urls

GIT URLS ----------

In general, URLs contain information about the transport protocol, the address of the remote server, and the path to the repository. Depending on the transport protocol, some of this information may be absent.

Git supports ssh, git, http, and https protocols (in addition, ftp and ftps can be used for fetching, but this is inefficient and deprecated; do not use them).

The native transport (i.e. `git://` URL) does no authentication and should be used with caution on unsecured networks.

The following syntaxes may be used with them:

- `ssh://[<user>@]<host>[:<port>]/<path-to-git-repo>`
- `git://<host>[:<port>]/<path-to-git-repo>`
- `http[s]://<host>[:<port>]/<path-to-git-repo>`
- `ftp[s]://<host>[:<port>]/<path-to-git-repo>`

An alternative scp-like syntax may also be used with the ssh protocol:

- `[<user>@]<host>:/<path-to-git-repo>`

This syntax is only recognized if there are no slashes before the first colon. This helps differentiate a local path that contains a colon. For example the local path `foo:bar` could be specified as an absolute path or `./foo:bar` to avoid being misinterpreted as an ssh url.

The ssh and git protocols additionally support `~<username>` expansion:

- `ssh://[<user>@]<host>[:<port>]/~<user>/<path-to-git-repo>`
- `git://<host>[:<port>]/~<user>/<path-to-git-repo>`
- `[<user>@]<host>:~<user>/<path-to-git-repo>`

For local repositories, also supported by Git natively, the following syntaxes may be used:

- `/path/to/repo.git/`
- `file:///path/to/repo.git/`

ifndef::git-clone[] These two syntaxes are mostly equivalent, except when cloning, when the former implies `--local` option. See linkgit:git-clone[1] for details. endif::git-clone[]

ifdef::git-clone[] These two syntaxes are mostly equivalent, except the former implies `--local` option. endif::git-clone[]

`git` `clone`, `git` `fetch` and `git` `pull`, but not `git` `push`, will also accept a suitable bundle file. See linkgit:git-bundle[1].

When Git doesn't know how to handle a certain transport protocol, it attempts to use the `remote-<transport>` remote helper, if one exists. To explicitly request a remote helper, the following syntax may be used:

- `<transport>::<address>`

where *<address>* may be a path, a server and path, or an arbitrary URL-like string recognized by the specific remote helper being invoked. See linkgit:gitremote-helpers[7] for details.

If there are a large number of similarly-named remote repositories and you want to use a different format for them (such that the URLs you use will be rewritten into URLs that work), you can create a configuration section of the form:

> "
> ```
> [url "__<actual-url-base>__"]
>     insteadOf = _<other-url-base>_
> ```

For example, with this:

```
[url "git://git.host.xz/"]
    insteadOf = host.xz:/path/to/
    insteadOf = work:
```

a URL like "work:repo.git" or like "host.xz:/path/to/repo.git" will be rewritten in any context that takes a URL to be "git://git.host.xz/repo.git".

If you want to rewrite URLs for push only, you can create a configuration section of the form:

> "
> ```
> [url "__<actual-url-base>__"]
>     pushInsteadOf = _<other-url-base>_
> ```

For example, with this:

```
[url "ssh://example.org/"]
    pushInsteadOf = git://example.org/
```

a URL like "git://example.org/path/to/repo.git" will be rewritten to "ssh://example.org/path/to/repo.git" for pushes, but pulls will still use the original URL.

□□□ #1
□ □□□ □□□ 2026-10-31 20:17:36 UTC
□ □□□ □□□ 2026-10-31 20:17:37 UTC