



<http://xyzcmd.syhpoon.name/>

Created 04.05.2010

Main features:

- * Tight integration with python run-time system — most of the settings can be changed "on the fly" via management console.
- * Powerful configuration system - define own actions, aliases, internal commands, key bindings.
- * Extensible plug-in system - even core functionality implemented mainly using plug-ins, keeping base system small and clean.
- * Events & hooks subsystem - a flexible way of reacting on certain system events.
- * Customizable look-n-feel - every widget component look can be changed using skins.
- * Unicode support

Tabs

(0 /) (1 UNIX) (2 usr)

```
./
/bin
/boot
~/cdrom
/dev
/etc
/home
/lib
/media
/mnt
/opt
/proc
/root
/sbin
/selinux
/srv
/sys
/tmp
/usr
/var
@initrd.img
@initrd.img.old
@vmlinuz
@vmlinuz.old
```

Active panel

(0 dev)

/dev

```
./
./initramfs
./udev
/block
/bsg
/bus
/char
/disk
~fd
/input
/mapper
/net
/pts
/shm
/snd
/syhoon
/v4l
./initramfs-tools
@MAKEDEV
-adsp
-audio
-autofs
@cdrom1
@cdrom1
-console
@core
-cpu_dma_latency
-dsp
@dvd1
```

Inactive panel

/ 4K drwxr-xr-x

/ 4K drwxr-xr-x

||

Prompt

Command line

Skins

Skin is the way to customize look'n'feel of XYZCommander.

Customizable elements are:

- * Elements of filesystem based on fsrule description:
- * All user interface widgets

XYZCommander comes with 5 standard skins:

Seablue (default)

```
(0 /) {1 UNIX} {2 usr} (0 dev) /dev  
/.. /bin  
/boot /dev  
~cdrom /etc  
/home /lib  
/media /mnt  
/mnt /opt  
/opt /proc  
/proc /root  
/root /sbin  
/sbin /selinux  
/selinux /srv  
/srv /sys  
/sys /tmp  
/tmp /usr  
/usr /var  
/var @initrd.img  
@initrd.img @initrd.img.old  
@initrd.img.old @vmlinuz  
@vmlinuz @vmlinuz.old  
@vmlinuz.old  
/ 4K drwxr-xr-x  
/.. /dev  
/.initramfs  
/.initramfs /udev  
/udev /block  
/block /bsg  
/bsg /bus  
/bus /char  
/char /disk  
/disk ~fd  
~fd /input  
/input /mapper  
/mapper /net  
/net /pts  
/pts /shm  
/shm /snd  
/snd /syhpoon  
/syhpoon /v4l  
/v4l .initramfs-tools  
.initramfs-tools @MAKEDEV  
@MAKEDEV -adsp  
-adsp -audio  
-audio -autofs  
-autofs @cdrom1  
@cdrom1 @cdrom1  
@cdrom1 -console  
-console @core  
@core -cpu_dma_latency  
-cpu_dma_latency -dsp  
-dsp @dvd1  
@dvd1  
/ 4K drwxr-xr-x
```


grass

```
{0 /} {1 UNIX} {2 usr}                               {0 dev}
/
/..
/bin
/boot
~cdrom
/dev
/etc
/home
/lib
/media
/mnt
/opt
/proc
/root
/sbin
/selinux
/srv
/sys
/tmp
/usr
/var
@initrd.img
@initrd.img.old
@vmlinuz
@vmlinuz.old

/ 4K drwxr-xr-x

||
```

```
/dev
/..
/.initramfs
/.udev
/block
/bsg
/bus
/char
/disk
~fd
/input
/mapper
/net
/pts
/shm
/snd
/syhpoo
/v4l
/.initramfs-tools
@MAKEDEV
-adsp
-audio
-autofs
@cdrom1
@cdrom1
-console
@core
-cpu_dma_latency
-dsp
@dvd1

/ 4K drwxr-xr-x
```

glamour

{0 /} {1 UNIX} {2 usr}

```
/.  
/bin  
/boot  
~cdrom  
/dev  
/etc  
/home  
/lib  
/media  
/mnt  
/opt  
/proc  
/root  
/sbin  
/selinux  
/srv  
/sys  
/tmp  
/usr  
/var  
@initrd.img  
@initrd.img.old  
@vmlinuz  
@vmlinuz.old  
  
/ 4K drwxr-xr-x
```

{0 dev}

```
/. /dev  
/.initramfs  
/.udev  
/block  
/bsg  
/bus  
/char  
/disk  
~fd  
/input  
/mapper  
/net  
/pts  
/shm  
/snd  
/syhpoon  
/v4l  
/.initramfs-tools  
@MAKEDEV  
-adsp  
-audio  
-autofs  
@cdrom1  
@cdrom1  
-console  
@core  
-cpu_dma_latency  
-dsp  
@dvd1  
  
/ 4K drwxr-xr-x
```

||

monochrome

```
{0 /} {1 UNIX} {2 usr} /dev
/..
/bin
/boot
~cdrom
/dev
/etc
/home
/lib
/media
/mnt
/opt
/proc
/root
/sbin
/selinux
/srv
/sys
/tmp
/usr
/var
@initrd.img
@initrd.img.old
@vmlinuz
@vmlinuz.old

/ 4K drwxr-xr-x

{0 dev} /dev
/..
/.initramfs
/.udev
/block
/bsg
/bus
/char
/disk
~fd
/input
/mapper
/net
/pts
/shm
/snd
/sylhpoon
/v4l
 .initramfs-tools
@MAKEDEV
-adsp
-audio
-autofs
@cdrom1
@cdrw1
-console
@core
-cpu_dma_latency
-dsp
@dvd1

/ 4K drwxr-xr-x
```


Lighty (256 high colors mode)

```
{0 /} {1 UNIX} {2 usr}
/
/..
/bin
/boot
~cdrom
/dev
/etc
/home
/lib
/media
/mnt
/opt
/proc
/root
/sbin
/selinux
/srv
/sys
/tmp
/usr
/var
@initrd.img
@initrd.img.old
@vmlinuz
@vmlinuz.old

/ 4K drwxr-xr-x

{0 dev}
/dev
/..
/.initramfs
/.udev
/block
/bsg
/bus
/char
/disk
~fd
/input
/mapper
/net
/pts
/shm
/snd
/syhpooon
/v4l
 .initramfs-tools
@MAKEDEV
-adsp
-audio
-autofs
@cdrom1
@cdrom1
-console
@core
-cpu_dma_latency
-dsp
@dvd1

/ 4K drwxr-xr-x
```

It is very easy to create your own skin theme!

Check the manual chapter for details:

<http://xyzcmd.syhpoon.name/static/doc/user-manual/skins.html>

FSRules

FSRules is a tiny DSL, widely-used in XYZCommander for describing filesystem objects.

FSRules are used for:

- * Describing objects in skins
- * Describing objects for custom actions
- * Tagging files in panel
- * More ...

Some FSRules examples

Tag all hidden files in current directory

```
(0 tmp) (1 UNIX) (2 usr) /tmp
/..
/.A
/.B
/.C
/.ICE-unix
/.X11-unix
/emacs1000
/home
/keyring-yPT40a
/orbit-syhpoo
/plugtmp
/ssh-pjyxzi1819
/svoftmp
.X0-lock
.a
.b
.c
.d
=OSL_PIPE_1000_Single0
=scim-helper-manager-s
=scim-panel-socket:0-syhpoo
=scim-socket-frontend-syhpoo

11 bytes (10) 6 drwxrwxrwt

(0 dev) /dev
/..
/.initramfs
/.udev
/block
/bsg
/bus
/char
/disk
~fd
/input
/mapper

Tag group
Type FS Rule
name{"^\\.({1}[^,]*)"
Press ENTER to submit value. ESCAPE to quit

-audio
-autofs
@cdrom1
@cdrom1
-console
@core
-cpu_dma_latency
-dsp
@dvd1
/ 4K drwxr-xr-x
```


Tag all files not owned by user syhpoon

```
(0 tmp) (1 UNIX) (2 usr) (0 dev)
/tmp /dev
/.. /..
/.A /.initramfs
/.B /.udev
/.C /block
/.ICE-unix /bsg
/.X11-unix /bus
/emacs1000 /char
/home /disk
/keyring-yPT40a ~fd
/orbit-syhpoon /input
/plugtmp /mapper
/ssh-pjyxzi1819
/svofc.tmp
.X0-lock
.a
.b
.c
.d
=OSL_PIPE_1000_Single0
=scim-helper-manager-s
=scim-panel-socket:0-syhpoon
=scim-socket-frontend-syhpoon
-11 bytes (4) 20 drwxr-xr-x / 4K drwxr-xr-x
11
```

Tag all files with execution bit set

```
(0 tmp) (1 UNIX) (2 usr) (0 dev)
/tmp /dev
/.. /..
/.A /.initramfs
/.B /.udev
/.C /block
/.ICE-unix /bsg
/.X11-unix /bus
/emacs1000 /char
/home /disk
/keyring-yPT40a ~fd
/orbit-syhpoo /input
/plugtmp /mapper
/ssh-pjyxi1819
/svofe.tmp
.X0-lock
.a
.b
.c
.d
=OSL_PIPE_1000_Single0
*exe1
*exe2
*exe3
=scim-helper-manager-socket-syhpoo
=scim-panel-socket:0-syhpoo
=scim-socket-frontent-syhpoo
0 bytes (3) 4K drwxr-xr-x
/svofe.tmp 4K drwxr-xr-x

Tag group
Type FS Rule
type(file) and perm(+0111)
Press ENTER to submit value, ESCAPE to quit

-audio
-autofs
@cdrom1
@cdrom1
-console
@core
-cpu_dma_latency
-dsp
@dvd1
/ 4K drwxr-xr-x
```

Tag those files which according to magic db match
as avi movies.

```
(0 tmp) (1 UNIX) (2 usr) (0 /)
/tmp /
/.. /bin
/.A /boot
/.B /cdrom
/.C /dev
/.ICE-unix /etc
/.X11-unix /home
/emacs1000 /lib
/home /media
/keyring-yPT40a /mnt
~link.good /opt
/orbit-syhpoo
/plugtmp
/ssh-pjyxi1819
/sv2fn.tmp
.X0-lock
.a
.b
.c
.d
289M
More-168
=OSL_PIPE_1000_SingleOfficeIPC_c1c6bb8665e225cf22a45146~
*exe1
*exe2
*exe3
file.txt
link.bad
=scim-helper-manager-socket-syhpoo
=scim-panel-socket:0-syhpoo
432_254_096 bytes (1)
/ssh-pjyxi1819 23 drwx----- / 4K drwxr-xr-x
```

Tag group

Type FS Rule

magic("avi")

Press ENTER to submit value. ESCAPE to quit

Tag all files with size exceeding 200 megabytes

```
(0 tmp) (1 UNIX) (2 usr) (0 /)
/tmp /
/.. /..
/.A /bin
/.B /boot
/.C ~/cdrom
/..ICE-unix /dev
/..X11-unix /etc
/emacs1000 /home
/home /lib
/keyring-yPT40a /media
~link.good /mnt
/orbit-syhpoo /opt
/plugtmp
/ssh-pjyxzi1819
.X0-lock
.a
.b
.c
.d
289M
More-1GB
*exe1
*exe2
*exe3
file.txt
link.bad
=scim-helper-manager-socket-syhpoo
=scim-panel-socket:0-syhpoo
=scim-socket-frontend-syhpoo

2_045_781_136 bytes (3)
/ssh-pjyxzi1819 23 drwx----- / 4K drwxr-xr-x
||
```

Tag group

Type FS Rule

size(">200M")

Press ENTER to submit value. ESCAPE to quit

More rule actions and description how to create your own FSRule are available in XYZCommander user manual:

http://xyzcmd.syhpoon.name/static/doc/user-manual/fs_rules.html

Events & hooks

One can create hooks reacting on internal system events

Events & hooks

Run load() method when system starts:

```
self.xyz.hm.register("event:startup", self.load)
```

Run save() method when system stops:

```
self.xyz.hm.register("event:shutdown", self.save)
```

Run handler before executing the command:

```
self.xyz.hm.register("event:core:shell:execute", self.handler())
```

A close-up photograph of a computer keyboard, focusing on the central keys. A semi-transparent white rectangular box is overlaid on the keyboard, centered horizontally and vertically. Inside this box, the text "Useful hints" is written in a black, sans-serif font. The background shows several keys with letters: 'T', 'Z', 'U' in the top row; 'F', 'G', 'H', 'J' in the middle row; and 'V', 'B', 'N' in the bottom row. The keys are white with dark lettering, and the keyboard's frame is visible in the background.

Useful hints

If something goes wrong:
logger window may help (META-I (lowercase L))

```
(0 /) {1 UNIX} {2 usr} (0 UNIX)
Logger console
[2010-04-29 23:57]: (ERROR) Error executing bind (<Shortcut: ['f5']>): Test ERROR!
[2010-04-29 23:57]: (DEBUG) Traceback (most recent call last):
  File "/home/syhpoon/UNIX/Work/Devel/Lang/python/Projects/xyzcmd/xyzcmd-trunk/libxyz/ui/panel.py", line 160, in
loop
    self._cmd.keypress(_dim, _input)
  File "/home/syhpoon/UNIX/Work/Devel/Lang/python/Projects/xyzcmd/xyzcmd-trunk/libxyz/ui/cmd.py", line 312, in
keypress
    return _meth()
  File "<string>", line 182, in x
RuntimeError: Test ERROR!
```

See active plugins exported methods, doc, events etc (META-t)

```
(0 /) (1 UNIX) (2 usr) (0 UNIX)
Active plugins list
:core:bindlist      0.2 - Show keybindings
:core:console       0.1 - Interactive management console
:core:keycodes      0.1 - Setup terminal keycodes
:core:pluginlist    0.2 - Show plugin list
:fsrules:magic      0.1 - Magic database FSRule
:misc:about         0.1 - About plugin
:misc:where         0.2 - Save panels locations
:sys:cmd            0.1 - Command line plugin
:sys:logger         0.1 - Logger plugin
:sys:panel          0.1 - Panel plugin
:sys:run            0.1 - Run plugin
:ui:bookmarks       0.1 - Frequently used directories
:ui:testinput       0.1 - Test input
:vfs:fileinfo       0.1 - Show VFS object information
:vfs:vfsutils       0.1 - Useful VFS routines
```