# ctys-uc-CLI(7)
# Use-Cases for CLI

May 19, 2010

## Contents

.

# 1 General

The CLI plugin itself is a pure command line interface, but due to the default activation of X11 forwarding any X11 command could be executed within a CLI session, thus any of the following examples could be used as a XTerm starter.
E.g. the interactive call of "xclock" will display correctly. This is particularly also true for the whole login-chain, when CLI is used for cascaded logins.
So, basically any firewall could be pierced in a secure manner by an SSH gateway in the DMZ. Which depends on the security facility of the gateway itself, of course.
**HINT**: Spaces within options, including suboptions, have to be masked by the reserved character '%'. This means just replace any SPACE with a '%' within any options/suboptions. The '%' itself is provided as '%%'.

# 2 Start a Local Interactive Session

This opens a second shell executed as login, almost the same as an ordinary shell call.

```
ctys -t CLI -a create=l:test
```

The "localhost" is hard-coded to behave as sub-shell call too.

```
ctys -t CLI -a create=l:test localhost
```

**REMARK**: Due to the implemented ambiguity-check for uniqueness of LABELs, only one localhost session is supported by the same label, when the label has to be non-altered, the usage of "-A 1" disables ambiguity-checks.

# 3 Start a Remote Interactive Session

This opens a second shell as a remote executed login.

```
ctys -t CLI -a create=l:test lab00
```

# 4 Execute a Remote Command

This opens a remote shell and executes the provided command only before termination. The connection will be kept open during the whole session, thus this is not executed in background mode by default.

```
ctys -t CLI -a create=l:test,cmd:uname\%-a lab00
```

The same forced to perform in background mode.

```
ctys -t CLI -a create=l:test,cmd:uname\%-a -b 1 lab00
```

# 5 Execute Multiple Remote Commands

The full scope of addressing of ctys is supported, thus the addressing of multiple targets, where each target could be a single host of a preconfigured hosts-group, is applicable. Intermixed addressing is supported too.

```
ctys -t CLI -a create=cmd:uname\%-a lab00 lab01
```

The same with parallel background execution:

```
ctys -t CLI -a create=cmd:uname\%-a -b 1 lab00 lab01
```

or

```
ctys -t CLI -a create=cmd:uname\%-a <host1> <group1> <host2> <group2>
```

The full scope of "include" files for group definitions and macros is applicable, thus e.g. tree-like reuse of groups could be applied.
Beginning with the current version path-based addressing of groups is supported, this allows for addressing like:

```
ctys -t CLI -a create dir0/subdir1/sbudir2/group-file
```

In combination with the enhanced features of **ctys-groups** for tree-views this allows for management of structured groups. Typical applications are the management of task-based destops for office applications, development environments, and the management of test-cases for Major-Projects.
Due to security reasons root-permission should be configured and handled properly, of course. It might be recognized that there is currently a chance (?) for users with appropriate skills and permissions to intercept the communications, when on the intermediate hops the message flow has to be re-encrypted after decryption.
The following call opens a session hop1 to lab01 via the intermediate relay lab00 by the session hop0.

```
ctys -t cli -a create=l:hop0cmd:ctys\%-t\%cli\%-a\%create=l:hop1\%lab00 lab01
```

The following call opens a session hop1 to lab01 via the intermediate relay lab00 by the session hop0 and starts a Xterm on lab01.

```
ctys -t cli -a create=l:hop0cmd:ctys\%-t\%cli\%-a\%create=l:hop1,cmd:xterm\%lab00 lab01
```

This approach is very similar to the equivalent usage of OpenSSH, and could be used in same manner to bypass routing as well as firewalls, when access and execution permissions on gateways are available.

# 6 Start Xterm with tcsh

This call starts an interactive XTerm session running tcsh inside.

```
ctys -t cli -a create=l:tstcall,s:tcsh\%-c,cmd:xterm\%-e\%tcsh -b 1 lab00
```

# 7 Start gnome-terminal

This call starts an interactive gnome-terminal session running tcsh inside.

```
ctys -t cli -a create=l:tstcall,s:tcsh\%-c,cmd:gnome-xterminal\%-e\%tcsh -b 1 lab00
```

# 8 SEE ALSO

*ctys(1)*, *ctys-CLI(7)*

# 9 AUTHOR

Written and maintained by Arno-Can Uestuensoez:


| | |
|---|---|
| Maintenance: | <acue_sf1@sourceforge.net> |
| Homepage: | <http://www.UnifiedSessionsManager.org> |
| Sourceforge.net: | <http://sourceforge.net/projects/ctys> |
| Berlios.de: | <http://ctys.berlios.de> |
| Commercial: | <http://www.i4p.com> |


# 10 COPYRIGHT