FAQ for CodeEasy   
(Frequently Asked Questions)

Below is a list of frequently asked questions and answers to typical problems experienced while learning and adopting the CodeEasy FlexClones for DevOps methodology.

Q0: Which Source Control tools work with CodeEasy/FlexClones?

A0: Perforce, Git, Subversion (SVN) and CVS have all been tested with Perforce. In fact Perforce and NetApp have created an open source Perforce plug-in called “P4 FlexClone” which can be found at <https://swarm.workshop.perforce.com/projects/perforce-software-p4flexclone> which implements a similar FlexClone methodology as to what is provided in this kit.

Q7: Can the CodeEasy/FlexClone methodology work with ClearCase?

A7: Yes and No. Clearcase workspaces can be created in two ways. The most common way is for users to create a VOB based workspace. A Versioned Object Base (VOB) is a centralized database that stores version information about the files and folders in a software configuration management (SCM) system. VOBs are like a virtualized storage container that can NOT be cloned. ClearCase users can also create a non-VOB based workspace where the files and directories are checked out directly into the local file system. The non-VOB based workspaces can be cloned using NetApp’s FlexClone technology.

Q0: What is Snapshot limit per volume?

A0: 255 Snapshots are allowed per volume. This limit will require a flow to clean-up unused snapshots. One way to do this is to name snapshots with a date stamp. This way Snapshots older than a certain date that are not associated with a FlexClone can be systematically removed. Refer to the docs/ Continuous\_Integration\_with\_NetApp\_Technology\_v2.docx.pdf document “Management Tasks” section for ideas for how to purge snapshots.

Q1: How many FlexClone can I have.

A1: The answer depends upon the NetApp FAS controller. A FAS8080 running cDOT8.2.x or later can support 1000 FlexClones per node. A 12 node cluster can support 12x1000 (or 12,000) FlexClones.

Q1: I don't see any storage savings (efficiencies) from my FlexClones.

A1: Check that the FlexClone was creating with Thin Provisioning enabled. By default, a FlexClone will take the provision settings of the parent volume. If the parent volume is Thick Provisioned, then the FlexClone will be Thick Provisioned. Add space-reserve='none' to the volume-clone-create command to override the parent volume settings. NOTE: the cDOT cmdline option and the API options are different. 'space-reserve' is the correct command for the API.

Q3: The CodeEasy scripts will not connect to the filer.

A3: All of the CeCreate\*.pl scripts include a -test option for just testing the API access connection to the filer. The typical API connections are as follows;

1. The filer/user access is not setup correctly. Make sure the user has ‘ontapi’ and ‘ssh’ access permissions for the vserver. Check the QUICKSTART document for setup instructions.
2. Check that the file/user can access the vserver command interface via ssh.
3. I have seen issues where the filer/user password contained special characters which did not pass thru correctly via the Perl API. Try a simple password and see if that works.

Q4: Will having too many clones on one controller cause too much load to be on that filer? How can I manage that?

A4: Yes it is possible that too many FlexClone volumes will be created on a single controller. Refer to the docs/ Continuous\_Integration\_with\_NetApp\_Technology\_v2.docx.pdf document “Volume Move” section which describes a methodology for regularly moving FlexClone volumes for load balancing.

Q5: Can this process be implemented in 7-mode?

A5: Yes but instead of using vservers with junction path relationships, FlexClones will be create directly on the 7-mode controller HA pairs and the new FlexClone volumes will need to be mounted using the UNIX host mount or auto-mount commands.

Q6: Workspaces live for a long time, even years. Will this cause snapshots to be locked for that period of time? Over time these workspaces will diverge considerably from the original clone – does this mean that the overall value diminishes?

A6: It is possible based on your design flow methodology that some FlexClones might live for a long time. Typical SW or HW workspaces are short lived as developers create clones as a way of getting the latest code builds. Once a newer build is available, developers typically create a new clone of the latest code line. Over time old out of date workspaces lose there value and are deleted. The CodeEasy tool kit now contains a CeSplitCone.pl script which allows the user to “split” the FlexClone from it’s parent volume/snapshot pair. Once split the FlexClone volume becomes a normal volume and loses the storage efficiencies associated with a clone. That said the new volume, can be move or archived like any other volume.

Q8: How do I manage all of these FlexClones? How do I know when I can delete them?

A8: There are a number of maintenance tasks associated with the CodeEasy/FlexClone methodology. Those maintenance tasks are documented in Refer to the docs/Continuous\_Integration\_with\_NetApp\_Technology\_v2.docx.pdf document section Maintenance Tasks.

Q9: What are the benefits of using FlexClones?

A9: There are multiple benefits which both improve storage efficiencies and engineering development time.

1. Since FlexClones contain the latest source code and the latest build artifacts, there is a NO IOPs generated from the code checkout or initial code build operations and virtually no storage consumed in by the FlexClone.
2. Creating a FlexClone is almost instantaneous, so the developer does not have to wait for the code checkout time and the initial build time. In some environments this can same minutes to hours depending on the size of the source code checkout and the build time.
3. Since FlexClones share common blocks of storage, there is a higher probability that the common blocks or storage will be cached in the FlashCache card. This leads to very low read latency and in turn faster filer performance.

Q10: Change the ownership of the files in the CVS directories

A10: %> find . -name Root -type f | xargs -P 140 perl -pi -e 's/old\_user/new\_user/g’