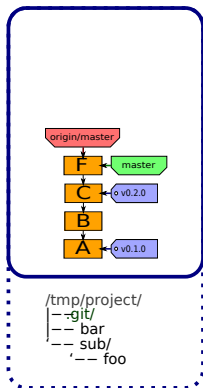


# Working with remotes

"origin"

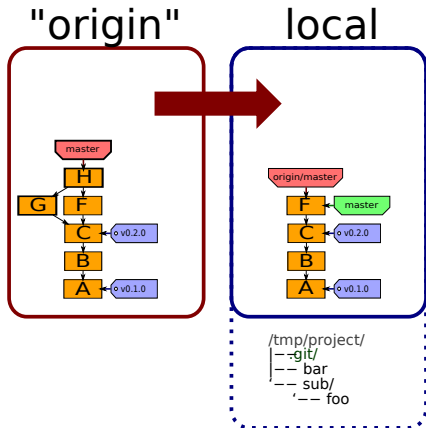


local



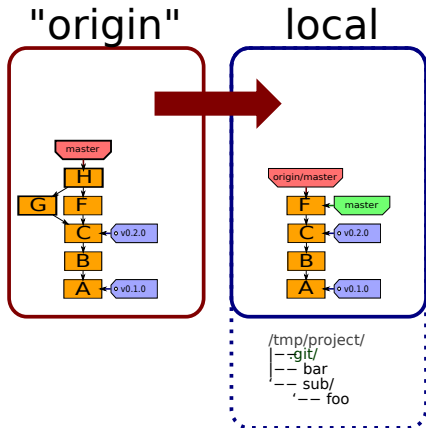
```
$ git fetch
```

# Working with remotes



```
$ git fetch
remote: Counting objects: 236, done.
remote: Compressing objects: 100% (190/190), done.
remote: Total 190 (delta 170), reused 0 (delta 0)
Receiving objects: 100% (190/190), 69.53 KiB, done.
Resolving deltas: 100% (170/170), completed with 40 local objects.
From mail.jukie.net:work/oclug/intro-to-git
 573ff80..06e3703  master    -> origin/master
```

# Working with remotes



```
$ git fetch
```

```
remote: Counting objects: 236, done.
```

```
remote: Compressing objects: 100% (190/190), done.
```

```
remote: Total 190 (delta 170), reused 0 (delta 0)
```

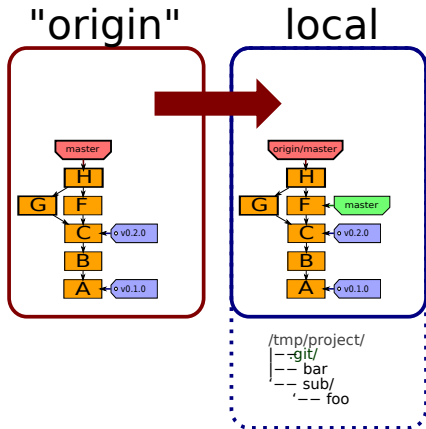
```
Receiving objects: 100% (190/190), 69.53 KiB, done.
```

```
Resolving deltas: 100% (170/170), completed with 40 local objects.
```

```
From mail.jukie.net:work/oclug/intro-to-git
```

```
573ff80..06e3703  master    -> origin/master
```

# Working with remotes



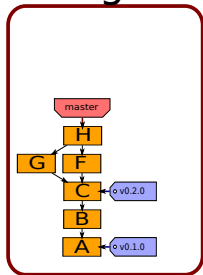
```
$ git fetch
```

```
remote: Counting objects: 236, done.  
remote: Compressing objects: 100% (190/190), done.  
remote: Total 190 (delta 170), reused 0 (delta 0)  
Receiving objects: 100% (190/190), 69.53 KiB, done.  
Resolving deltas: 100% (170/170), completed with 40 local objects.
```

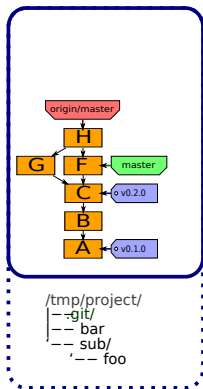
```
From mail.jukie.net:work/oclug/intro-to-git  
573ff80..06e3703 master -> origin/master
```

# Working with remotes

## "origin"

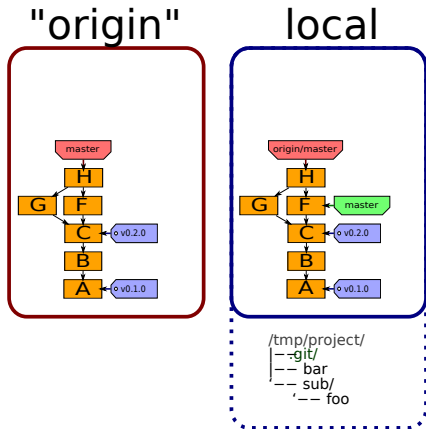


## local



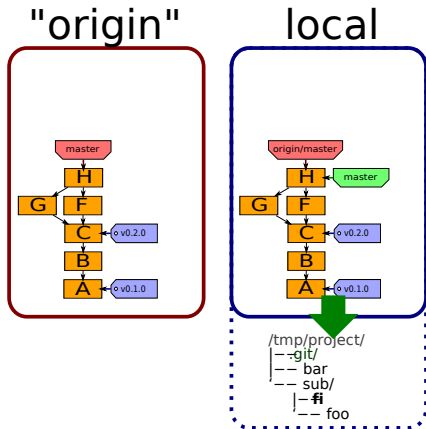
... no disk changes?

# Working with remotes



`git fetch` only updates DAG

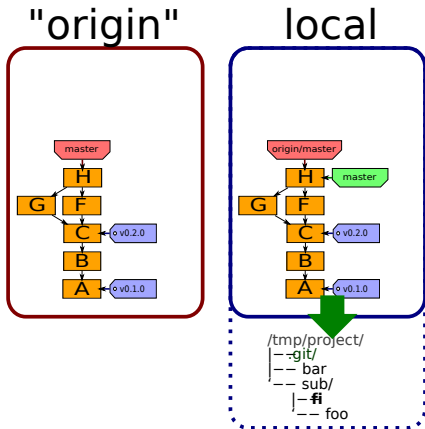
# Working with remotes



```
git merge origin/master
```

*fast-forwards* master to match *origin/master*  
and updates the working tree

# Working with remotes



```
git merge origin/master
```

*fast-forwards* master to match **origin/master**  
and updates the working tree

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

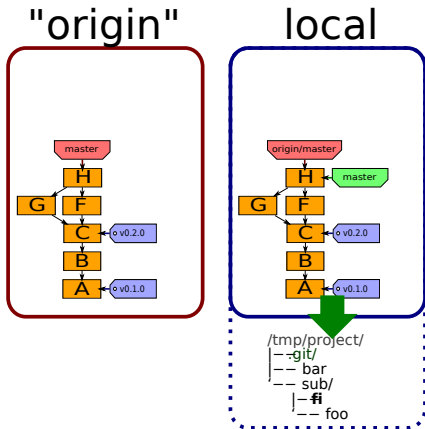
Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
**Remotes**  
GUI tools  
Loose ends  
-



# Working with remotes



```
git merge origin/master
fast-forwards master to match origin/master
and updates the working tree
```

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

GUI tools  
Loose ends

-

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

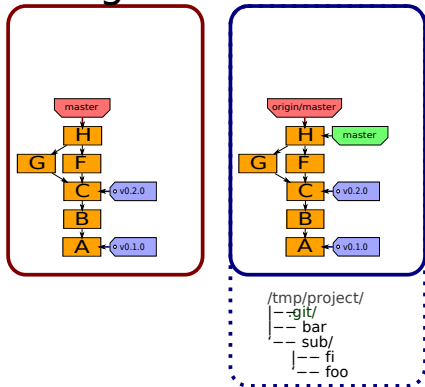
## Remotes

GUI tools  
Loose ends

-

"origin"

local



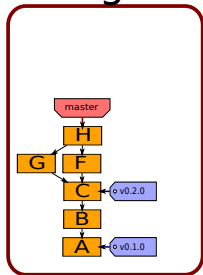
`git fetch + git merge`

=

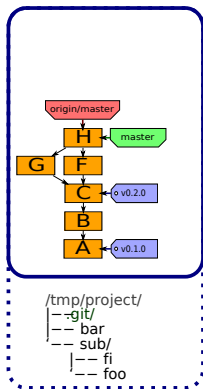
`git pull`

# Working with remotes

"origin"



local



`git fetch + git merge`  
=  
`git pull`

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

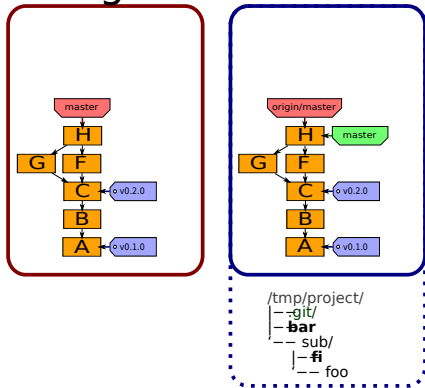
Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
**Remotes**  
GUI tools  
Loose ends  
-

"origin"

local



let's make some local commits

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

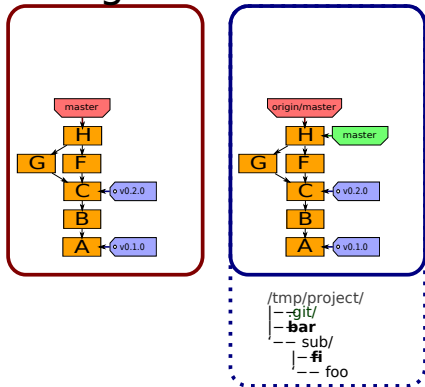
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

**Remotes**  
GUI tools  
Loose ends  
-

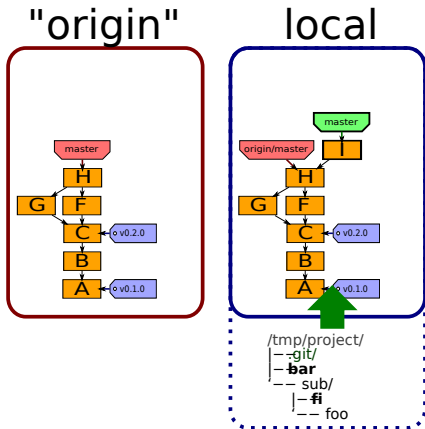
"origin"

local



```
$ git commit -a -m 'I'
```

# Working with remotes



```
$ git commit -a -m('I')
Created commit b618aed: I
 2 files changed, 11 insertions(+), 20 deletions(-)
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

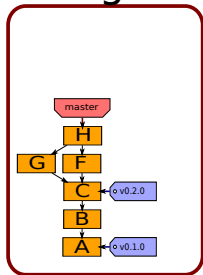
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

### Remotes

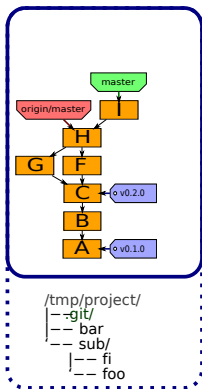
- GUI tools
- Loose ends
-

# Working with remotes

## "origin"



## local



Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

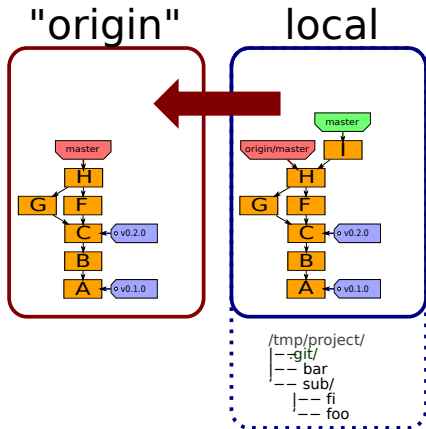
**Remotes**

GUI tools

Loose ends

-

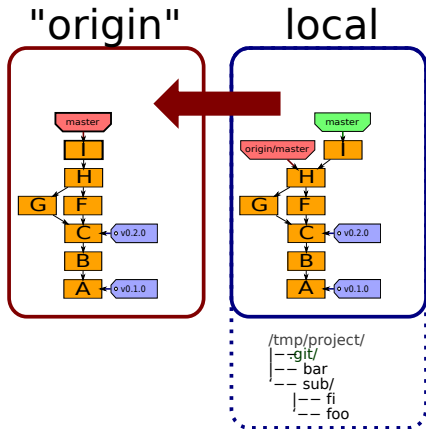
# Working with remotes



```
$ git push
```



# Working with remotes



```
$ git push
```

```
Counting objects: 9, done.
```

```
Compressing objects: 100% (5/5), done.
```

```
Writing objects: 100% (5/5), 810 bytes, done.
```

```
Total 5 (delta 4), reused 0 (delta 0)
```

```
refs/heads/master: 9ddc135 -> 15b67c0
```

```
To ssh+git://repo.or.cz/project.git
```

```
9ddc135..15b67c0 master -> master
```

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

## Repository

Structure

Objects

## Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

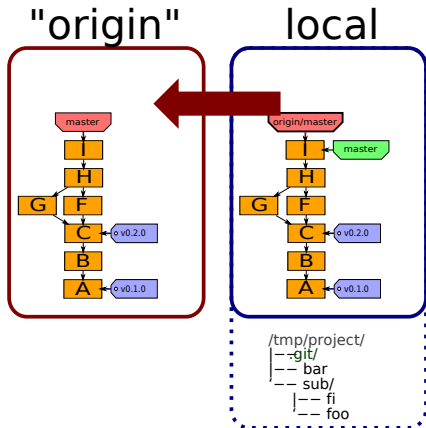
**Remotes**

GUI tools

Loose ends

-

# Working with remotes



```
$ git push
```

```
Counting objects: 9, done.
```

```
Compressing objects: 100% (5/5), done.
```

```
Writing objects: 100% (5/5), 810 bytes, done.
```

```
Total 5 (delta 4), reused 0 (delta 0)
```

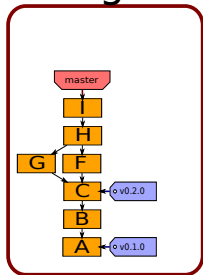
```
refs/heads/master: 9ddc135 -> 15b67c0
```

```
To ssh+git://repo.or.cz/project.git
```

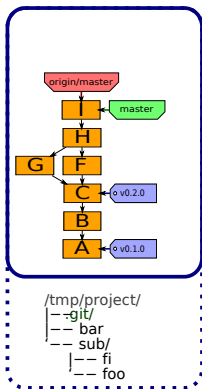
```
9ddc135..15b67c0 master -> master
```

# Working with remotes

## "origin"



## local



### Concepts

- SCM components
- SCM operations
- Decentralization

### GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

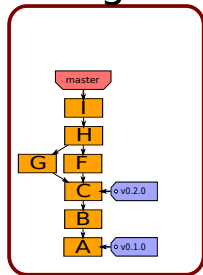
### Remotes

- GUI tools
- Loose ends

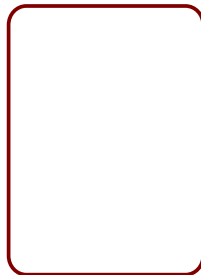
-

# Working with remotes

"origin"

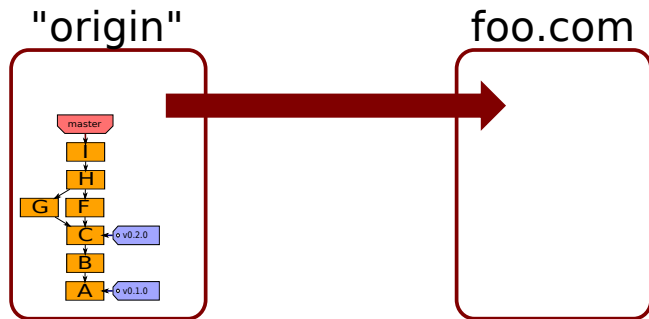


foo.com



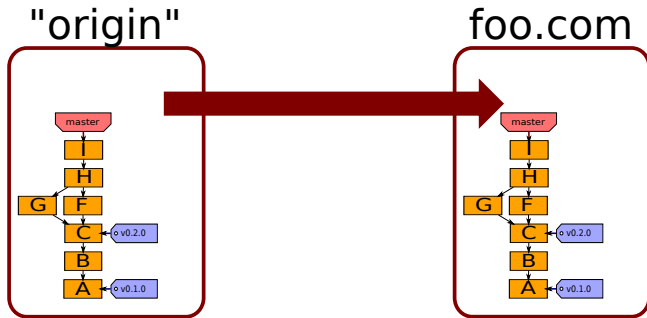
... meanwhile, elsewhere on the internet ...

# Working with remotes



Fred clones the project.

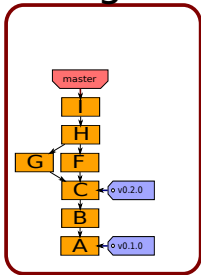
# Working with remotes



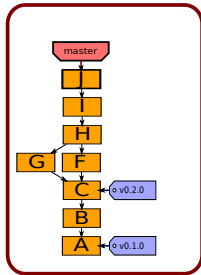
Fred clones the project.

# Working with remotes

"origin"



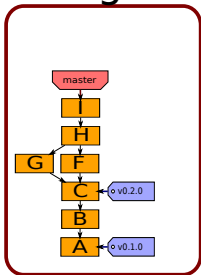
foo.com



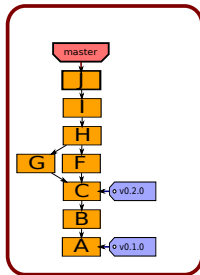
... and makes some changes.

# Working with remotes

"origin"



foo.com

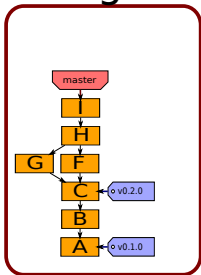


Fred cannot push.

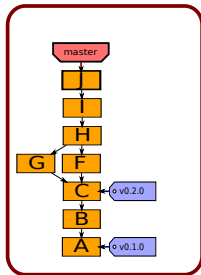


# Working with remotes

## "origin"



## foo.com



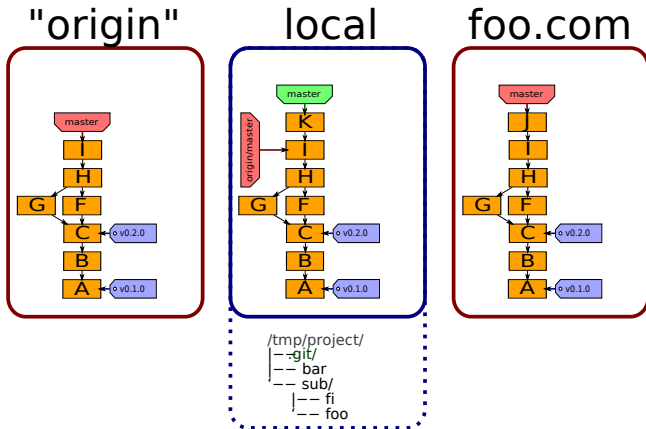
```
From: fred@foo.com
Subject: I fixed a bug
```

Please pull from

<http://foo.com/project.git/>

-Fred

# Working with remotes



```
From: fred@foo.com
Subject: I fixed a bug
```

Please pull from

```
http://foo.com/project.git/
```

-Fred

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

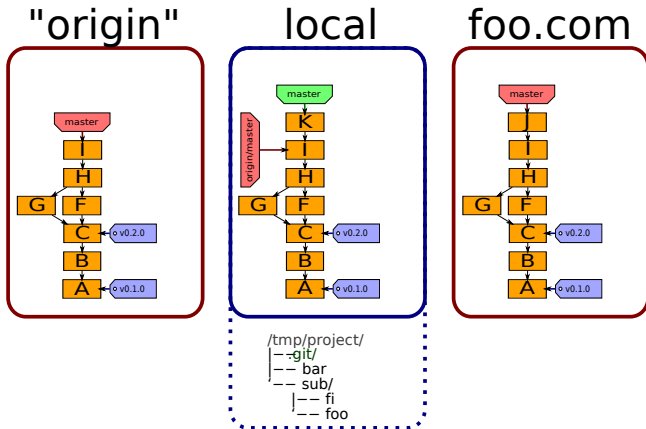
### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

### Remotes

- GUI tools
- Loose ends

# Working with remotes



```
$ git remote add fred http://foo.com/project.git/
```

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

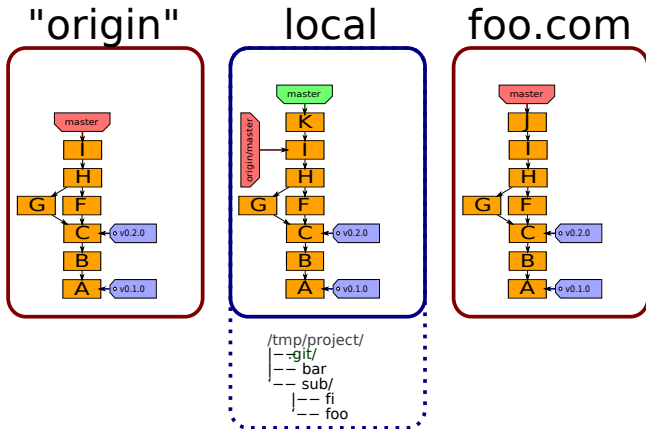
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

GUI tools  
Loose ends  
-

# Working with remotes



```
$ git remote add fred http://foo.com/project.git/
```

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

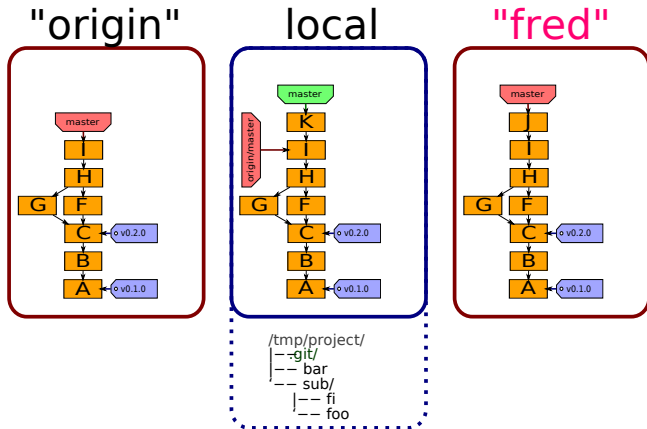
Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

GUI tools  
Loose ends

-

# Working with remotes



```
$ git remote add fred http://foo.com/project.git/
```

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

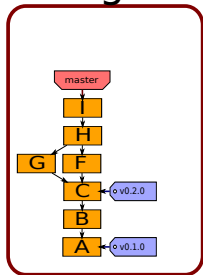
Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

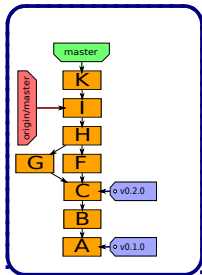
GUI tools  
Loose ends

-

"origin"

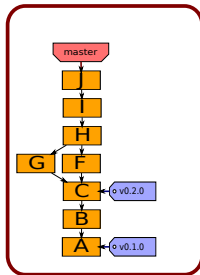


local



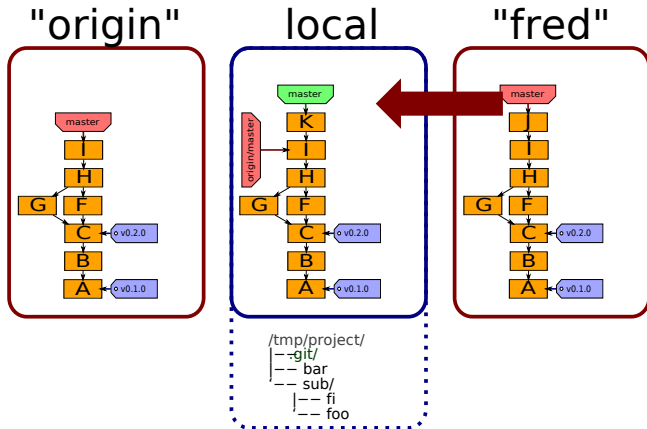
```
/tmp/project/  
|--git/  
|--bar  
|--sub/  
|  |--fi  
|  |--foo
```

"fred"



```
$ git fetch fred
```

# Working with remotes



```
$ git fetch fred
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

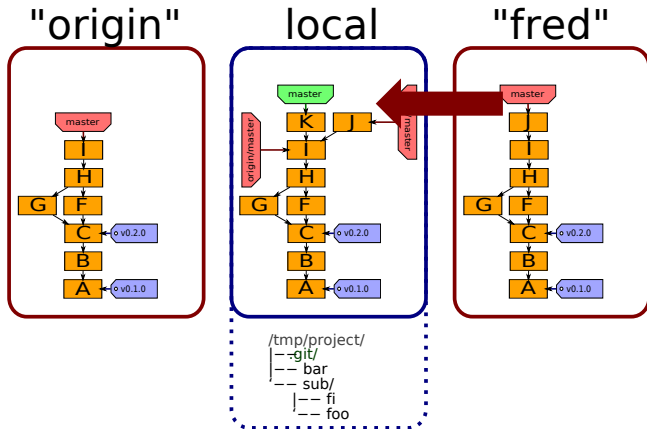
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git fetch fred
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

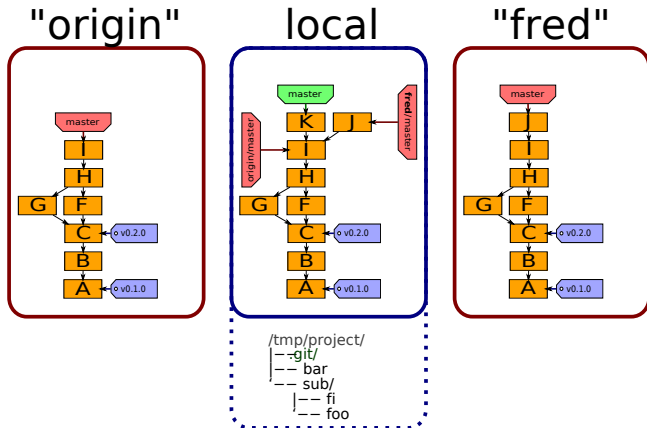
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends
-



# Working with remotes



## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

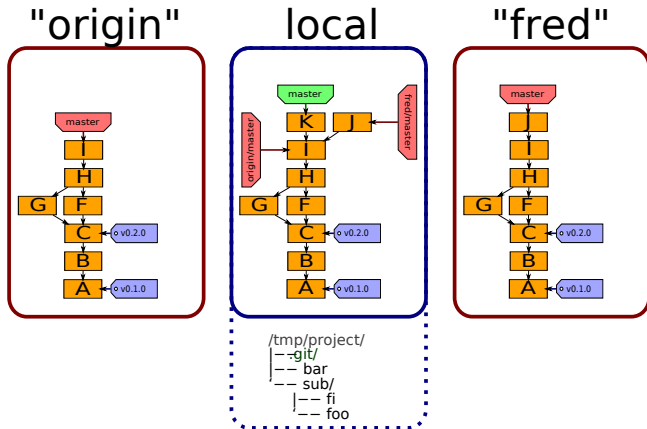
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git log fred/master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

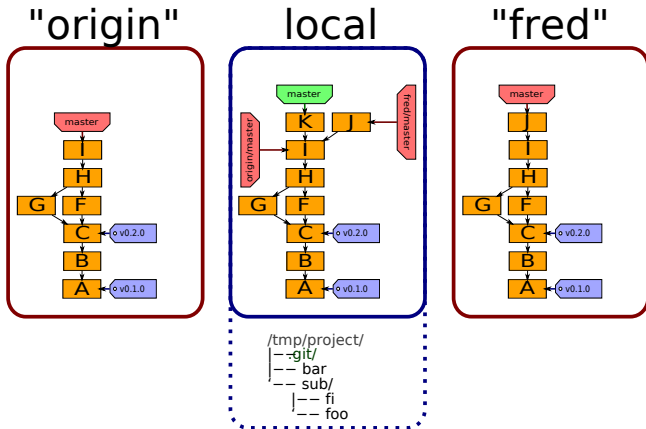
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

### Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git log fred/master  
$ git log fred/master ^master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

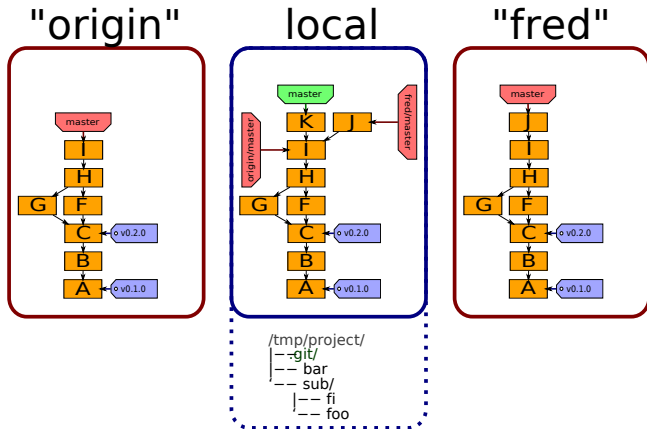
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

### Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git log fred/master  
$ git log fred/master ^master  
$ git log -p fred/master ^master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

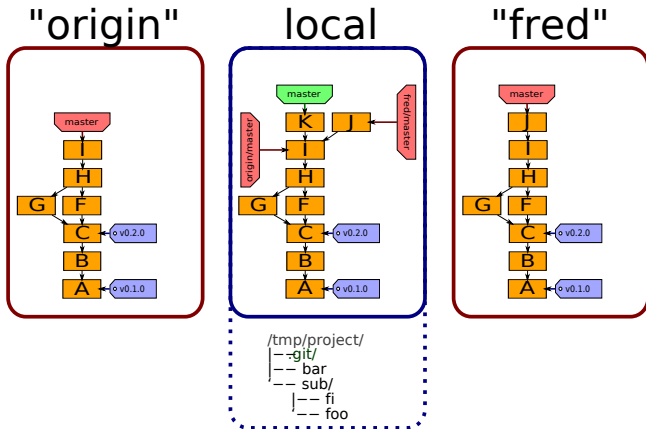
## Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

# Working with remotes



```
$ git checkout -b fred-fix fred/master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

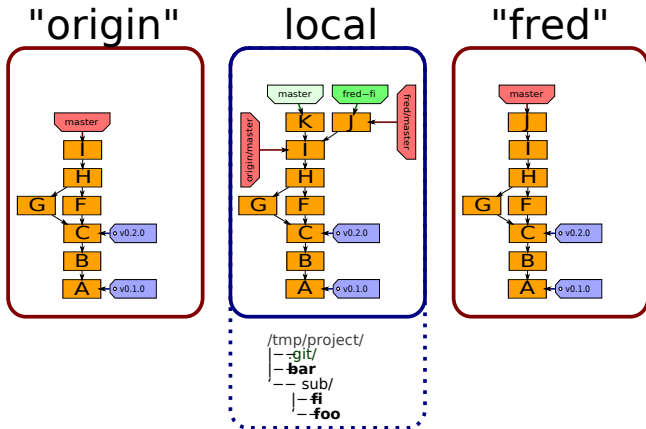
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git checkout -b fred-fix fred/master
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

### Repository

- Structure
- Objects

### Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing
- Remotes**
- GUI tools
- Loose ends
-

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

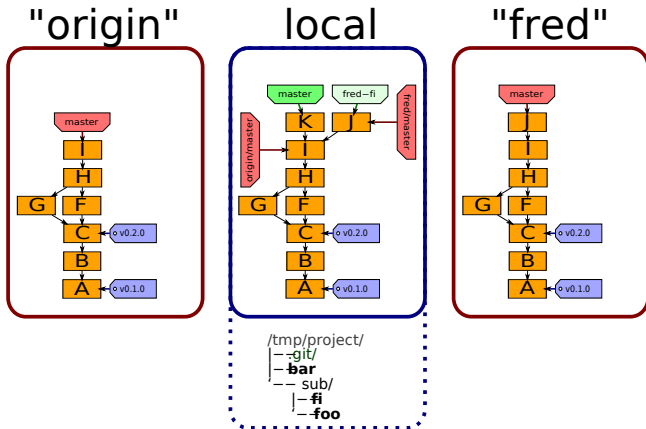
## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
**Remotes**  
GUI tools  
Loose ends  
-



```
$ git checkout master
```

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

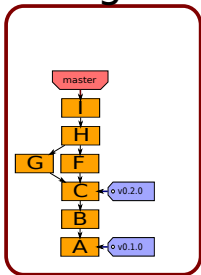
### Repository

Structure  
Objects

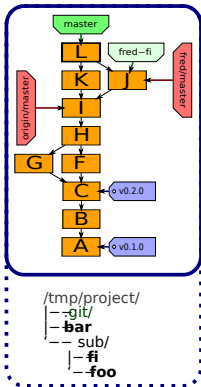
### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
**Remotes**  
GUI tools  
Loose ends  
-

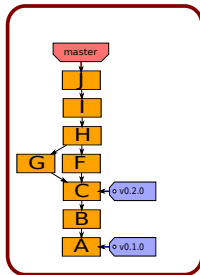
"origin"



local



"fred"



```
$ git checkout master
```

```
$ git merge fred-fix
```



# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

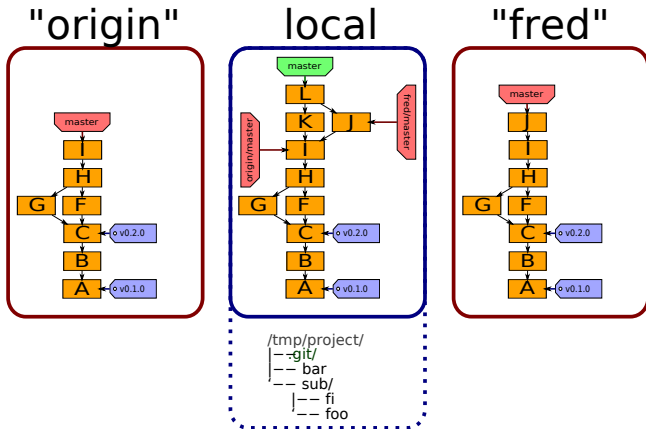
## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

GUI tools  
Loose ends

-

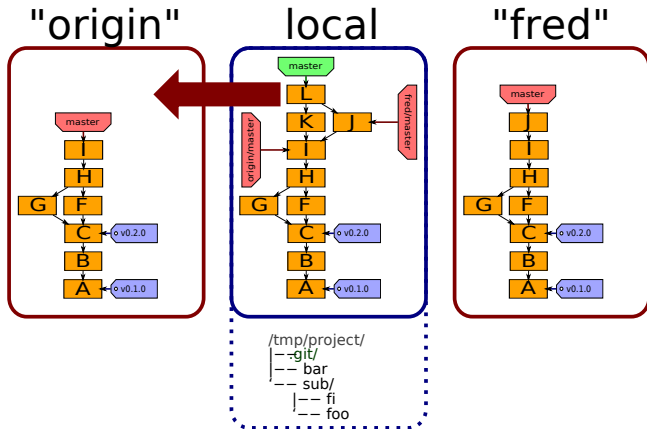


```
$ git checkout master
```

```
$ git merge fred-fix
```

```
$ git branch -d fred-fix
```

# Working with remotes



```
$ git push
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

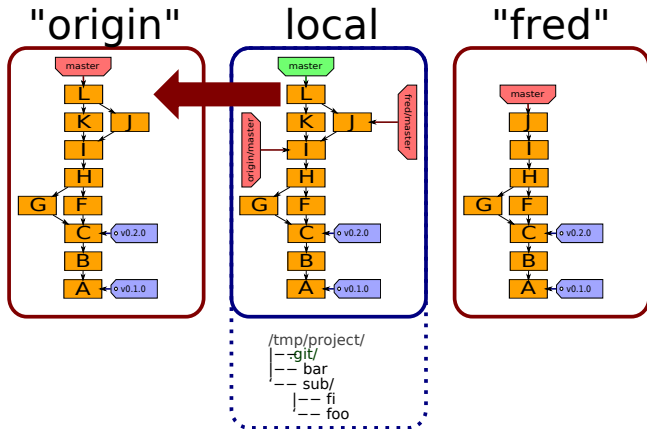
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git push
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

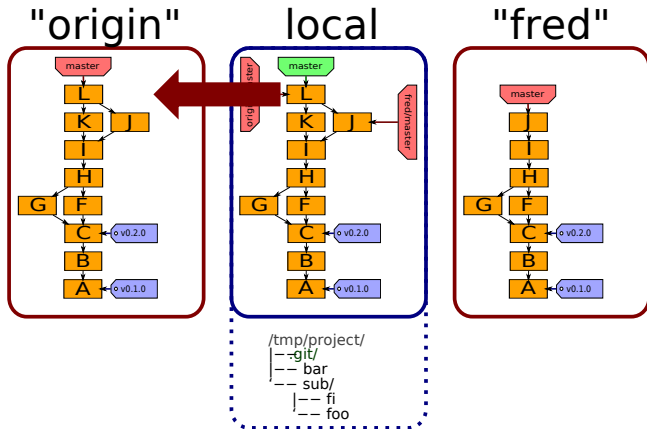
- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes



```
$ git push
```

## Concepts

- SCM components
- SCM operations
- Decentralization

## GIT History

## Repository

- Structure
- Objects

## Using GIT

- Commands
- Commit
- Inspection
- Branching
- Merging
- Rebasing

## Remotes

- GUI tools
- Loose ends

-

# Working with remotes

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

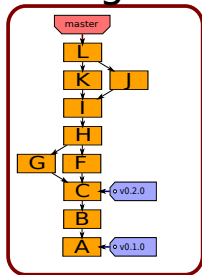
Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing

## Remotes

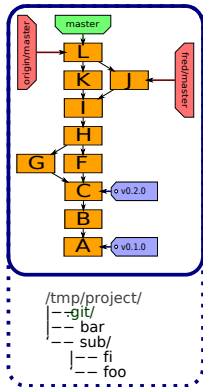
GUI tools  
Loose ends

-

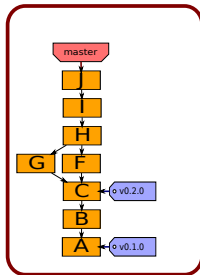
"origin"



local



"fred"



**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes

**GUI tools**

Loose ends

-

# GUI tools

- ▶ `gitk`  
sexy view of your revision tree
- ▶ `git gui`  
perform trivial tasks from a GUI  
ex: add/rm files, make commits, branch, etc

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes

**GUI tools**

Loose ends

-

# GUI tools

- ▶ `gitk`  
sexy view of your revision tree
- ▶ `git gui`  
perform trivial tasks from a GUI  
ex: add/rm files, make commits, branch, etc



how sexy?

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes

**GUI tools**

Loose ends

-

Bart  
Trojanowski,  
bart@jukie.net

File Edit View
Help

- let git-case-assign assign to no one
- load git-case-common from the location of the exec (closes 3ed4c7)
- arguments for git-case-new
- use gen\_short\_case\_id() to shorten case ID's
- Documentation/comment-format-proposal
- git-case-assign cleanup
- git-case-merge will now merge (closes 0dc37c)
- Merge commit 'origin/aidan/master'
- git-case-assign: Extend it to allow assigning to others
- Simplify add\_file\_to\_index
- possible fix for git-case-merge
- Work around problem with ' in second argument to add\_file\_to\_index.
- Merge branch 'master' of git+ssh://dave@tachyon.jukie.net/home/git/g
- oops, git-case-edit-description was storing the comment in the descri
- git-case-edit-description
- Allow git 1.5.3, as it's in etch-backports
- git-case-merge is broken, see case 0dc37c

Bart Trojanowski <bart@jukie.net>	2008-04-16 16:04:24
Bart Trojanowski <bart@jukie.net>	2008-04-16 15:41:56
Bart Trojanowski <bart@jukie.net>	2008-04-16 11:44:51
Bart Trojanowski <bart@jukie.net>	2008-04-16 11:21:38
Bart Trojanowski <bart@jukie.net>	2008-04-15 23:07:42
Bart Trojanowski <bart@jukie.net>	2008-04-15 20:26:18
Bart Trojanowski <bart@jukie.net>	2008-04-15 17:25:12
Bart Trojanowski <bart@jukie.net>	2008-04-15 13:06:20
Aidan Van Dyk <aidan@highrise.ca>	2008-04-15 11:36:51
Aidan Van Dyk <aidan@highrise.ca>	2008-04-15 11:34:52
Bart Trojanowski <bart@jukie.net>	2008-04-15 11:30:11
Dave O'Neill <dmo@dmo.ca>	2008-04-15 10:02:13
Dave O'Neill <dmo@dmo.ca>	2008-04-15 10:00:12
Bart Trojanowski <bart@jukie.net>	2008-04-15 09:47:21
Bart Trojanowski <bart@jukie.net>	2008-04-15 09:41:35
Dave O'Neill <dmo@dmo.ca>	2008-04-15 10:00:00
Bart Trojanowski <bart@jukie.net>	2008-04-15 00:17:20

SHA1 ID: 563bda419b5dd3d4ddab14e63ff0ec9588e24119 ← → Row 17 / 97

Find next prev commit containing: - Exact All fields -

Search
• Patch • Tree

• Diff • Old version • New version
Lines of context: 3 spa

```

Author: Bart Trojanowski <bart@jukie.net> 2008-04-16 15:41:56
Committer: Bart Trojanowski <bart@jukie.net> 2008-04-16 15:41:56
Parent: 3d7686c39cf8ec8a19ceb722e4f95c0f4f5e0f92 (arguments for git-case-
Child: c893e62db0bb8babad42eac40f4b50ef1cd82e08 (let git-case-assign ass
Branches: master, remotes/origin/aidan/master, remotes/origin/master
Follows:
Precedes:

    load git-case-common from the location of the exec (closes 3ed4c7)

----- git-case-active -----
index 5cbc782..bd5b17d 100755
@@ -5,7 +5,7 @@
# - we can access git-case-common
# - the branch defined by GIT_CASE_HEAD exists (see git-case-common)

-source git-case-common
+source $(dirname $0)/git-case-common

ID=$1
if test -z "$ID" ; then

----- git-case-assign -----

```

## Concepts

SCM components

SCM operations

Decentralization

## GIT History

## Repository

Structure

Objects

## Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

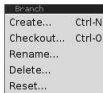
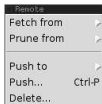
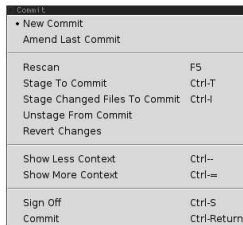
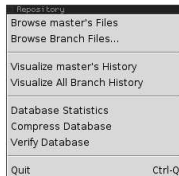
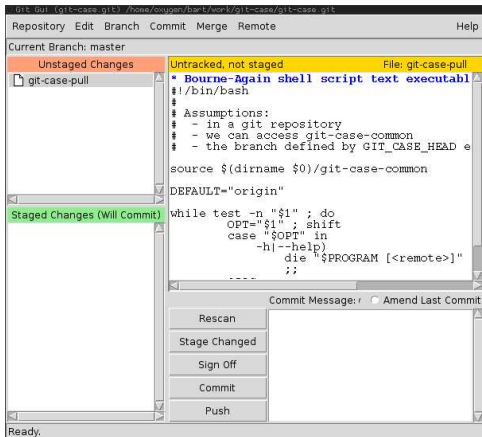
GUI tools

Loose ends

# git-gui

Git the basics

Bart  
Trojanowski,  
bart@jukie.net



Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
**GUI tools**

Loose ends

-

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
**Loose ends**  
-

Remember *loose objects* ?

# Loose objects

```
.git/objects
```

```
|-- 23
|   '-- d4bd826aba9e29aaace9411cc175b784edc399
|-- 76
|   '-- 49f82d40a98b1ba59057798e47aab2a99a11d3
|-- c4
|   '-- aaefaa8a48ad4ad379dc1002b78f1a3e4ceabc
|-- e7
|   '-- 4be61128eef713459ca4e32398d689fe80864e
|-- info
|   '-- packs
'-- pack
    |-- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.idx
    '-- pack-b7b026b1a0b0f193db9dea0b0d7367d25d3a68cc.pacc
```

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

Loose ends

# Loose objects

```
$ git gc
```

```
Counting objects: 636, done.
```

```
Compressing objects: 100% (635/635), done.
```

```
Writing objects: 100% (636/636), done.
```

```
Total 636 (delta 486), reused 0 (delta 0)
```

```
Removing duplicate objects: 100% (256/256), done.
```

- ▶ GC happens automatically when you get to 6700 objects (configurable)

# Loose objects

```
$ git gc
```

```
Counting objects: 636, done.
```

```
Compressing objects: 100% (635/635), done.
```

```
Writing objects: 100% (636/636), done.
```

```
Total 636 (delta 486), reused 0 (delta 0)
```

```
Removing duplicate objects: 100% (256/256), done.
```

- ▶ GC happens automatically when you get to 6700 objects (configurable)



Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

**Loose ends**

-

# Stash

# Stashing

- ▶ you're hacking
- ▶ got debug code you don't want to commit
- ▶ need to work on something else

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

# Stashing

- ▶ you're hacking
- ▶ got debug code you don't want to commit
- ▶ need to work on something else

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

## Repository

Structure  
Objects

## Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

# Stashing

- ▶ you're hacking
- ▶ got debug code you don't want to commit
- ▶ need to work on something else

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

# Stashing

- ▶ `git stash "description"`
- ▶ do that other thing
- ▶ `git stash apply`

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

# Stashing

- ▶ `git stash "description"`
- ▶ do that other thing
- ▶ `git stash apply`

## Concepts

SCM components  
SCM operations  
Decentralization

## GIT History

### Repository

Structure  
Objects

### Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools

**Loose ends**

-

# Stashing

- ▶ `git stash "description"`
- ▶ do that other thing
- ▶ `git stash apply`

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-



## Further reading

[git-scm.org](https://git-scm.org)  
[git.or.cz/gitwiki](https://git.or.cz/gitwiki)

[gitready.com](https://gitready.com)

[GitCasts.com](https://gitcasts.com)  
(Scott Chacon)

[GitHub.com](https://github.com)

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Further reading

[git-scm.org](https://git-scm.org)  
[git.or.cz/gitwiki](https://git.or.cz/gitwiki)

[gitready.com](https://gitready.com)

[GitCasts.com](https://gitcasts.com)  
(Scott Chacon)

[GitHub.com](https://github.com)

Git the basics

Bart  
Trojanowski,  
bart@jukie.net

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Further reading

[git-scm.org](https://git-scm.org)  
[git.or.cz/gitwiki](https://git.or.cz/gitwiki)

[gitready.com](https://gitready.com)

[GitCasts.com](https://gitcasts.com)  
(Scott Chacon)

[GitHub.com](https://github.com)

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Further reading

[git-scm.org](http://git-scm.org)  
[git.or.cz/gitwiki](http://git.or.cz/gitwiki)

[gitready.com](http://gitready.com)

[GitCasts.com](http://GitCasts.com)  
(Scott Chacon)

[GitHub.com](http://GitHub.com)

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Further reading

[git-scm.org](http://git-scm.org)  
[git.or.cz/gitwiki](http://git.or.cz/gitwiki)

[gitready.com](http://gitready.com)

[GitCasts.com](http://GitCasts.com)  
(Scott Chacon)

[GitHub.com](http://GitHub.com)

Git the basics

Bart  
Trojanowski,  
[bart@jukie.net](mailto:bart@jukie.net)

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

## Further reading

<http://www.jukie.net/~bart/blog/>

<git://git.jukie.net/intro-to-git.git/>

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

## Further reading

<http://www.jukie.net/~bart/blog/>

<git://git.jukie.net/intro-to-git.git/>

Git the basics

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components

SCM operations

Decentralization

GIT History

Repository

Structure

Objects

Using GIT

Commands

Commit

Inspection

Branching

Merging

Rebasing

Remotes

GUI tools

Loose ends

-

**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

Thank you.



**Bart  
Trojanowski,  
bart@jukie.net**

Concepts

SCM components  
SCM operations  
Decentralization

GIT History

Repository

Structure  
Objects

Using GIT

Commands  
Commit  
Inspection  
Branching  
Merging  
Rebasing  
Remotes  
GUI tools  
Loose ends  
-

Q & A