The most important NetWorker 19.x Commands

1. General

Function	UNIX / Linux	Windows
Start the NetWorker daemons		
- All	**sysvinit Systeme** …/init.d/networker start **systemd Systeme** systemctl start networker	net start nsrd
- Only the NW client daemon		net start nsrexecd
- Then the server daemons		net start nsrd
Stop the NetWorker daemons		
- Only the Server daemons		net stop nsrd
- The client daemon (ehich will stop all others)	**sysvinit Systeme** …/init.d/networker start **systemd Systeme** systemctl stop networker	net stop nsrexecd
- alternatively	nsr_shutdown [-a] [-q]	

Function	UNIX / Linux	Windows
Start the NMC server	**sysvinit Systeme** …/init.d/gst start **systemd Systeme** systemctl start gst	net start gstd
Stop the NMC server	**sysvinit Systeme** …/init.d/gst stop **systemd Systeme** systemctl stop gst	net stop gstd

Function	UNIX / Linux	Windows
Login for a (new) session (default: 8 hrs)	nsrlogin [-s *server*] [-p *password*] [-d domain] -u *user*	nsrlogin [-s *server*] [-p *password*] [-d *domain*] -u *user*

Function	UNIX / Linux	Windows
Starting the NetWorker Admin GUI (NMC)	http://*nmc_server*:9000	http://*nmc_server*:9000
Star the NetWorker HTML5 GUI (NWUI)	https://*nmc_server*:9090/nwui	https://*nmc_server*:9090/nwui
Start the command line admin tool	nsradmin [-s *server*]	nsradmin [-s *server*]
Start the text-based monitoring tool	nsrwatch [-s *server*]	nsrwatch [-s *server*]

2. Backup Client Data

Function	UNIX / Linux	Windows
Start a client-initiated backup		
- Preview	`save [-s server] -n path`	`save [-s server] -n pfad`
- hot' run	`save [-s server] path`	`save [-s server] pfad`
- 'hot' run to a specific pool	`save [-s server] -b pool` `        path`	`save [-s server] -b pool` `        pfad`

- 'hot' run with setting specific browse- & retention policies	`save [-s server]`
	`   -w "browse_date"`
	`   -e "retention_date"`
	`   path`
		`save [-s server]`
		`   -w "browse_date"`
		`   -e "retention_date"`
		`   pfad`

Start a client-initated backup using client-direct	additional option `-a DIRECT_ACCESS=Yes`	additional option `-a DIRECT_ACCESS=Yes`
Start the backup GUI		`winworkr [-s server]` *Backup*

Function	UNIX / Linux	
Start a server-initiated backup (Policy)	nicht vorhanden	
- Preview	`nsrpolicy        ? -p "policy" -w "workflow"` `                [-c "client"]`	
- 'hot' run	`nsrpolicy    start -p "policy" -w "workflow"` `                [-c "client"]`	
- 'hot' restart of a 'group'	`nsrpolicy  restart -p "policy" -w "workflow"`	
Start a server-initiated backup (Policy)		

- Preview	`nsrworkflow     -p "policy" -w "workflow"`
	`                [-c "client"]`
	`                -A "'backup' -p"`	

- 'hot' run	`nsrworkflow     -p "policy" -w "workflow"`
	`                [-c "client"]`
	`                -A "'backup' -l level"`	

- Adhoc backup (NW 9.2.1+, overrules scheduled *skip* levels)	`nsrworkflow     -p "policy" -w "workflow"`
	`                [-c "client"]`
	`                -A "'backup' -l level"`
	`                -a`	

Using the NMC/Admin GUI		
- Start a 'Group'	*Monitoring > Policy > Workflow > Start*	
- Start an individual client	*Monitoring > Policy > Workflow > Start Individual Client...*	
- Restart a 'Group'	*Monitoring > Policy > Workflow > Restart*	

3. Restore Client Data

Function	UNIX / Linux	Windows
Start the recovery from the command line		
- Interactively	recover [-s *server*]	recover [-s *server*]

- a complete save set	recover [-s *server*]
	        -S *ssid*[/*cloneid*]
		recover [-s *server*]
		        -S *ssid*[/*cloneid*]

- a certain file of a save set	recover [-s *server*]
	        -S *ssid*[/*cloneid*]
	        -a *pfad*
		recover [-s *server*]
		        -S *ssid*[/*cloneid*]
		        -a *pfad*

- directed recovery of files	recover [-s *server*]
	        -c *source_client*
	        -R *dest_client*
	        -i [yYnNrR]
		recover [-s *server*]
		        -c *source_client*
		        -R *dest_client*
		        -i [yYnNrR]

- directed recovery of a save set	recover [-s *server*]
	        -R *dest_client*
	        -i [yYnNrR]
	        -S *ssid*
		recover [-s *server*]
		        -R *dest_client*
		        -i [yYnNrR]
		        -S *ssid*

Function	UNIX / Linux	Windows
Starten the recovery from the NetWorker User GUI		
- individual files		winworkr [-s *server*] *Recover*
- the whole save set		winworkr [-s *server*] *Operation - Save Set Recover*
- directed recovery		winworkr [-s *server*] *Operation - Directed Recover* *Operation - Save Set Recover*
Start the Recovery Managers	*Recover*	*Recover*

4. Job Query & Abortion

Function	UNIX / Linux	Windows
Query jobs from the jobs db	`jobquery`	`jobquery`
Query all active jobs (interactively)	`jobkill`	`jobkill`

Query a specific active jobs
 (interactively)
 	
 Known job types:
    archive job
    bootstrap save job
    BMR job
    Client push master job
    Client push client job
    Client push worker job
    clone job
    generic remote command
    index save job
    notification job
    probe job
    recover job
    save job
    savefs job
    savegroup job
    snapshot job
    synthetic full job
    task job
    utility job
    VSS job
	`jobkill -t "`*job type*`"`	`jobkill -t "`*job type*`"`

Stop a job
  Interactively
  Directly
	`jobkill`
	  jobid
	  `-j` *jobid*
		`jobkill`
		  jobid
		  `-j` *jobid*

5. Database Manipulation

Function	File-Index	Medien-Index
Query the size	`nsrls`	`nsrls -m`
Query the contents	`nsrinfo` `nsrinfo client`	`mminfo -q "query"` `       -r "report"`

Most important `mminfo reports`
 - backups within the last 24 hrs
 - … with additional information
 - all backups
 - media report
 - bootstrap report
 - total summary
		
		`mminfo`
		`mminfo -v`
		`mminfo -a [-v]`
		`mminfo -m [-v]`
		`mminfo -B [-v]`
		`mminfo -X [-v]`

Check the databases	`nsrck` `nsrck -L#`	`nsrck -X`

Delete file index information
 - of a save set
 - of a media (Volume ID)
 - of a media (Volume)
	
	`nsrmm -dP -S ssid[/cloneid]`
	`nsrmm -dP -V volid`
	`nsrmm -dP volume`
		
		`nsrmm -d -S ssid[/cloneid]`
		`nsrmm -d -V volid`
		`nsrmm -d volume`

Later change of the browse/ retention date of a save set		`nsrmm -w "browse_date"`
		`      -e "retention_date"`
		`      -S ssid`

Synchronize a AFTD volume with a AFTD/VTL media		`nsrmm -x    source_volume`
		`             dest_volume`
		`nsrmm -x -V source_volid`
		`             dest_volume`

Clone …
 - a single save set
 - a media (volume ID)
 - a media (volume name)
		
		`nsrclone -S ssid[/cloneid]`
		`nsrclone -V volid`
		`nsrclone -V volume`

Stage
 - a single save set
 - a volume
		
		`nsrclone -m -S ssid[/cl.id]`
		`nsrclone -m -V volume`

Change state of
 - a single save set
 	
 - a media (volume ID)
 - a media (volume)
		
		`nsrmm -o mode`
		`      -S ssid[/cloneid]`
		`nsrmm -o mode -V volid`
		`nsrmm -o mode volume`

5. Database Manipulation (continued)

Function	File-Index	Medien-Index
Add deleted db information of		

- a single backup
  (<u>and all</u> clones, if available
	`nsrmm -e new_ret_date`
	        `-S ssid[/cloneid]`
	`nsrmm -o notrecyclable`
	        `-S ssid[/cloneid]`
	`scanner -i[v][-F] -S ssid`
	          `device`
		`scanner -m[v][-F] -S ssid`
		          `device` **not useful**

- a media (volume name)	`scanner -i[v][-F] device`	`scanner -m[v][-F] device`
Verify the last bootstrap on a backup media		`scanner -B device`
Verify all bootstraps on a backup media		`scanner -Bv device` `[> file] [2>&1]`
Restore the deleted file index information to a speific time	`nsrck -t time client`	
Restore the last file index information - for a single client - for all clients	`nsrck -L7 client` `nsrck -L7`	
Restore a bootstrap (disaster recovery)		`nsrdr [-N] -[-NF]`

6. Hardware Verification Commands

Function	UNIX / Linux	Windows
Listi all SCS devices **(OS dependent)**	`cd /usr/sbin`	
- all	`./inquire`	`inquire`
- Only tape drives		`tapes`
- Only autochangers	`./changers`	`changers`
Execute a destructive short test for a tape device	`./tapeexercise [-v] drive`	`tapeexer [-v] drive`

Execute a destructive performance test for a tape device	`./tape_perf_test -f drive`
	`  [-t total_test_size]`
	`  [-x max_block_size]`
		`./tape_perf_test -f drive`
		`  [-t total_test_size]`
		`  [-x max_block_size]`

7. Jukebox Control Commands

Function	File-Index	Medien-Index
Inventory a jukebox	`nsrjb -HE`	`nsrjb -HE`
Query the jukebox' database	`nsrjb -C`	`nsrjb -C`
Diesplay the last 100 commands	`nsrjb -h`	`nsrjb -h`
Inventory all media in all slots	`nsrjb -I`	`nsrjb -I`
Inventory all media in specifc slots	`nsrjb -I -S from#-to#`	`nsrjb -I -S from#-to#`

Laden a media
  - using the volume name
  - using the slot number
  - using a specifiv device
	`nsrjb -l volume`
	`nsrjb -l -S #`
	`nsrjb -l -S # -f device`
		`nsrjb -l volume`
		`nsrjb -l -S #`
		`nsrjb -l -S # -f device`

Label a media
 - for a specific pool
 - for a specific pool
  with a specific name
	`nsrjb -L -S # -b pool`
	`nsrjb -L -S # -b pool`
	`    volume_name`
		`nsrjb -L -S # -b pool`
		`nsrjb -L -S # -b pool`
		`    volume_name`

Label multiple media - for a specific pool	`nsrjb -L -S from#-to#` `    -b pool`	`nsrjb -L -S from#-to#` `    -b pool`

Recyclen
  - a single media
 	
  - specific slots
	`nsrjb -R -S #`
	`nsrjb -R volume`
	`nsrjb -R -S from#-to#`
		`nsrjb -R -S #`
		`nsrjb -R volume`
		`nsrjb -R -S from#-to#`

Deposit media from the CAP into the jukebox	`nsrjb -d -S from#-to#` `      -P from#-to#`	`nsrjb -d -S from#-to#` `      -P from#-to#`
Withdraw media from the jukebox into the CAP	`nsrjb -w -S from#-to#` `      -P from#-to#`	`nsrjb -w -S from#-to#` `      -P from#-to#`

8. `nsr_render_log` Options

Filter Parameter	
`-A` *activityID*	Only messages with this *activityID*(s)
`-B` *number*	Only messages starting with this *number*
`-C` *category*	Only message of this *category* (i.e. RAP)
`-E` *end time*	**No** messages later than *end time*
`-F` *device*	Only messages of this *device*
`-G` *group*	Only messages refering to this *group(s)*
`-H` *hostname*	Only messages created by *hostname*
`-J` *hostname*	Only messages with reference of *hostname*
`-L` *locale*	Translate massages to this *locale* language
`-M` *messageID*	Only messages with this *messageID*
`-N` *lines*	Output of this number of *lines*
`-O` *program*	Only messages with reference of *program*
`-P` *process id*	Nur Meldungen mit Bezug zu *processID(s)*
`-R` *rhostname*	Reads the `*.raw` file from another *hostname*
`-S` *start time*	**No** messages earlier than *start time*
`-T` *threadID*	Only messages with this *threadID(s)*
`-Y` *severity*	Only messages of at leats this severity

Output Parameter	
`-a`	No output of the *activityID*
`-c`	No output of the *category*
`-d`	No output of the *time*
`-e`	No output of the *error number*
`-h`	No output of the *hostname*
`-l`	Print a head line
`-m`	No output of the *messageID*
`-o`	No output of the *program name*
`-p`	Keine Ausgabe der *processID*
`-r`	For another hostname (use with option `-R`)
`-t`	No output of the *threadID*
`-y`	No output of the *severity*
`-z`	Disguises important information (hostnames, user names and IP Adressess)