

**NAME**

`git-request-pull` – Generates a summary of pending changes

**SYNOPSIS**

```
git request-pull [-p] <start> <url> [<end>]
```

**DESCRIPTION**

Generate a request asking your upstream project to pull changes into their tree. The request, printed to the standard output, begins with the branch description, summarizes the changes and indicates from where they can be pulled.

The upstream project is expected to have the commit named by **<start>** and the output asks it to integrate the changes you made since that commit, up to the commit named by **<end>**, by visiting the repository named by **<url>**.

**OPTIONS**

`-p`

Include patch text in the output.

**<start>**

Commit to start at. This names a commit that is already in the upstream history.

**<url>**

The repository URL to be pulled from.

**<end>**

Commit to end at (defaults to HEAD). This names the commit at the tip of the history you are asking to be pulled.

When the repository named by **<url>** has the commit at a tip of a ref that is different from the ref you have locally, you can use the **<local>:<remote>** syntax, to have its local name, a colon `:`, and its remote name.

**EXAMPLES**

Imagine that you built your work on your **master** branch on top of the **v1.0** release, and want it to be integrated to the project. First you push that change to your public repository for others to see:

```
git push https://git.ko.xz/project master
```

Then, you run this command:

```
git request-pull v1.0 https://git.ko.xz/project master
```

which will produce a request to the upstream, summarizing the changes between the **v1.0** release and your **master**, to pull it from your public repository.

If you pushed your change to a branch whose name is different from the one you have locally, e.g.

```
git push https://git.ko.xz/project master:for-linus
```

then you can ask that to be pulled with

```
git request-pull v1.0 https://git.ko.xz/project master:for-linus
```

**GIT**

Part of the **git(1)** suite