

# ctys-**CLI**(1)

## Command Line Interface

September 29, 2020

### Contents

<b>1</b>	<b>NAME</b>	<b>2</b>
<b>2</b>	<b>SYNTAX</b>	<b>2</b>
<b>3</b>	<b>DESCRIPTION</b>	<b>2</b>
<b>4</b>	<b>OPTIONS</b>	<b>3</b>
<b>5</b>	<b>SEE ALSO</b>	<b>5</b>
<b>6</b>	<b>AUTHOR</b>	<b>5</b>
<b>7</b>	<b>COPYRIGHT</b>	<b>5</b>

## 1 NAME

ctys-CLI - Command Line Interface

## 2 SYNTAX

```
ctys -t CLI -a action[=<suboptions>] ...
ctys -T CLI -a action[=<suboptions>] ...
ctys -T ALL -a action[=<suboptions>] ...
```

## 3 DESCRIPTION

The CLI-plugin starts a remote shell from within a local shell. No specific default desktop functionality such as XTerm is supported, just a pure CLI access is performed. The handling of desktop windows such as XTerm or GTerm is supported by the X11-Plugin. The default and standard usage of the X11 displayforwarding by activation of the **'-X'** option of the underlying OpenSSH provides the usage of the CLI-plugin as a scripting and automation starter for any X11 application. This provides particularly any application with a proven on-demand encryption for the network interconnection by seamless integration into the **ctys** user interface. An Xterm, gnome-terminal or any arbitray X11 tool could be started from within a CLI sub-option "CMD". Chained logins by gateways with Overall-Display-Forwarding is supported by OpenSSH, thus by ctys too.

When executing CLI almost the same functionality as for an ordinary SSH session is supported. The main advance of using ctys instead of an ordinary ssh-call is the seamless integration into the ctys environment. Resulting from this feature instead of an IP address the `<machine-address>` could be used. Thus the administration of address-to-target-mapping could be completely delegated to ctys and will be utilized by internal usage of the full scope of ctys-vhost. E.g. the **LABELS** defined within ctys could be used to open a remote shell to any "labeled" instance such as a VM or PM. Therefore internal CLI type CONSOLES only use the CLI plugin.

Another quite important aspect is the persistancy of any symbolic address entity. The complete set of addresses defined by symbolic elements such as labels is mapped to actual runtime addresses when required. Thus the automation of post-attachement of an X11 application could be easily stored into a script without additional effort for determination of the dynamic runtime identifier such as arbitrary display identifiers.

Another option might be the usage of UUIDs or MAC-Addresses for persistent definition of the handling of commands on frequently changing and reinstalled systems with reassigned IP-Addresses and DNS names. This is provided for any system by dynamic mapping.

The second important feature of CLI is the facility to start native remote commands as shell calls within any managed PM or VM instance. Therefore CLI is the working horse for native execution of GUI-less ctys-tasks. Similar to the call of a system command, any internal ctys library and plugins function could be called remotely by CLI sub-option "CMD".

The default shell used on the target instance is bash, which could be altered interactive by the sub-option "SHELL|S", or persistently by the variable `CLI_SHELL_CMD_DEFAULT`.

It should be mentioned and explained here that there are two quite similar suboptions, which on the first view seems to be redundant, which is actually not the case. The main **difference between** the **CMD** and **SHELL** suboptions is the behaviour. Whereas the SHELL just overlays the standard shell within a Session, the CMD suboption executes a command from within the executed shell - either the standard shell, or a pre-set custom shell.

Additional information containing use-cases with application examples is available from **ctys-uc-CLI**.

## 4 OPTIONS

-a action[=<suboptions>]

CANCEL

Not applicable.

CREATE

```
CREATE=[<machine-address>]
      [REUSE|CONNECT|RECONNECT|RESUME]
      [CONSOLE:<>]
      [(CALLOPTS|C):<callopts>]
      [CD:<working-directory>] [,]
      [(XOPTS|X):<xopts>]
      [(SHELL|S):<shell>]
      [(STUBMODE|STUB)[:on]] [,]
      [(CMD):<cmd>]
```

<machine-address>

See standard for syntax. Due to the limited attribute set of a completely dynamic CLI session without own persistent resources only the LABEL is applicable. The hosting OS has to be handled by it's own plugin. The LABEL sub-option is here mandatory.

CALLOPTS

Refer to common options description .

CD:<working-directory>

Change current working directory for the started command. This provides for example the pre-configuration of specific work-sets, where each terminal could be opened for a specific directory.

CMD:<cmd>

Refer to common options CMD description.

CONSOLE

Not yet supported.

<xopts>

Refer to common options xopts description.

REUSE|CONNECT|RECONNECT|RESUME

Not applicable.

(SHELL|S):<shell>

Refer to common options description .

A specific exception to the synchronous execution style of a CLI shell occurs, when multiple execution instances are addressed by the arguments of current call. When the "-b off" option for background a.k.a asynchronous operations is selected, the standard synchronous foreground operation works quite well. The list of arguments is just executed sequentially as expected. The CLI CREATE action sets implicitly the forced usage of a pseudotty. This is the same as using "-z pty,pty".

(STUBMODE|STUB)[:on]

The STUBMODE suppresses the remote execution of the full ctys set, instead just a remote shell by a simple SSH call is executed. This could be utilized particularly for sessions to machines without an installed ctys.

ENUMERATE

Not applicable.

#### LIST

Almost the same output as common standard, with following changes in semantics.

**id** The PID of the local SSH termination point, which is the locally executed relay-instance of ctys and is the parent shell of actual running batch/interactive shell. The CLI plugin does not support a cancel or connect|reconnect|reuse action, so the "id" is here a non-functional hint only.

**pid** PID of current ctys sessions top.

The following values are not applicable: uuid, mac, dsp, cp, sp

-L (LOCALONLY|LO) | (DISPLAYFORWARDING|DF)

Even though a DISPLAY will actually not be used, this parameter also adopts the behavior of the execution to a providing environment, which is frequently for ctys true.

These are the only location parameters to be applied.

## 5 SEE ALSO

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

## 6 AUTHOR

Arno-Can Uestuensoez <<https://arnocan.wordpress.com/>>  
<<https://unifiedsessionsmanager.sourceforge.io/>>  
<<https://github.com/unifiedsessionsmanager>>



## 7 COPYRIGHT

Copyright (C) 2008, 2009, 2010, 2011, 2020 Ingenieurbuero Arno-Can Uestuensoez  
This is software and documentation from **BASE** package,

- for software see GPL3 for license conditions,
- for documents see GFDL-1.3 with invariant sections for license conditions.  
The whole document - all sections - is/are defined as invariant.

For additional information refer to enclosed Releasenotes and License files.

