

# Backup

- How to backup your (Caché) servers reliably?
- Useful tips & pitfalls
- Creating a backup script
- Monitoring your backup!

# How to backup your servers reliably?

- ALWAYS use journaling in Caché (put journal files on a DIFFERENT disk!)
- ALWAYS use RAID (min. level 1 - mirroring) to protect against HDD/SSD failure
- RAID is NO BACKUP!
- SSD's can FAIL too! And when they fail, you have LESS chances to recover!
- Backup DAILY (at least)
- Use external HDD's (USB, NAS) when possible (or backup storage server)
- DISCONNECT your backup (drive) if possible when backup is complete, especially during office hours
- Test and monitor your backup daily
- Sync your backup to a second (remote!) location when possible

# Useful tips & pitfalls

- RAID is NO BACKUP!
- RAID tip: use RAID 1 (mirroring) when possible + choose a disk format that you can read on every type of (server) board, e.g. Intel ESRT II
- If you use RAID 5 or 6: you ALWAYS need a spare controller!
- Test your RAID controller before going into production! Use ^INTEGRIT to test your CACHE.DAT after copy
- Never use your backup disk from a client (NAS) pc (ransomware)
- USB backup HDD connected to the server: NEVER share it on the LAN (ransomware)
- NAS device: let your backup script connect & disconnect using a dedicated backup user – make sure your NAS backup share is READ-ONLY for all other users except the backup user! Never use the backup user account on a client pc
- Your main backup should use add & update files mode – DON'T sync (ransomware) – you can always sync your backup manually (!) regularly – never schedule sync's automatically!

# Useful tips & pitfalls (cont'd)

- Deploy & update a reliable system security solution on your server and all pc clients (anti-virus, system watcher, network monitoring, ...)
- Don't allow other devices to connect to your LAN without enforcing your security policy
- Be very careful with BYOD!
- Use a good spam filter (SpamHaus + Barracuda)

# Creating a backup (script)

- Must be able to backup the whole server
- Two options:
  - Hot, Caché online backup (+ external backup application) or custom backup script
  - Cold, offline backup (using external backup application or custom backup script)
- Caché backup options:
  - Use built-in Caché (built-in) online backup (HOT, ONLINE)
  - Freezing WD using CACHE.DAT file copy (HOT, ONLINE)
  - Stop Caché for backup and CACHE.DAT file copy (COLD, OFFLINE)
- Which one to use depends on your situation and requirements!
- Backup method can increase your downtime (e.g. incremental backups)

# Creating a backup: external application

- Perfect for copying Caché online backup files
- In case you want to stop Cache and just copy CACHE.DAT, most applications will allow you to add a pre- and post-backup script to stop & start Caché
- Pre-backup script: stop Caché (e.g. on Windows):
  - `net stop Cache_c-_intersystems_cache`
- Post-backup script: start Caché (e.g. on Windows):
  - `net start Cache_c-_intersystems_cache`
- Pay attention that all commands in your script work synchronously (e.g. the « `sc` » command will not work on Windows)
- Don't forget to stop dependent services first (web servers, Node.js connections, ...) and restart them after restarting Caché first

# Creating a backup: custom script

- The perfect solution when Caché runs 24x7 and you need control over other services on your server – e.g. also works for open files in Office
- **WARNING: CACHE.DAT filetime is NOT UPDATED** as long as you don't stop Caché - ROBOCOPY doesn't work!
- Backup needs to be done properly using two key technologies:
  - disk snapshot technology (Windows ShadowCopy, Linux LVM snapshot)
  - freezing writing daemon in Caché
- **Steps:**
  - Freeze writing daemon using `##Class(Backup.General).ExternalFreeze()`
  - Check if WD is suspended using `##Class(Backup.General).IsWDSuspended()` (attention: limited to max. 10 mins!)
  - Create snapshot of volume where CACHE.DAT's are using [Win32\\_ShadowCopy](#) or [LVM snapshot](#) in OS
  - Thaw writing daemon using `##Class(Backup.General).ExternalThaw()`
  - Backup now the drive's snapshot (can take a long time)
  - When backup is complete, remove snapshot in OS

# Creating a backup: custom script (example)

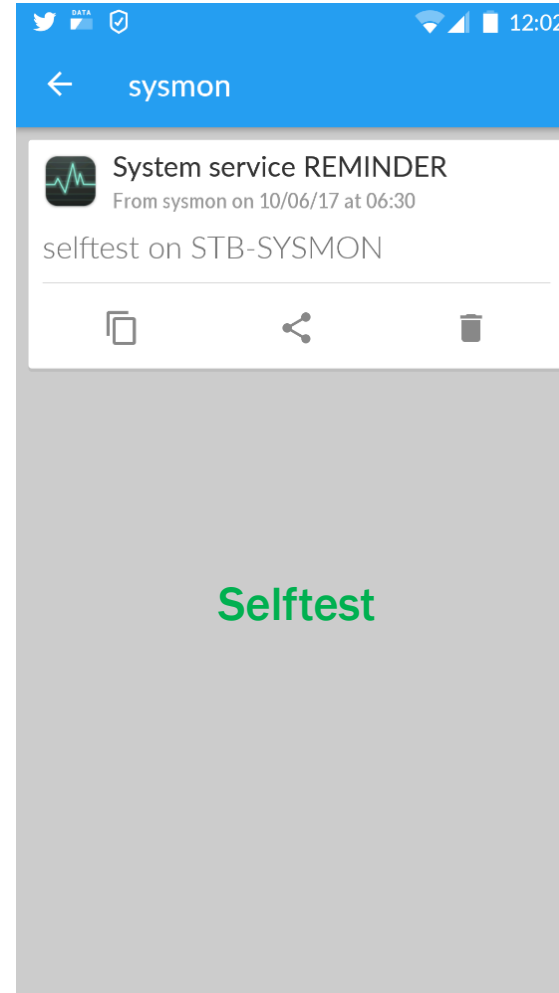
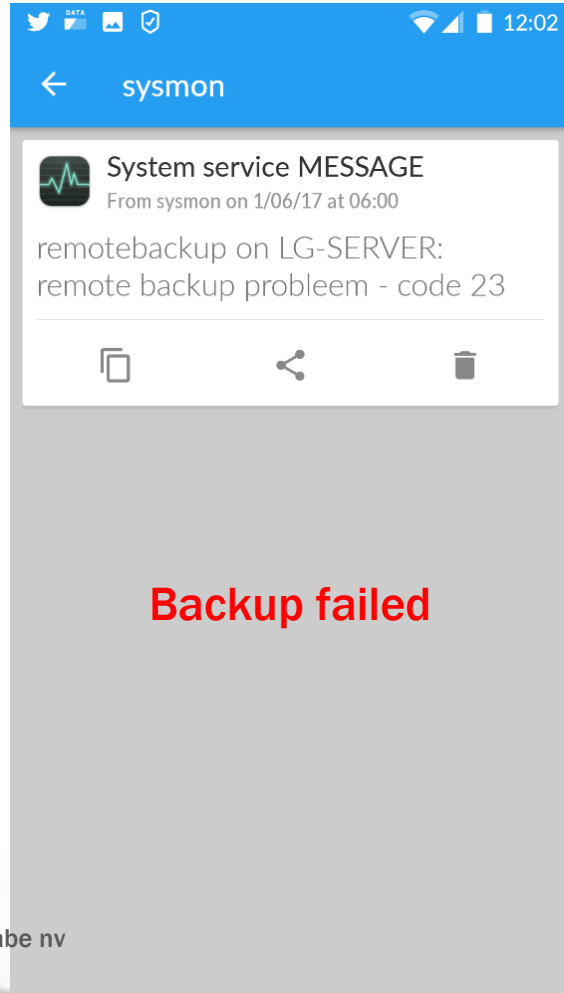
- On Windows: backup using a Node.js script + uses PowerShell and standard commandline file copy
- Advantages:
  - complete control over backup
  - Consistent backup using snapshots, Caché WD is only frozen for a very short time
  - backups complete server with other (dependent) services too
  - verifies backup copy immediately against current snapshot
  - Has proper failure checks (and returns results as JSON)
  - When disaster strikes you can restore very fast, just copy CACHE.DAT (+ restore journal files)
- Disadvantage: you have to write it ... 😊
- Uses Node.js for complete control over Caché (and other services if needed), allows very easy integration with online monitoring services like PushOver
- Uses PowerShell (or bash) for easy access to OS features (shadow copy creation)



# Creating a backup: custom script (example)

- Demo time ...

# Creating a backup: custom script (example)



# Monitoring your backup

- Very important: ALWAYS monitor, verify & test your backups
- Integrity checks with e.g. file comparison tools and Caché ^INTEGRIT
- Restore your backup on a another machine and verify it
- Always check for backup failures
- On failure: send e-mail, sms, message to monitoring apps (PushOver, Slack, ...)
- Don't forget to self-test your monitoring too!