

# JGit

"Quo vadis?"

# What Is JGit?

100% Pure Java implementation of Git

# What Is JGit?

100% Pure Java implementation of Git  
No native code

# What Is JGit?

100% Pure Java implementation of Git

No native code

No fork+exec of "C Git" (aka git.git, aka Junio's tree)

# Why Reimplement?

fork+exec performance is slow

# Why Reimplement?

fork+exec performance is slow... especially on Win32

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)
- high level development language

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)
- high(er) level development language

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)
- high(er) level development language
- easier to include in IDE distributions

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)
- high(er) level development language
- easier to include in IDE distributions
  - NetBeans
  - Eclipse
  - ...

# Why Reimplement?

fork+exec performance is slow... especially on Win32  
No Git library... GSoC 2007 project never completed

100% Pure Java gives us:

- automatically cross platform (Win32, POSIX)
- high(er) level development language
- easier to include in IDE distributions
  - NetBeans
  - Eclipse
  - ...
- server side applications (e.g. servlets)

# Current Features

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

# Current Features

Object Database: read/write loose, read packed, idx v1, v2  
Parsers for: trees, commits, tags

# Current Features

Object Database: read/write loose, read packed, idx v1, v2  
Parsers for: trees, commits, tags (two implementations!)

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

Fetch/Push Client Transport:

http://, ftp://, sftp://, amazon-s3:// (dumb walker)

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

Fetch/Push Client Transport:

http://, ftp://, sftp://, amazon-s3:// (dumb walker)

git://, ssh://, file:// (executes git-{upload,receive}-pack)

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

Fetch/Push Client Transport:

http://, ftp://, sftp://, amazon-s3:// (dumb walker)

git://, ssh://, file:// (executes git-{upload,receive}-pack)

bundle (fetch only)

no rsync://

# Current Features

Object Database: read/write loose, read packed, idx v1, v2

Parsers for: trees, commits, tags (two implementations!)

Ref "Database": read/write loose, packed, append reflog

Revision "Machinery": rev-list, --objects, merge-base

Graphic Log: (aka gitk) for AWT, SWT, ...

Fetch/Push Client Transport:

http://, ftp://, sftp://, amazon-s3:// (dumb walker)

git://, ssh://, file:// (executes git-{upload,receive}-pack)

bundle (fetch only)

# Where Are We Going?

Backend for:

# Where Are We Going?

Backend for:

IDEs (NetBeans, Eclipse, ...)

# Where Are We Going?

Backend for:

IDEs (NetBeans, Eclipse, ...)

Servers (Gerrit, ...)

# Where Are We Going?

Backend for:

IDEs (NetBeans, Eclipse, ...)

Servers (Gerrit, ...)

Missing Features:

Merge

Diff (patch generation)

Patch Apply

Blame

Rename Detection