P4Bucket

Or … Now you see it, now you don’t

What is it?

- P4Bucket implements a set of buckets
  - The term comes from AlienBrain and is familiar to users
- A bucket is
  - A unique name
  - A location on an accessible hard drive (direct, network)
  - A place where binary assets (that is, depot files) go when they are old but not obsolete (no obliterate or +S)
- Why?
  - Reduced verify and backup time
  - Reduced space requirement for the Perforce server
  - One example: 1 file, 360 revisions, 66 GB old data

Requirements

- In order to use P4Bucket you need:
  - A Perforce server (capable of fstat -Oazl)
  - Python 2.5 or higher
  - P4Python 2008.2 or higher (for P4.Map)
  - A user that can access the depot files and buckets

  - The tool can only be run on the server host
    - This will be checked by the tool
  - The server needs to be running to use this tool

What can you archive?

- Only binary files of type +F and +C can be archived
  - Not +Sx – those files would be purged eventually
  - Not +D – delta files cannot be archived (rare for binary)
- You cannot archive lazy copies
- You cannot archive sources of lazy copies
  - But you can snap them automatically
- All these rules are in place to follow the guideline of “least surprise”

USAGE OF THE SCRIPT

- python p4bucket.py command [options]
  - init
  - create <name> -r <root>
  - edit <name> -r <root>
  - delete <name>
  - list [name] –a

  - archive –b <name> [-n] [opts] file[range]...
  - restore –b <name> [-n] file[range]...
    - opts are

-f	force head revision archiving
-s	snap lazy copies
-m <size>	only archive files larger than size

Usage (cont)

- First, set the location of the config file in the script
  - CONFIG_FILE = "p4bucket.conf“
- Run “p4bucket init” to define server parameters
- Define any set of buckets with their root directory
- Now you ready to archive:
  - p4bucket archive bucket1 //depot/assets/...
- By default, no #head revision is archived
  - unless –f is specified, for example for old branches
- Use –m <size> to limit archiving to larger files

WHAT ACTUALLY HAPPENS IN THE ARCHIVE?

- File gets moved to identical location in the bucket
  - bucketroot/depotname/directory/filename,d/1.18723.gz
- File is replaced with a placeholder (ghost)
  - "# P4BUCKET: This file has been archived.\n“
  - followed by some information about where and when
- A set of 4 attributes is added to the file revision
  - "archiveUser"
  - "archiveDate"
  - "archiveDigest"
  - "archiveBucket"
- The digest is recalculated (“p4 verify –v”)

RESTORE OPERATION

- p4bucket restore bucket1 //depot/assets/...
  - bucket name has to match stored bucket name
  - allows users to filter different assets in separate buckets
- Attributes have to exist or the restore will fail
- You can restore by hand if necessary
  - move the file back to the depot from the bucket
  - run verify –v on the revision
  - remove the attributes (there is a script removeAttr.py)

REPORTING MODE (-N)

- If run in reporting mode, archive reports which files it would archive
- Completes with a summary reporting total number of file revisions and size of these files

Documentation

- Please see P4Bucket.pdf

- Find the tool and documentation in the public depot
  - //guest/sven_erik_knop/p4bucket