

Go utility

Originally created to assist the NetYCE developers to jump between the various directories of the application when using the bash CLI, the **go** utility offers sufficient benefits for the application manager to warrant some explanation.

The go utility is loaded automatically when logging in as **yce** user but can also be sourced by any bash shell from `/opt/yce/system/go`

Jumping directories

To use the utility, simply use it as a command with one or two arguments: `go <cmd>`. There are several reserved commands that can be used to manipulate YCE flag files and processes, but anything else is taken as a partial directory name. If that name matches an existing directory within the YCE application, a `cd` to that locations is executed.

for example, the user jumps first to the 'netyce 'system' directory, then to the 'logs' directory, and finally back to his previous directory (being the system dir):

```
$ go sys
goto sys -> /opt/yce/system
yce@yceseven /opt/yce/system
$

$ go logs
goto logs -> /var/opt/yce/logs
yce@yceseven /var/opt/yce/logs
$

$ go -
goto /opt/yce/system
/opt/yce/system
yce@yceseven /opt/yce/system
$
```

The jumps are initially tried as case sensitive but failing a match are attempted as case INsensitive:

```
$ go YCE
goto YCE -> /opt/yce/YCE
yce@yceseven /opt/yce/YCE
$ go yce
goto yce -> /opt/yce/htdocs/angular/assets/js/yce
yce@yceseven /opt/yce/htdocs/angular/assets/js/yce
$ go nms
goto nms -> /var/opt/mysql/NMS
yce@yceseven /var/opt/mysql/NMS
$
```

To optimize performance, the available directories are loaded on the first use of the go jump facility. To allow jumping to the more dynamically created job directories using the job ID, the available directories are reloaded when the typed name is longer than 8 characters.

The example below demonstrates the initial jump, a subsequent jump and a jump to a job ID:

```
$ go etc
loading paths...
goto etc -> /opt/yce/etc
yce@yceseven /opt/yce/etc
$ go logs
goto logs -> /var/opt/yce/logs
yce@yceseven /var/opt/yce/logs
$ go 0915_0053
loading paths...
goto 0915_0053 -> /var/opt/yce/jobs/0915_0053
yce@yceseven /var/opt/yce/jobs/0915_0053
$
```

Special jumps

As illustrated in the go - of the first example, there are some jumps with special meaning.

- The - location will jump to the previous jump or cd you used.
- The . location will jump to /opt/yce.
- The **xjob** will jump you to the most recent (highest) job ID directory.

Application processes

“Go” also has some process reporting and manipulation facilities that are specific to the YCE application.

go ps

The Unix standard 'ps' command is very versatile and powerful, but is hard to use and requires in-depth knowledge of all the application's processes to find out what is running properly or what processes are missing.

The **go ps** command creates a very NetYCE specific overview of the various daemons and processes, but also reports on processes that are missing and flag status. The output is similar to 'ps':



It shows all processes running under the yce user or are part of the NetYCE application. The column FNAME has the name of the process which will often be different from the command name. Indentation of the FNAME also indicates the parent - child relationships.

NetYCE daemon processes have their FNAME printed in blue.

At the bottom of the list, any missing processes are listed in red. Any process-ignore flags-files found are also listed here. The example above demonstrates that the yce_nccmd daemons is missing and that the process will not be restarted by the yce_psmon (process monitor daemon) because the flag file for the nccmd process exists.

go down

The **go down** command shuts the entire NetYCE application down. All daemons are killed and when done, the go ps results are displayed.

```
$ go down
halting NetYCE
stop 'psmon'
Stopping daemon 'yce_psmon'
stop 'mojo'
Stopping daemon 'mojo'
stop 'morbo'
Daemon 'morbo' not running
stop 'httpd'
Stopping httpd: [ OK ]
stop 'yce_* daemons'
stop 'vsftpd'
Shutting down vsftpd: [ OK ]
stop 'mysql'
Shutting down MySQL. SUCCESS!
UID          PID    PPID    %MEM   %CPU   START      FNAME      CMD
yce          6833   6831    0.09   0.02   08:07:10   sshd      sshd:
yce@pts/1
yce          6834   6833    0.12   0.02   08:07:10   bash      -bash
           yce_psmon      not running
           mysqld_safe  not running
           mysqld      not running
           httpd      not running
           vsftpd      not running
           yce_tftpd    not running
           yce_sched    not running
           mojo      not running
           yce_xch     not running
           yce_nccmd   not running
           yce_nccmd   ignore flag '/opt/yce/etc/ignore_nccmd' is set
           yce_skulker not running
```

go up

The **go up** command is used to launch the entire applications after the 'down' was issued. It removes all process ignore flags beforehand.

restart

The **go restart** <proc> is used to stop and immediately restart the NetYCE daemon process indicated with '<proc>'.

The valid values for '<proc>' are:

- **'mysql'** (will use sudo)
- **'httpd'** or **'yce_httpd'** (will use sudo)
- **'psmon'** or **'yce_psmon'** (will use sudo)
- **'vsftpd'** (if installed; will use sudo)
- **'skulker'** or **'yce_skulker'**
- **'xch'** or **'yce_xch'**
- **'nccmd'** or **'yce_nccmd'**
- **'sched'** or **'yce_sched'**
- **'tftp'** or **'yce_tftp'**
- **'ibd'** or **'yce_ibd'** (if the Infoblox integration is licensed)

The `go restart` command may be abbreviated to `go res`.

NOTE: some restart commands will under the current implementation not function properly. This limitation will be resolved in version 7.1.

Application flags

NetYCE uses within its application two types of application-flags that control process behaviour. These are the **debug flags** and the **ignore flags**.

Both types use the 'go' commands **go set** <flag> and **go clear** <flag>. The 'set' creates the flag file, the 'clear' removes it. The 'clear' command may also be invoked as **go clr** <flag>.

Debug flags

The two debug flags are:

- **debug**
This flag file controls the debug logging of *all* NetYCE daemons. The daemons will detect the debug mode at startup (requiring restart), although some will do this dynamically. The debug flag file corresponds with `/opt/yce/etc/yce_debug`. When present, the debug mode is enabled.
- **dev**
This flag controls the (very) extensive (development) logging of the vendor modules that are used to communicate with the network devices. It should not be activated on servers unless requested by NetYCE support.

The debug flag file corresponds with `/opt/yce/etc/yce_development`. When present, the debug mode is enabled.

The `go set dev` and `go clear dev` command affects *both* the debug and the development flags.

When the **debug** flag is set, the following files are used:

- `yce_auth_debug.log`
- `yce_db_debug.log`
- `yce_morbo_debug.log`
- `yce_nccm_debug.log`
- `yce_st_debug.log`
- `yce_cgi_debug.log`
- `yce_skulker_debug.log`
- `yce_tftp_debug.log`

When the **development** flag is set, the following file is used:

- `yce_dev_debug.log`

All the log files are located in `/var/opt/yce/logs`.

Ignore flags

The **ignore** flags control the NetYCE process monitoring for the various daemons. The monitoring is performed by the **yce_psmon** daemon which should be running at all times to ensure proper application behaviour.

Monitoring by `yce_psmon` ensures that the process is behaving within some constraints for cpu usage and number of child processes. When ill-behaved the process is killed and re-spawned. Should a process die by itself or be killed by a user, the process is likewise re-spawned. The parameters for each process it must monitor are specified in the configuration file `/opt/yce/etc/<servername>_psmon.conf`. This file is automatically created when installing a version (update).

The `yce_psmon` has a unique flag file configured for each of its daemons. When the flag file is present it indicates the `yce_psmon` should NOT monitor this process. To stress this behaviour, the flag file is named after the process name prepended by "ignore_".

The ignore flags can be controlled using the **go set <flag>** and **go clear <flag>** commands. The values for <flag> and their corresponding files are:

- **'mysql'** sets `/opt/yce/etc/ignore_mysql`
- **'httpd'** sets `/opt/yce/etc/ignore_httpd`
- **'skulker'** sets `/opt/yce/etc/ignore_skulker`
- **'xch'** sets `/opt/yce/etc/ignore_xch`
- **'mojo'** sets `/opt/yce/etc/ignore_mojo`
- **'sched'** sets `/opt/yce/etc/ignore_sched`
- **'nccmd'** sets `/opt/yce/etc/ignore_nccmd`
- **'tftpd'** sets `/opt/yce/etc/ignore_tftpd`
- **'ibd'** sets `/opt/yce/etc/ignore_ibd`

Log file shortcuts

NetYCE support and developers frequently examine (debug) log files. Some specific log-file access shortcuts were included in the 'go' utility to simplify these tasks.

These shortcuts are only useful if the **debug** and/or **development** flags are set.

Database logging

- **go db**
Opens 'less' on the yce database debug file `/var/opt/yce/yce_db_debug.log`.
Use 'q' to exit, 'G' to goto last entry, '/<pat>' to search forward, '?<pat>' to search backwards.
- **go tdb**
Opens 'tail -f' on the yce database debug file.
- **go cdb**
Clears the yce database debug file by `cp /dev/null /var/opt/yce/yce_db_debug.log`.

Development logging

- **go dev**
Opens 'less' on the yce development debug file `/var/opt/yce/yce_dev_debug.log`.
Use 'q' to exit, 'G' to goto last entry, '/<pat>' to search forward, '?<pat>' to search backwards.
- **go tdev**
Opens 'tail -f' on the yce development debug file.
Use '<control>-C' to exit.
- **go cdev**
Clears the yce development debug file by `cp /dev/null /var/opt/yce/yce_dev_debug.log`.

Development compare

A special command was included to compare two files side by side *on different servers*. This is used mostly to find code changes between different versions.

The command uses `vimdiff` to show the files side by side. Refer to Vim manuals to learn how to operate 'vi' or 'vim' in this mode.

The **go cmp <file>** can only be invoked when <file> resides in the current directory and is present on the local server AND the server against which to compare.

This remote server must be defined in the environment parameter RDEV. It is set in `/home/yce/.bash_profile`:

```
if [ -r "/opt/yce/system/go" ]; then
    export RDEV="ycesix"
```

```
    source "/opt/yce/system/go"  
else  
    echo "Skipping 'go'"  
fi
```

From:

<https://wiki.netyce.com/> - **Technical documentation**

Permanent link:

https://wiki.netyce.com/doku.php?id=maintenance:general:tools:go_utility

Last update: **2024/07/03 12:31**

