NAME

git-archimport - Import a GNU Arch repository into Git

SYNOPSIS

git archimport [-h] [-v] [-o] [-a] [-f] [-T] [-D <depth>] [-t <tempdir>]
<archive>/<branch>[:<git-branch>]...

DESCRIPTION

Imports a project from one or more GNU Arch repositories. It will follow branches and repositories within the namespaces defined by the <archive>/<branch> parameters supplied. If it cannot find the remote branch a merge comes from it will just import it as a regular commit. If it can find it, it will mark it as a merge whenever possible (see discussion below).

The script expects you to provide the key roots where it can start the import from an *initial import* or *tag* type of Arch commit. It will follow and import new branches within the provided roots.

It expects to be dealing with one project only. If it sees branches that have different roots, it will refuse to run. In that case, edit your <archive>/<branch> parameters to define clearly the scope of the import.

git archimport uses **tla** extensively in the background to access the Arch repository. Make sure you have a recent version of **tla** available in the path. **tla** must know about the repositories you pass to *git archimport*.

For the initial import, *git archimport* expects to find itself in an empty directory. To follow the development of a project that uses Arch, rerun *git archimport* with the same parameters as the initial import to perform incremental imports.

While *git archimport* will try to create sensible branch names for the archives that it imports, it is also possible to specify Git branch names manually. To do so, write a Git branch name after each <archive>/<branch> parameter, separated by a colon. This way, you can shorten the Arch branch names and convert Arch jargon to Git jargon, for example mapping a "PROJECT--devo--VERSION" branch to "master".

Associating multiple Arch branches to one Git branch is possible; the result will make the most sense only if no commits are made to the first branch, after the second branch is created. Still, this is useful to convert Arch repositories that had been rotated periodically.

MERGES

Patch merge data from Arch is used to mark merges in Git as well. Git does not care much about

tracking patches, and only considers a merge when a branch incorporates all the commits since the point they forked. The end result is that Git will have a good idea of how far branches have diverged. So the import process does lose some patch-trading metadata.

Fortunately, when you try and merge branches imported from Arch, Git will find a good merge base, and it has a good chance of identifying patches that have been traded out-of-sequence between the branches.

OPTIONS

-h
: Display usage.

-v
: Verbose output.

-T
: Many tags. Will create a tag for every commit, reflecting the commit name in the Arch repository.

-f
: Use the fast patchset import strategy. This can be significantly faster for large trees, but cannot handle directory renames or permissions changes. The default strategy is slow and safe.

-o
: Use this for compatibility with old-style branch names used by earlier versions of *git archimport*. Old-style branch names were category--branch, whereas new-style branch names are archive,category--branch--version. In both cases, names given on the command-line will override the automatically-generated ones.

-D <depth>
: Follow merge ancestry and attempt to import trees that have been merged from. Specify a depth greater than 1 if patch logs have been pruned.

-a
: Attempt to auto-register archives at **http://mirrors.sourcecontrol.net** This is particularly useful with the -D option.

-t <tmpdir>
: Override the default tempdir.

<archive>/<branch>

<archive>/<branch> identifier in a format that **tla log** understands.

GIT

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