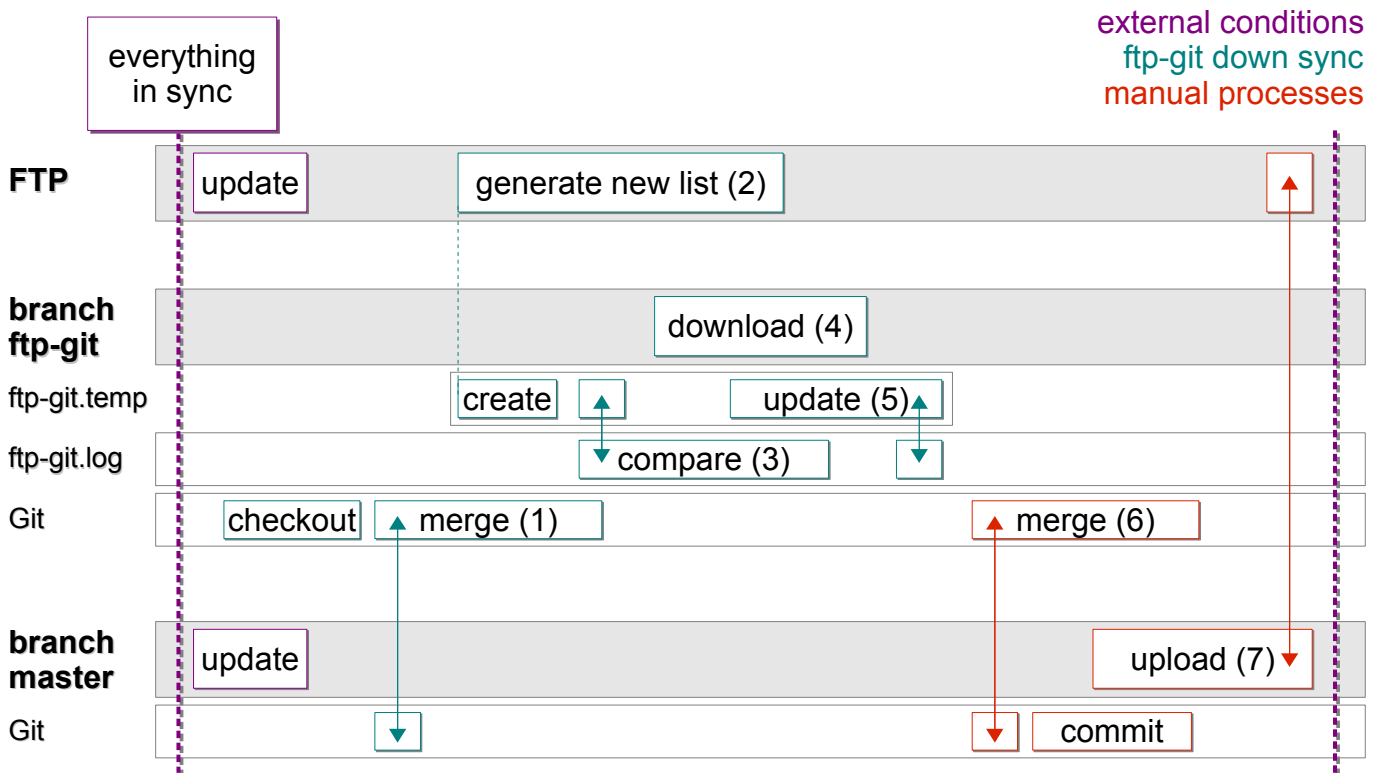


# Synchronizing working directory with FTP



- (1) Merge master onto ftp-git branch. The ftp-git branch has to be up to date, so that a fast-forward merge back onto master is possible. If this fails, abort and ask user to bring the ftp-git branch up to date.
- (2) Generate and download the new list. Store temporarily.
- (3) Compare the new list to the one in the working directory. The result is the set of changes that were made since the last down sync before the current commit. Storing the list in git makes resyncing an old commit possible.
- (4) Download all changes. Since the comparison is based on the last time, the down sync was performed, files that were changed and uploaded in the mean time will just be downloaded again. This is OK, because git will recognize, that nothing has actually changed. Further, git-ftp and ftp-git can only be used independently if we choose this approach.
- (5) Update the list in the working directory and perform a git add on it. If the user merges the branches successfully, the next sync will be based on this list.
- (6) Ask user to merge ftp-git branch onto master branch. This is where the users can check what has changed and decide how to handle it.
- (7) Upload the changes. **The changes that made it through the merge will be downloaded again when the next down sync is made, since the list is not updated.**

Update ftp-git.log partially when uploading via git-ftp?  
 Use git-ftp for up sync => extend diagram