

All change comments to perforce branch between 2 labels? (including merges)

Asked 15 years, 11 months ago Modified 15 years, 10 months ago

Viewed 4k times



Our perforce admin limits "max-row" scans so that my first idea of running the following will not work:

1



1. All changes including integrates into a branch at particular label time 1



2. All changes including integrates into a branch at particular earlier label time 2



3. Subtract time 2 changes from time 1 to get the new changes with comments.

Is there an alternative way of getting the same result without having such a massive query (when perforce contains 7yrs of history and -i triggers a scan back to the dawn of history)

Based on Gregs comments added this comment:

Basically the point is to see what bugs got fixed in a particular release branch between 2 labels (or more commonly, some old label and today). I wish to simplify (speedup) way too complex script that we

currently have which looks at changes that went into a release branch, it follows files that went into them at least 2 branches up in order to printout all the changeset comments from the original change(the interim merge comments tend to just say something like merge123 etc instead of description of the actual change comments, so we need to walk up the tree to the original comment as well), script finally outputs something like below(we put quality center IDs into changeset comments):

1. *qualityCenterId123 - fixed some bug*
2. *in gui qcId124 - fixed some other*
3. *bug qcId125 - fixed some other bug*
4. *merge123*

UPDATE based on comments:

Problem with Toby's approach is that most of the changes into the code branch came via integrations, -i would include those change, but as stated that explodes the query to such a degree that due the load on perforce server our admin won't allow it to run. So this is why I am looking for an alternative approach to get the same result.

version-control

merge

perforce

Share

edited Jan 31, 2009 at 1:57

Improve this question

Follow

asked Jan 22, 2009 at 22:19



Ville M

2,029 ● 8 ● 31 ● 47

4 Answers

Sorted by:

Highest score (default)



2

I can't see an easy answer to this, but do have a couple more suggestions that perhaps may help point in the right direction.



+100



1. Persuade your admin to raise the maxscan rows limit. If he is nervous that this will lead to problems with the whole user base, just get him to add you to a new user group (e.g. "Scripting"), and set the limits for just that group. That will have the effect that only members of that group can use the upper limits, and you can then negotiate for suitable times to run the script. You could even do it overnight.
2. Have a look at the [P4 admin guide](#) and see if any of the hints on scripting will help - e.g. maybe a tighter view on the data will limit the query enough to not break the maxscanrows limits.
3. How's your SQL? You may be able to construct an efficient query using the [P4Report](#) tool.
4. Try asking the question on the Perforce mailing list. It's a very active list that has a lot of very experienced people who are very helpful. See [this link](#) for the sign-up page. There's a good chance that they will suggest some good approaches.

5. Probably too late for your existing labels, but consider using the job system to track work. Perforce has inbuilt query tools to track what jobs have made it into different branches. It does require a working-practice change for your team, however.

Sorry I can't provide a more specific answer.

Share Improve this answer

answered Feb 4, 2009 at 23:20

Follow



[Greg Whitfield](#)

5,719 ● 2 ● 31 ● 32

I think this is the answer I was looking for, "no easy answer", good to know, will follow up on the list etc as you suggested, thanks. – [Ville M](#) Feb 5, 2009 at 17:37

Thanks. It's one of those tricky problems where you already have a solution that does the main problem of generating a report, but it does not fit within the constraints you are under. Good luck. – [Greg Whitfield](#) Feb 5, 2009 at 22:03



1



Are your labels more than simply the most recent changelist when they were created? Eg did you really need to record specific files in client workspaces? If not you can very easily compare the two changelists closest to labels.



Say the closest change to your first label date is 23000 and your closes change to your second label date is 25000 then

p4 changes

//depot/PATHTOMYCODE/...@23000,@25000

will give you all changes to your code path between these two changelists.

Share Improve this answer

answered Jan 26, 2009 at 17:27

Follow



Toby Allen

11.2k ● 12 ● 79 ● 131

Nope - see answers.perforce.com/articles/KB/3429 which agrees with my experiments - the resulting output is incorrect.

– Arlie Stephens Feb 10, 2016 at 23:17



1



Won't a normal Label-diff do what you want?

- From P4V, Tools->Diff. Select the two labels
- From P4Win, right click label, select diff files in 2 labels
- From command line, p4 diff2 //codeline/...@label1 //codeline/...@label2

Or am I missing exactly what you are after?

Further suggestion after Ville's comment on the above

If you are only after the info per changelist, rather than per file, then try "p4 interchanges" from the command

line. This will give you just the summary of what changes in one branch have not happened in another, and you can supply a revision range to limit it to the labels you need.

Do "p4 help interchanges" from command line for details.

Unfortunately the interchanges command is not yet exposed in P4V or P4Win.

Share Improve this answer

edited Jan 27, 2009 at 11:04

Follow

answered Jan 26, 2009 at 13:28



[Greg Whitfield](#)

5,719 ● 2 ● 31 ● 32

I would need the comments for each change, using the diff 2 labels approach doesn't print comments for each change.

– [Ville M](#) Jan 26, 2009 at 21:28

I've edited my answer to give you another possibility. If that does not work for you, could you perhaps type a few lines of what you expect the output to look like? – [Greg Whitfield](#) Jan 27, 2009 at 11:05

Thanks,I did just start looking into the interchanges command,very interesting.Your question;I wish to simplify(speedup) script that looks at changes into branch,follows up to 2 branches to printout changeset comments,outputs: qualityCenterId123 - fixed some bug in gui qcld124 - fixed some other bug – [Ville M](#) Jan 29, 2009 at 22:48

I been playing with the p4 interchanges command, but one thing I don't quite understands is if I give this: p4

interchanges -l //depot/path/...@label_old //depot/path/... it will say all already integrated, even though diff between head and that label will show difference? Why is that? – [Ville M](#)

Jan 30, 2009 at 19:09



Toby Allen's Answer is the best approach if your labels are simple changelists.

0



If the labels are more complicated, then I think you need to look at all the files in each label and where their versions differ, find the changelist where the version changed.



You can get the file and version lists with:

```
p4 fstat -Of //...@MyLabel
```

EDIT:

Consider two complex labels:

```
VERSION_A:  
  //depot/file_A.cpp#4  
  //depot/file_B.cpp#7  
  //depot/file_C.cpp#1  
  
VERSION_B:  
  //depot/file_A.cpp#6  
  //depot/file_B.cpp#5  
  //depot/file_C.cpp#4
```

In this example, the labels do not describe a particular changelist, the head change for each file may be

different.

If you can have labels like this, then you can run the p4 fstat command on each label and then find the differences. In this example, `file_A.cpp` has changed twice and `file_C.cpp` has changed 3 times. `file_B.cpp` is older in the second label, so it can be ignored.

So now you need to look at the changes that involved these versions:

```
file_A.cpp#5  
file_A.cpp#6  
file_C.cpp#2  
file_C.cpp#3  
file_C.cpp#4
```

Those changes can be retrieved with p4 filelog, so you want to run something like this:

```
p4 filelog file_A.cpp#6  
p4 filelog file_C.cpp#4
```

Then you need to remove any duplicates and any history for earlier versions.

Like I said, you only need this if you have messy labels. If there's any way to make your labels represent changelists, you should use Toby Allen's answer.

Share Improve this answer

edited Feb 4, 2009 at 1:17

Follow

answered Jan 30, 2009 at 20:30



[Mark James](#)

681 ● 4 ● 5

Added UPDATE to question based on your answers first part. Not sure if I got understand how the fstat will help me get the commit comments between 2 labels? – [Ville M](#) Jan 31, 2009 at 1:58

The fstat will get you the version numbers for each file in a label. My assumption was that your labels were not simply the most recent changelist when they were created. If the labels are complicated you need to look at the individual file versions, then find the diffs. I'll add details above.
– [Mark James](#) Feb 4, 2009 at 1:01

Thinking about it more, you're still going to need to run the filelog with the -i flag, so my example probably doesn't help you. – [Mark James](#) Feb 4, 2009 at 1:20

Mark, I think you are right, the whole problem really is the -i flag and the huge query it triggers. I maybe just need to change my process, and run the interchanges command before doing the merge, and it should more or less give me what I need. However I'll leave the question open for a bit more. – [Ville M](#) Feb 4, 2009 at 19:44
