zCustom.SnapBackup Site Notes

Version History

* 20131101 – initial version, Glen Pitt-Pladdy (InterSystems)
* 20131118 – updated with backup history, Glen Pitt-Pladdy (InterSystems)
* 20131212 – updated with JournalSwitch function, Glen Pitt-Pladdy (InterSystems)
* 20131219 – added LVM2 warning level, Glen Pitt-Pladdy (InterSystems)
* 20140714 – ready for giving to customers, Glen Pitt-Pladdy (InterSystems)
* 20140812 – notes about excluding some databases, Glen Pitt-Pladdy (InterSystems)

# Purpose

Backups are fundamental to a Business Continuity plan and up to the site to ensure that systems are backed up and backups tested in accordance with site policies. InterSystems provide a set of tools to enable backup of TrakCare which sites may use to integrate TrakCare into their backup systems. It is good practice to regularly review and test backup systems.

With this in mind backup tools for TrakCare deployments in the UK have been updated introducing a number of improvements. The aims of this iteration of tools are:

* Consolidate the methods used for backup for all UK sites going forward. This means that we can leverage scale to ensure the tools are well tested and provide better focused support
* Ensure that the tools were as robust as practical and as effective as possible at detecting failures and anomalies that occur
* Provide a flexible interface to allow sites to easily integrate TrakCare aspects of backups into their backup systems
* Simplify restore, allowing entire systems to be captured and restored easily without the need to rebuild systems (eg. database-only backups)

# TrakCare Backup Strategy

InterSystems is not a backup vendor nor consultant and the appropriate strategy depends on the Business Continuity policies and requirements of a particular site. Typically InterSystems will provide and configure the tools for backing up TrakCare and the particular requirements of the site and integration into the site's backup systems will be handled by the site and their suppliers.

This section (and document) provides high-level guidance on key TrakCare relevant aspects of backup that are important to consider in your backup strategy, but the overall backup strategy needs to be determined by the site relevant to their own needs.

## Parts of the TrakCare system

TrakCare it's self is largely a set of databases and the web/ directory full of static files, plus the associated cache.cpf that describes how the data is stored in the databases. These are the major parts needed to recover, however there is considerable other components and configuration outside of this that will make a big difference to how easily and quickly the system can be recovered. Without the other parts recovery is largely a process of rebuilding and could take days or weeks to complete depending on the particular design and resources available.

### Minimum data set

This is the absolute minimum required to rebuild a TrakCare system, but the process may take days or weeks:

* TrakCare Databases – this is typically by far the biggest data volume. There is the possibility (with additional risk) of excluding some databases:
  + MONITOR – this is system performance data and if purges are not in place can grow large, however maintained properly should be considerably smaller than DATA
  + inactive AUDIT\* / LOG\* – multiple AUDIT and LOG databases (AUDIT1, AUDIT2… / LOG1, LOG2… etc.) should be in place to allow for archiving. It is possible to consider alternate arrangements with these to avoid backing up the data regularly, however as the data is not changing effective mix of full and incremental backups would minimize the impact of backing up this data anyway.
* TrakCare web/ directory
* cache.cpf for the TrakCare Caché/Healthshare instance

Also consider the same (except web/ directory) for Integration instances:

* Integration Databases
* cache.cpf for the Integration Caché/Healthshare instance

### Additional data

Beyond the above, all additional data and parts of the system that are backed up will save time and human error on restore. These may include but are not limited to:

* Caché / Healthshare as a whole – saves re-installing, re-configuring, any other tweaks that may have been made since build
* Apache configuration
* SSL keys, certificates
* CSP Gateway configuration
* CUPS configuration
* OS configuration
* Third party systems (QAS, SimpleCode etc.)
* Potentially complete systems – saves reinstalling packages etc. and piecing together configuration

Ultimately, if the entire system can be captured as a whole (only practical via snapshot backup) then recovery is largely a case of restoring the whole system and continuing. As the TrakCare databases will typically be by far the biggest part of any backup, the additional overhead of backing up complete systems (assuming effective de-dupe) will generally not be a large impact. Ensuring the right level of data being backed up with the right frequency for the site is something that needs to be taken into account in your backup strategy.

### Web, Application and Print Servers

These are typically very similar in configuration. eg. The major difference between the Application Server #1 and Application Server #8 in and environment is likely IP addresses and host names. The same applies to one web server to another or one print server to another within an environment. With Print servers the one additional difference is the LOCALENS database which holds server specific configuration, however this can be reconfigured quickly on restore in the Ensemble Production configuration.

It's worth considering if a complete backup of all of these is necessary, but at minimum periodic backups of one of each type may be wise for rapid recovery.

### QAS, Simplecode, others

These systems should not be forgotten. In some cases there may be additional Caché/Ensemble/Healthshare instances that may need backup as appropriate for each system. Also see the online-help documentation for backups with these systems.

## Precautions with VM backups

Many VMware backup tools do not take the memory of the VM when snapshots are created for backup despite what people (including in some cases support staff for the backup tool) may believe.

This means that VMware snapshot backups will not provide a consistent means of backing up VMs containing a Caché/Ensemble/Healthshare instance. While it may be able to use the recovery mechanisms build in when the system is restored, it is not recommended to rely on this for backup/restore.

Sites should verify their particular tools are appropriate for the task. A simple test is that if a VM is restored and boots (rather than resumes) after restore then the memory (running state) of the VM was not preserved in the backup.

## Window of loss

The maximum amount of data that can be lost is the interval between backups which will typically be daily with the large volumes of data involved. That means that up to 24 hours of data can be lost.

One option of reducing this with TrakCare and Integration instances is to backup Journals which can then be applied after restoring a nightly backup. As Journals are typically relatively small this means that hourly or more frequent backups of journals are often practical to reduce the window of loss further. This is something to consider in your backup strategy.

## Testing Backup / Restore

An often neglected aspect of backups is testing. The most important part being to test that the system can be restored completely which validates that the data being captured is sufficient to recover in a disaster scenario. Periodic restore tests are something that should be considered as part of a good backup strategy.

# Understanding (Storage) Snapshot Backups

For the purposes of backing up TrakCare where we refer to Snapshot Backups, it's assumed that this is Storage level (eg. SAN, LVM or filesystem) snapshots in conjunction with Write Daemon Freeze/Thaw and on VM snapshots discussed above.

## Snapshot vs Online Backups (database dumps)

Backup:

* Snapshots require only space to store changes while snapshots are active, where Online backups typically requires space to store a full database dump
* Snapshots only introduce performance overhead relating to changes (delta) while snapshots are active, where Online backups typically have to process the entire data set each time for a full backup

Restore:

* Snapshots are simply a case of restoring files where Online backups require to restore the backup file, then to restore the backup file into the database
* Snapshots restore files directly where Online backup requires space to store the backup file before the databases can be restored
* It is feasible to restore an complete machine (OS, applications, files, databases) in one shot using Snapshot backups, where restore will always be a multi-stage process with Online backups where the process is closer to rebuilding TrakCare aspects for Online backups than the restore-and-go with snapshots

## Snapshot Process

The process used for backing up Caché/Healthshare instances and data (which TrakCare runs on) is relatively straight forward:

* The Caché Write Daemon of the instance(s) are frozen to put the databases in a consistent state
* A snapshot of the databases and other relevant data is made using an appropriate mechanism (eg. Filesystem, LVM, SAN)
* The Caché Write Daemon of the instance(s) are thawed to continue normal operation and writing to the databases
* Snapshots are mounted and backed up, potentially along with other static data
* Snapshots are removed
* Recommended: Set Backup History in Caché to enable normal operation of Journal Purging

It is important to ensure that the Write Daemons are frozen for minimal time. With most snapshot mechanisms it should be possible to only freeze Write Daemons for a matter of seconds to allow for snapshots to be created. If frozen for a longer time (typically 600 seconds) the Write Daemon may automatically thaw and the integrity of the data on the snapshot will be uncertain. This should not normally be a problem as any snapshot that takes a significant amount of time is likely to be indicative of a fault.

## Snapshot Mechanisms

There are many different ways that snapshots can be achieved including with mechanisms unique to storage appliances. These should not be confused with VM snapshots which happen at a VM host level where these occur at some point in the storage stack. The two approaches most widely used approaches are:

### SAN Snapshots

In this case SAN features are used to capture a volume. The exact way this is achieved depends on the particular SAN. Key things to note:

* The filesystem on the volume likely must be quiesced before the SAN snapshot is triggered
* There is a performance impact for snapshots, typically for COW (Copy On Write) or back-fill (catchup) when snapshots are removed. This is not normally a problem as backups have a performance impact anyway and will be scheduled after hours.
* Sufficient space needs to be reserved on the SAN to accommodate the snapshots while they are active
* Snapshot volumes need to be sized appropriately to avoid them filling up during the time they are active
* Great care may need to be taken depending on the particular SAN to ensure that snapshots are never filled up as this may affect the data adversely

Sites will need to deploy suitable tools and integrate them into their backups systems along with calling the InterSystems tools to freeze and thaw Caché write daemons.

### Linux LVM2 Snapshots

These are typically used on smaller (possibly virtualized) servers where it is not practical or desirable to use SAN snapshots. Key things to note:

* The filesystem on the volume will be automatically quiesced when the snapshot is created
* There is a performance impact for LVM2 snapshots for COW (Copy On Write) but no back-fill as on some SAN snapshots. This is not normally a problem as backups have a performance impact anyway and will be scheduled after hours.
* Sufficient space needs to be reserved in the Volume Group to accommodate the snapshots while they are active
* Care needs to be taken to size the snapshots to avoid them filling up during the time they are active
* Linux LVM snapshots are automatically discarded if they fill up. This does not affect the source volume which remains operational but the backup integrity will be uncertain if this occurs.

Major Linux distributions ship with LVM2 and some even configure initial installs using it by default. There is a race condition that manifests it's self on some distributions with the removal of snapshots, however there are simple workarounds if this affects systems.

# Tools provided

## zCustom.SnapBackup-YYYYMMDD.xml

This is the Caché side tools used in UK TrakCare deployments which coordinate Write Daemon Freeze and Thaw with the call-in script. This will normally be loaded into the %SYS namespace at build and writes status files prefixed with "zCustom.SnapBackup" to mgr/Temp for each Caché instance it is used with. This also logs actions to cconsole.log

## zCustom.SnapBackup.sh

This is the call-in script which requires Operating System authentication to be enabled for Caché Terminal and will require a user to be configured (typically "cachebackup") in both the Operating System and Caché. This is normally done along with placing the script in /opt/iscscripts/ or hs/scripts/ for each environment (often on clustered builds) at build. It may however require some customization (eg. call-in user) for the particular site.

Note that this script will only work with instances where the corresponding Caché class has been deployed.

This calls methods of the zCustom.SnapBackup class and checks the status files created under mgr/Temp to verify operations have completed successfully.

This script is called as follows:

/path/to/zCustom.SnapBackup.sh Freeze INSTANCE

Where INSTANCE is the instance name you want to freeze, or \_ALL which will cause the script to freeze all instances found with "ccontrol qlist" to be frozen.

/path/to/zCustom.SnapBackup.sh Thaw INSTANCE

Same as above but this time thaws the Write Daemon to continue normal operation.

/path/to/zCustom.SnapBackup.sh History INSTANCE /path/to/log/file

Same as above but this time writes Backup History to allow normal operation of Journal Purging. The final argument (the path to a log file) is optional and allows this to be linked in with the backup history within Caché.

Additionally, if regular (say hourly) backups of Journal files are done then it is safest to switch journal files prior to the backup to ensure that a recent complete journal file is available as there may be problems applying incomplete journal files. For this an additional function is available

/path/to/zCustom.SnapBackup.sh SwitchJournal INSTANCE /path/to/log/file

This requests that this instance switch journals and can be used in a pre-backup script for regular Journal backups.

The return values may be customized by changing the EXITSUCCESS and EXITFAILURE variables (typically 0 and 1 respectively in keeping with Unix standards) as well as the call-in user changed by setting a different CACHEUSR variable (requires running as root to use this) by editing this script. If CACHEUSR is omitted or left blank then the current user is used and the script need not be run as root.

## For Linux LVM2: Initial Capture / Clear Scripts

For builds using Linux LVM2 snapshots (ie. not using SAN snapshots) two scripts are provided as a minimal pre-backup (Capture) and post-backup (Clear) example solution for backing up key volumes used by TrakCare – these are not officially supported by InterSystems and just to give an initial proof of concept. They should be comprehensively tested before depending on them or anything derived from them. These scripts will normally be in /opt/iscscripts/ and named:

* <hostname>\_Capture.sh
* <hostname>\_Clear.sh

These scripts manage LVM2 snapshots and mounting/unmounting the volumes for backups.

The supplied initial scripts will normally mount the key snapshotted volumes on their own directories below /backup/

As there is little value in backing up the data in the corresponding source volumes, the source mounpoints corresponding to the snapshots mounted under /backup/ may be excluded from the backups.

If the site would like to backup the entire system in one tree (may be easier to restore), they may modify these scritps to snapshot or bind (eg. mount /somelocaltion/ /backup/somelocation/ -o bind) other mountpoints from the system below the /backup/ directory, and then backup the /backup/ directory as the root of the system. Note that InterSystems does not support using the same mechanism for backing up the rest of the system and this is better discussed with the site's support suppliers for the relevant parts of the system.

### snaplib.sh and overriding defaults

This is a small library that is included in the Capture and Clear scripts and provides a number of frequently used utility subroutines. These should be clear to anyone skilled in shell scripting by reading the script. These are not supported by InsterSystems and are provided for convenience – ensure anything using them is tested comprehensively before depending on it.

Among the things that this script sets are some sane defaults which may be overridden after the include in the Capture / Clear scripts:

* CALLIN – this is the path to the zCustom.SnapBackup.sh call-in script and defaults to the same location as the running script
* EXITFAILURE – this is the value returned on failure and is set to 1 in keeping with Unix convention
* EXITSUCCESS – this is the value returned on success and is set to 0 in keeping with Unix convention
* WORKAROUNDLVM – this is set to 1 to enable (default) or 0 to disable workarounds for failure to remove LVM2 snapshots seen on some Linux distributions
* LVMWARNINGLEVEL – this sets the % usage of the LVM2 snapshot when it is remove that will trigger a warning about safety margin. The aim is to alert admins that a significant amount of snapshot space is being used and there is a risk of the snapshot becoming full before the backup completes.
* LVMWARNINGFIAL – if non-zero this will cause linuxlvremovesafe() to return 1 when the above warning level is met. The intention is to draw attention to the problem. This is set to 0 (disabled) by default.
* MOUNTPOINT – this is the mountpoint which is used for backup snapshots and defaults to /backup

# Example Scripts (Linux LVM2)

These are basic examples of an environment where all instances, web/ directory and everything else is below a single mountpoint.

## Key areas only

This assumes the system as a whole is backed up and the areas covered by snapshot backups are excluded. Only these key areas are snapshotted and mounted below /backup/ and will have to be relocated their regular location to restore the system.

### Pre-backup

#!/bin/sh –e

# ========================================================================

# Note that InterSystems provides this script as an exxample of backups

# of a TrakCare system and does not support this script nor include it in

# the SLA. It is provided as a basis that sites may use as an example or

# extend it as required.

# ========================================================================

. `dirname $0`/snaplib.sh

backuperror=0

# check for things already mounted

if ! ismounted $MOUNTPOINT; then

echo "ERROR - already mounts under \"$MOUNTPOINT\"" >&2

exit 1

fi

# freeze all write daemons

heading 'Freeze Cache'

$CALLIN Freeze \_ALL || backuperror=$?

heading 'Creating LVM2 Snapshots'

# create LVM Snapshots of all (manual config)

# IMPORTANT: The sizes have to be sufficient to track canges for the full time snapshots are in use

lvcreate --size 20G --name snap\_base --snapshot /dev/mapper/vg\_trakbase/lv\_base || backuperror=$?

# thaw all write daemons

heading 'Thaw Cache'

$CALLIN Thaw \_ALL || backuperror=$?

# mount all the snapshots (manual config)

heading "Mount Snapshots under $MOUNTPOINT"

makemount /dev/vg\_trakbase/snap\_base $MOUNTPOINT/BASE/ -o noatime || backuperror=$?

# TODO optionally put in integrity checks

# sort the overall status

echo

exitwithstatus $backuperror

### Post-backup

#!/bin/sh –e

# ========================================================================

# Note that InterSystems provides this script as an exxample of backups

# of a TrakCare system and does not support this script nor include it in

# the SLA. It is provided as a basis that sites may use as an example or

# extend it as required.

# ========================================================================

. `dirname $0`/snaplib.sh

backuperror=0

# run through mountpoint and unmount everything

heading "Unmounting everything below $MOUNTPOINT"

umountall "$MOUNTPOINT" || backuperror=$?

# remove snapshots (manual config)

heading "Remove LVM2 Snapshots"

for lv in \

/dev/vg\_trakbase/snap\_base \

; do

linuxlvremovesafe $lv || backuperror=$?

done

# mark te backup as complete

[ $backuperror -eq 0 ] && $CALLIN History \_ALL

# sort the overall status

echo

exitwithstatus $backuperror

## Whole system with binds

This allows the whole system tree to be mounted below /backup/ and backed up as a whole with everything in place. This simplifies restore as everything is available. Note that the approach here is to bind existing mountpoints such as / and /var/ below /backup/ however these could also be snapshotted in the same manner.

It's important to understand with the binds that these areas are just binding the two mountpoints together so unlike snapshots, any changes made on bind mounted areas actually change the filesystem where any changes made on the shapshotted volumes only affect the snapshot. This is vital to understand if any additional pre-backup operations are done that modify the filesytems before backup (eg. removing Caché lock files) as this could impact running systems if due care is not taken.

### Pre-backup

#!/bin/sh –e

# ========================================================================

# Note that InterSystems provides this script as an exxample of backups

# of a TrakCare system and does not support this script nor include it in

# the SLA. It is provided as a basis that sites may use as an example or

# extend it as required.

# ========================================================================

. `dirname $0`/snaplib.sh

backuperror=0

# check for things already mounted

if ! ismounted $MOUNTPOINT; then

echo "ERROR - already mounts under \"$MOUNTPOINT\"" >&2

exit 1

fi

# freeze all write daemons

heading 'Freeze Cache'

$CALLIN Freeze \_ALL || backuperror=$?

heading 'Creating LVM2 Snapshots'

# create LVM Snapshots of all (manual config)

# IMPORTANT: The sizes have to be sufficient to track canges for the full time snapshots are in use

lvcreate --size 20G --name snap\_base --snapshot /dev/mapper/vg\_trakbase/lv\_base || backuperror=$?

# thaw all write daemons

heading 'Thaw Cache'

$CALLIN Thaw \_ALL || backuperror=$?

# mount all the snapshots (manual config)

heading "Mount System under $MOUNTPOINT"

mount / $MOUNTPOINT/ –o bind || backuperror=$?

mount /var/ $MOUNTPOINT/var/ –o bind || backuperror=$?

mount /boot/ $MOUNTPOINT/boot/ –o bind || backuperror=$?

makemount /dev/vg\_trakbase/snap\_base $MOUNTPOINT/trak/site/BASE/ -o noatime || backuperror=$?

# TODO optionally put in integrity checks

# sort the overall status

echo

exitwithstatus $backuperror

### Post-backup

#!/bin/sh –e

# ========================================================================

# Note that InterSystems provides this script as an exxample of backups

# of a TrakCare system and does not support this script nor include it in

# the SLA. It is provided as a basis that sites may use as an example or

# extend it as required.

# ========================================================================

. `dirname $0`/snaplib.sh

backuperror=0

# run through mountpoint and unmount everything

heading "Unmounting everything below $MOUNTPOINT"

umountall "$MOUNTPOINT" || backuperror=$?

# remove snapshots (manual config)

heading "Remove LVM2 Snapshots"

for lv in \

/dev/vg\_trakbase/snap\_base \

; do

linuxlvremovesafe $lv || backuperror=$?

done

# mark te backup as complete

[ $backuperror -eq 0 ] && $CALLIN History \_ALL

# sort the overall status

echo

exitwithstatus $backuperror

# Restore Process

As backups capture all files with the key areas snapshotted (eg. mounted below /backup/ in Linux LVM2 examples), restore is simply a case of restoring the backup, ensuring that the files in the snapshots backed up are relocated to the original mountpoints the snapshots were against. There are a number of possible ways that this can be achieved:

* Temporarily mount the volumes in the location of their corresponding snapshots so that restoring the backup will place the files on the volumes, then re-mount them to their normal locations
* Restore the data as it was captured (ie. to locations below /backup/ in Linux LVM2 examples), ensuring that sufficient space is available and then move the data to the relevant volumes/mountpoints
* Re-map the locations to restore paths to in the backup system to restore the snapshot paths to the original volumes
* Do individual restores of each snapshot location to the corresponding original volume
* Extend the backup scripts to mount snapshots (or bind mount) the entire system tree (say below /backup/ in Linux LVM examples) with the snapshots so that it may be restored to / with all the data already in place

After restore and completion of post-restore tasks (eg. for Linux re-install bootloader, reboot etc.) it should be possible to start Caché/Healthshare instances to bring the system back into operation.

It's up to the site to test and verify that systems can be fully restored and decide which approaches are most relevant to achieve the site's backup requirements. Each site should work out the restore process (and any variants on it) to match their particular Business Continuity requirements and backup configuration.

## Note on Interfaces/Integration and Printing

These may all have messages/prints queued on the systems, and on restore these may be re-sent / re-printed if precautions are not taken. The exact impact of this depends on the particular messages/prints involved, however this needs careful consideration as part of the restore process so that impact is avoided.

Possible options for addressing these include:

* Disabling auto-start of CUPS, interfaces, productions
* Starting up after restore on isolated systems (no network) while these are addressed