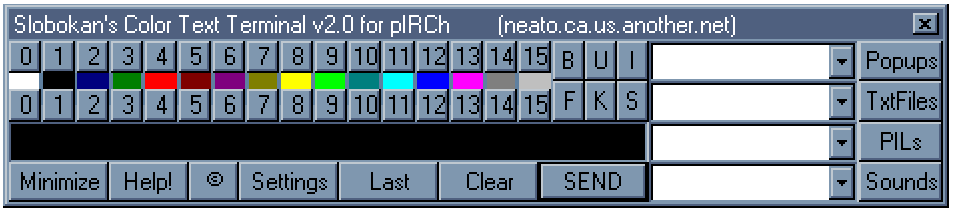
# Slobokan's
# Color Text Terminal
## version 2.0 (PIL) for pIRCh



Text Color Buttons
BackGround Color Buttons
Bold Attribute
Underline Attribute
Italics Attribute
Fixed Font Attribute
Control K
Symbol Font Attribute
Help Button
Editor Window
Settings
Last Button
Clear Button
Send Button
PopUps
TxtFiles
Pils
Sounds
Color Codes
Variables
**About CTERM**

## Text Color Buttons

Located in the top row of the CTERM interface, the text color buttons are used to insert the color code for the corresponding color of text. The corresponding color appears below the text color button. When clicking a text color button, the correct color code is inserted into the Editor Window. Any text entered after that code, will be displayed in that color, when sent to the channel. To reset the text color after entering your text, you can click the Control-K button or another text color button.

### Examples:

If you click text color 4 and type TEST, it will appear in the Editor Window as:

ž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

TEST

If you click text color 4 and type TESTING, click text color 3 and type AGAIN, it will appear in the Editor Window as:

ž4TESTING ž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

If you click text color 4 and type TESTING, click the Control-K button and type AGAIN, it will appear in the Editor Window as:

ž4TESTING žAGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

See also: BackGround Colors, Bold, Underline, Italics, Fixed Font, Symbol Font, Control-K.

## BackGround Color Buttons

Located on the left side of the CTERM interface, below the Text Color Buttons, the background color buttons are used to insert the color code for the corresponding background color. The corresponding color appears above the background color button. When clicking a background color button, the correct color code is inserted into the Editor Window. Any text entered after that code, will be displayed on that background color, when sent to the channel.

### Examples:
### (Note: I was unable to display the background colors in this section)

If you click text color 4 and background color 1, then type TEST, it will appear in the Editor Window as:

4,1TEST

When you click the Send Button the text is sent to the channel, and appears as:

TEST on a black background.

If you click text color 4 and background color 1, then type TESTING, click text color 3 and background color 4, and type AGAIN, it will appear in the Editor Window as:

4,1TESTING 3,4AGAIN

When you click the send button the text is sent to the channel, and appears as:

TESTING on a black background and AGAIN on a red background.

See also: Bold, Underline, Italics, Fixed Font, Symbol Font, Control-K.

## Bold Attribute

Located to the right of the Text Color Buttons, in the middle of the CTERM interface, the bold attribute button is used to apply a bold face to your text. When clicking the bold attribute button, the bold character is inserted into the editor window. Any text entered after that code, will be displayed as bold text, when sent to the channel. To reset the bold attribute after your text, you can click the bold attribute button again.

(Note: You MUST insert the BOLD Attribute before the color code or pIRCh will strip the color code.)

### Examples:

If you click the bold attribute and text color 4 then type TEST, it will appear in the Editor Window as:

žž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

**TEST**

If you click the bold attribute and text color 4 then type TESTING, click the bold attribute again then type AGAIN, it will appear in the Editor Window as:

žž4TESTING žAGAIN

When you click the Send Button the text is sent to the channel, and appears as:

**TESTING** AGAIN

If you click the bold attribute and text color 4 then type TESTING, click the bold attribute again, click text color 3, then type AGAIN, it will appear in the Editor Window as:

žž4TESTING žž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

**TESTING** AGAIN

See also: Underline, Control-K, Italics, Fixed Font, Symbol Font.

# Underline  Attribute

Located to the right of the BackGround Color Buttons, in the middle of the CTERM interface, the underline attribute button is used to apply the underline attribute to your text.   When clicking the underline attribute button, the underline character is inserted into the Editor Window. Any text entered after that code, will be displayed   as underlined text, when sent to the channel.   To reset the underline attribute after your text, you can click the underline attribute button again.

(Note: You MUST insert the UNDERLINE Attribute before the color code or pIRCh will strip the color code.)

## Examples:

If you click the underline attribute and text color 4 then type TEST, it will appear in the Editor Window as:

žž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

TEST

If you click the underline attribute and text color 4 then type TESTING, click the underline attribute again, and type AGAIN, it will appear in the Editor Window as:

žž4TESTINGž AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

If you click the underline attribute and text color 4 then type TESTING, then click the bold attribute, and click text color 3 and type AGAIN,   it will appear in the Editor Window as:

žž4TESTING žž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING **AGAIN**

See also: Bold, Control-K, Italics, Fixed Font, Symbol Font.

## Control-K Button

Located to the right of the Bold Attribute Button, in the middle of the CTERM interface, the Control-K button is used to insert a Control-K code to your text. When clicking the Control-K button, the Control-K character is inserted into the editor window. The Control-K is the essential character needed to creating colored text. The Text Color Buttons and the BackGround Color Buttons automatically insert the Control-K character with the correct code to produce the desired color. The purpose behind having the Control-K button is so you can reset color codes when needed. A Control-K entered by itself will reset the text color to black.

### Example:

If you click text color 4 and type TESTING, click the Control-K button and type AGAIN, it will appear in the Editor Window as:

ž4TESTING žAGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

See also: Bold, Underline, Italics, Fixed Font, Symbol Font.

## Help Button

Located to the right of the Underline Attribute Button, in the middle of the CTERM interface, the help button is used to bring up this help file.

## Editor Window

Located underneath the BackGround Color Buttons, towards the bottom of the CTERM interface, the Editor Window is the main tool used to view text before it is sent to the channel. Clicking any of the text manipulating buttons will send a set of characters to this window. You may then enter any text desired into the window, and then click the send button to send it to the channel. The Bold, Underline, Italics, Fixed Font, and Symbol Font attributes, as well as the Control-K will appear as a ž in the window, but will perform their functions when sent to the channel.

### Examples:

If you type "/me " then click text color 4, and type "is having a good time!", it will appear in the Editor Window as:

/me ž4is having a good time!

When you click the Send Button the text is sent to the channel, and appears as:

Slobokan is having a good time!

If you type "/me " then click text color 4, type "is laughing at" then click text color 3, and type " $snicks !", it will appear in the Editor Window as:

/me  ž4is laughing atž3 $snicks !

When you click the send button the text is sent to the active window, and appears as:

Slobokan is laughing at Slobokan !

See also: Variables.

## Settings

Located underneath the Editor Window at the bottom of the CTERM interface, the Settings Button enables you to use multiple INI files to store all your saved information. Upon clicking the Settings Button the Color Text Terminal Settings Window will open, showing you the filenames of the INI files currently in use by CTERM. To create a new INI file, all you need to do is enter the name of the new file under the appropriate heading. Upon clicking OK, CTERM will create the new INI file, and you will be able to start adding, editing and using your preset information in that file. To change an INI file setting, just enter the name of the INI file which you wish CTERM to use, and click OK.

See also: PILs, PopUps, TxtFiles, Sounds, Variables, Color Codes.

## Last Button

Located underneath the Editor Window at the bottom of the CTERM interface, the last button enables you re-insert the last text sent from CTERM, back into the Editor Window.   Using this button makes adding and editting entries to your INI files much easier.

See also: Clear, Send.

## Clear Button

Located underneath the Editor Window at the bottom of the CTERM interface, the clear button enables you clear the contents of the Editor Window.

See also: Last, Send.

## Send Button

Located underneath the Editor Window at the bottom of the CTERM interface, the Send Button enables you to send the contents of the Editor Window to the channel. Multiple lined entries are supported in CTERM by using the & symbol. (Note: The & symbol is usually the SHIFT-7 key on your keyboard).

CTERM will automatically convert the INI variables from the INI file, when you select a stored definition.

### Examples:

If you click text color 4 and type TESTING AGAIN&, click text color 3 and type I SURE LOVE TESTING!, it will appear in the Editor Window as:

ž4TESTING AGAIN&ž4I LOVE TESTING!

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN
I LOVE TESTING!

See also: Clear, Last.

# PopUps

Located on the right side of the CTERM interface, the PopUps Window contains all of the popup definitions you have saved in your PopUps INI file. To change which INI file is in use by CTERM, click the Settings Button. To the right of the PopUps Window is the PopUps Button. Clicking this button, sends the selected PopUp (showing in the PopUps Window) to the Editor Window. Clicking the Send Button will then send the popup to the channel.

CTERM will automatically convert the INI variables, when displaying a pre-defined definition in the Editor Window.

See also: Adding PopUps, Editing PopUps.

# TxtFiles

Located on the right side of the CTERM interface, the TxtFiles Window contains all of the text file definitions you have saved in your TxtFiles INI file. To change which INI file is in use by CTERM, click the Settings Button. To the right of the TxtFiles Window is the TxtFiles Button. Clicking this button, sends the selected TxtFile (showing in the TxtFiles Window) to the Editor Window. Clicking the Send Button will then send the text file to the channel.

CTERM will automatically convert the INI variables, when displaying a pre-defined definition in the Editor Window.

See also: Adding TxtFiles, Editing TxtFiles.

## Sounds

Located on the right side of the CTERM interface, the Sounds Window contains all of the sound definitions you have saved in your Sounds INI file. To change which INI file is in use by CTERM, click the Settings Button. To the right of the Sounds Window is the Sounds Button. Clicking this button, sends the selected Sound (showing in the Sounds Window) to the Editor Window. Clicking the Send Button will then send the sound to the channel.

CTERM will automatically convert the INI variables, when displaying a pre-defined definition in the Editor Window.

See also: Adding Sounds, Editing Sounds.

## Color Codes

The following is a list of the color codes used by pIRCh. (These are the codes that are sent to the channel via CTERM):

0 White is displayed as  
1 Black is displayed as Black  
2 Navy is displayed as Navy  
3 Green is displayed as Green  
4 Red is displayed as Red  
5 Maroon is displayed as Maroon  
6 Purple is displayed as Purple  
7 Olive is displayed as <unavailable>  
8 Yellow is displayed as Yellow  
9 Lime is displayed as Lime  
10 Teal is displayed as Teal  
11 Aqua is displayed as Aqua  
12 Blue is displayed as Blue  
13 Fuschia is displayed as Fuschia  
14 Gray is displayed as Gray  
15 Silver is displayed as Silver

For more information on color codes, please refer to the pIRCh help file.

See also: Text Color Buttons, BackGround Color Buttons, Control-K.

# Variables

The following is a list of variables which are used in the CTERM INI file configuration:

+A represents the $activewindow variable, inserting the active channel name.
+N represents the $1 variable, inserting the selected nickname.
+D represents the $date variable, inserting the current date (US Format).
+T represents the $time variable, inserting the current time.
+Z represents the $day variable, inserting the current day.

When adding definitions to your PopUps, TxtFiles, or Sounds, you can use the + variables listed above, or you can use the $ variables.

## Examples:

If your nick is UnderDog, the selected nickname is Slobokan, and the line in the Editor Window is:

```
/say Hello there, +N !!!
```

The following text would be displayed in the channel after clicking the Send Button:

```
<UnderDog> Hello there, Slobokan !!!
```

If your nick is UnderDog, you are in the #SloPace channel, and the line in the Editor Window is:

```
/me says hello to everyone in +A !!!
```

The following text would be displayed in the channel after clicking the Send Button:

```
UnderDog says hello to everyone in #SloPace !!!
```

If your nick is UnderDog, and the line in the Editor Window is:

```
/me is from Albuquerque, New Mexico, where it is +T (MST) on +Z , +D .
```

The following text might be displayed in the channel after clicking the Send Button:

UnderDog is from Albuquerque, New Mexico, where it is 08:05:00AM (MST) on Tuesday , 07/15/97 .

Other variables from pIRCh may be used in the CTERM INI files, by entering the variables just as you would if using them in pIRCh.   (Note: Not all variables will work).

See also:   PILs, PopUps, TxtFiles, Sounds.

## Adding PILs

To add a definition to your PILs, select Add from the PILs Window, and click the PILs Button. Upon clicking the button, the Add PIL Window will open.

Enter the name of your new PIL in the "Name for PIL" Window, then enter the name of the new PIL in the "PIL Filename" Window. If you make a mistake you may clear the entry by clicking the Clear Button. When you are finished making your entry, click the "Add PIL To CTERM" button.

(Note: In order for a PIL to work with CTERM you MUST install it in pIRCh like any other PIL, and you MUST edit the lines of the pil that send the text to the channel.)

Any line in a PIL that resembles the following line:

```
command('/msg ',$audience,' ',$target);
```

Must be replaced with:

```
command('/verbose PRIVMSG ',$activewin,' :',$target);
command('/display > ',$activewin,' ','<',$me,'> ','',$target);
```

in order for that PIL to work with CTERM 2.0 (PIL) for pIRCh.

### Examples:

Here is a list of example NAMES and FILENAMES for PILS:

| NAME | DEFINITION |
|---|---|
| Backwards | bckwrds |
| Banner | banner |
| Stars! | stars1 |

See also: PILs, Settings, Editing PILs, Variables.

# Editing PILs

To edit a definition in your PILs, select Edit from the PILs Window, and click the PILs Button. Upon clicking the button, the Edit PIL Window will open.

Select the name of the PIL you wish to edit from the "PIL to Edit" Window. The current filename will be displayed in the "Current PIL Filename" window (this line is READ ONLY). You may enter a new name or filename for your PIL or both. If you make a mistake, you may clear the entry by clicking the Clear Button. When you are finished making your entry, click the "Update PIL In CTERM" button.

(Note: CTERM does NOT support deleting PILs, to avoid catastrophic results for some end users, therefore, to DELETE a PIL, just EDIT the entry and change it to something different.)

(Note: In order for a PIL to work with CTERM you MUST install it in pIRCh like any other PIL, and you MUST edit the lines of the pil that send the text to the channel.)

Any line in a PIL that resembles the following line:

```
command('/msg ',$audience,' ',$target);
```

Must be replaced with:

```
command('/verbose PRIVMSG ',$activewin,' :',$target);
command('/display > ',$activewin,' ','<',$me,'> ','',$target);
```

in order for that PIL to work with CTERM 2.0 (PIL) for pIRCh.

## Examples:

Here is a list of example NAMES and FILENAMES for PILS:

| NAME | DEFINITION |
|---|---|
| Backwards | bckwrds |
| Banner | banner |
| Stars! | stars1 |

See also: PILs, Settings, Adding PILs, Variables.

## Editing PopUps

To edit a definition in your PopUps, select Edit from the PopUps Window, and click the PopUps Button. Upon clicking the button, the Edit PopUp Window will open.

Select the name of the PopUp you wish to edit from the "PopUp to Edit" Window. The current definition will be displayed in the "Current Definition of PopUp" window (this line is READ ONLY). You may enter a new name or definition for your popup or both. If you make a mistake, you may clear the entry by clicking the Clear Button. If the definition for your new PopUp is in the CTERM Editor Window, you can click the "Grab Definition From CTERM Editor Window" Button, and the text in the Editor Window will be copied to the "New Definition of PopUp" Window.

When you are finished making your entry, click the "Update PopUp In CTERM" button.

(Note: CTERM does NOT support deleting PopUps, to avoid catastrophic results for some end users, therefore, to DELETE a PopUp, just EDIT the entry and change it to something different.)

### Examples:

Here is a list of example NAMES and DEFINITIONS for PopUps:

Hugz
+B+K1{+K4{+K3{ +K12+N +K3}+K4}+K1}
**{{{ Slobokan }}}**

Slap!
/me slaps+K4 +N +K6in the eye with a +K6S+K3p+K4a+K6g+K3h+K4e+K6t+K3t+K4i+K6o+K3!
UnderDog slaps Slobokan in the eye with a Spaghettio!

HiHo Neighbor
/me says +K3"HiHo Neighbor" +B+K4--+K5ooo+K4-+K5O¿O+K4-+K5ooo+K4---+K+B+K3 as he peeks over+K12 +N 's +K3fence!
UnderDog says "HiHo Neighbor" **--ooo-O¿O-ooo--** as he peeks over Slobokan 's fence!

See also: PopUps, Settings, Adding PopUps, Variables.

## Adding PopUps

To add a definition to your PopUps, select Add from the PopUps Window, and click the PopUps Button. Upon clicking the button, the Add PopUp Window will open.

Enter the name of your new PopUp in the "Name for PopUp" Window, then enter the definition of the new PopUp in the "Definition of PopUp" Window. If you make a mistake you may clear the entry by clicking the Clear Button. If the definition for your new PopUp is in the CTERM Editor Window, you can click the "Grab Definition From Editor Window" Button, and the text in the Editor Window will be copied to the "Definition of PopUp" Window.

When you are finished making your entry, click the "Add PopUp To CTERM" button.

### Examples:

Here is a list of example NAMES and DEFINITIONS for PopUps:

Hugz
+B+K1{+K4{+K3{ +K12+N +K3}+K4}+K1}
**{{{ Slobokan }}}**

Slap!
/me slaps+K4 +N +K6in the eye with a +K6S+K3p+K4a+K6g+K3h+K4e+K6t+K3t+K4i+K6o+K3!
UnderDog slaps Slobokan in the eye with a Spaghettio!

HiHo Neighbor
/me says +K3"HiHo Neighbor" +B+K4--+K5ooo+K4-+K5O¿O+K4-+K5ooo+K4---+K+B+K3 as he peeks over+K12 +N 's +K3fence!
UnderDog says "HiHo Neighbor" **--ooo-*O¿O*-ooo--** as he peeks over Slobokan 's fence!

See also: PopUps, Settings, Editing PopUps, Variables.

## Editing TxtFiles

To edit a definition in your TxtFiles, select Edit from the TxtFiles Window, and click the TxtFiles Button. Upon clicking the button, the Edit TxtFile Window will open.

Select the name of the TxtFile you wish to edit from the "TxtFile to Edit" Window. The current definition will be displayed in the "Current Location of TxtFile" window (this line is READ ONLY). You may enter a new name or definition for your text file or both. You can do this manually, if you know the path to the specified text file, or you can select a text file from the convenient directory listings. If you make a mistake, you may clear the entry by clicking the Clear Button.

(Note: You DO NOT have to precede your entry with the /playfile command, as CTERM will automatically insert it for you.)

(Note: It is recommended that you enter a DELAY after the txtfile, so you don't flood yourself off the server. See the example below).

When you are finished making your entry, click the "Update TxtFile In CTERM" button.

(Note: CTERM does NOT support deleting TxtFiles, to avoid catastrophic results for some end users, therefore, to DELETE a TxtFile, just EDIT the entry and change it to something different.)

### Examples:

Here is a list of example NAMES and DEFINITIONS for TxtFiles:

Beatles
e:\files\txt\beatles.txt 1000
**This will play the beatles.txt file to the current channel with a 1 second delay between lines.**

Kiss!
e:\files\txt\kiss.txt 2000
**This will play the kiss.txt file to the current channel with a 2 second delay between lines.**

See also: TxtFiles, Settings, Adding TxtFiles, Variables.

## Adding TxtFiles

To add a definition to your TxtFiles, select Add from the TxtFiles Window, and click the TxtFiles Button. Upon clicking the button, the Add TxtFile Window will open.

Enter the name of your new TxtFile in the "Name for TxtFile" Window, then enter the location of the new TxtFile in the "Location of TxtFile" Window. You can do this manually, if you know the path to the specified text file, or you can select a text file from the convenient directory listings. If you make a mistake you may clear the entry by clicking the Clear Button.

(Note: You DO NOT have to precede your entry with the /playfile command, as CTERM will automatically insert it for you.)

(Note: It is recommended that you enter a DELAY after the txtfile, so you don't flood yourself off the server. See the example below).

When you are finished making your entry, click the "Add TxtFile To CTERM" button.

### Examples:

Here is a list of example NAMES and DEFINITIONS for TxtFiles:

Beatles
e:\files\txt\beatles.txt 1000
**This will play the beatles.txt file to the current channel with a 1 second delay between lines.**

Kiss!
e:\files\txt\kiss.txt 2000
**This will play the kiss.txt file to the current channel with a 2 second delay between lines.**

See also: TxtFiles, Settings, Editing TxtFiles, Variables.

## Adding Sounds

To add a definition to your Sounds, select Add from the Sounds Window, and click the Sounds Button. Upon clicking the button, the Add Sound Window will open.

Enter the name of your new Sound in the "Name for Sound" Window, then enter the filename of the new Sound in the "Filename of Sound" Window. You can do this manually, if you know the path to the specified sound, or you can select a sound from the convenient directory listings. If you make a mistake you may clear the entry by clicking the Clear Button.

You may add text AFTER the filename to make personalized Sound PopUps. See the example below.

(Note: You DO NOT have to precede your entry with the /sound command or the channel name, as CTERM will automatically insert it for you.)

When you are finished making your entry, click the "Add Sound To CTERM" button.

### Examples:

Here is a list of example NAMES and DEFINITIONS for Sounds:

Animaniacs
e:/files/wavz/anmaniac.wav
**This will play the wav file named "anmaniac.wav" to the current channel.**

Crazy
e:/files/wavz/crazy.wav is +K4C+K3R+K12A+K6Z+K4Y+K6 about+K4 +N +K6...
**This will play the wav file named "crazy.wav" to the current channel and display as:**
Slobokan is CRAZY about kittyspaw ...

See also: Sounds, Settings, Editing Sounds, Variables.

## Editing Sounds

To edit a definition in your Sounds, select Edit from the Sounds Window, and click the Sounds Button. Upon clicking the button, the Edit Sound Window will open.

Select the name of the Sound you wish to edit from the "Sound to Edit" Window. The current definition will be displayed in the "Current Filename of Sound" window (this line is READ ONLY). You may enter a new name or definition for your sound or both. You can do this manually, if you know the path to the specified sound, or you can select a sound from the convenient directory listings. If you make a mistake you may clear the entry by clicking the Clear Button.

(Note: You DO NOT have to precede your entry with the /sound command or the channel name, as CTERM will automatically insert it for you.)

When you are finished making your entry, click the "Update Sound In CTERM" button.

(Note: CTERM does NOT support deleting Sounds, to avoid catastrophic results for some end users, therefore, to DELETE a Sound, just EDIT the entry and change it to something different.)

### Examples:

Here is a list of example NAMES and DEFINITIONS for Sounds:

Animaniacs
e:/files/wavz/anmaniac.wav
**This will play the wav file named "anmaniac.wav" to the current channel.**

Crazy
e:/files/wavz/crazy.wav   is +K4C+K3R+K12A+K6Z+K4Y+K6 about+K4 +N +K6...
**This will play the wav file named "crazy.wav" to the current channel and display as:**
Slobokan is CRAZY about kittyspaw ...

See also: Sounds, Settings, Adding Sounds, Variables.

# About CTERM

This version of Slobokan's Color Text Terminal is a PIL specifically designed for use with the pIRCh® IRC Client.   CTERM provides an easy to use interface that makes sending colored text to the channel as easy as a click of the mouse.   CTERM supports multiple INI file support (see Settings), so you can organize and sort your favorite PILs, PopUps, TxtFiles, and Sounds.    CTERM is released as FREEWARE, but donations are greatly appreciated.

Those making donations will receive news and information regarding future releases.

If you wish to make a donation, send email to slobokan@usa.net.

## Italics Attribute

Located to the right of the Text Color Buttons, in the middle of the CTERM interface, the italics attribute button is used to apply an italics face to your text.   When clicking the italics attribute button, the italics character is inserted into the editor window. Any text entered after that code, will be displayed as italic text, when sent to the channel.   To reset the itlaics attribute after your text, you can click the italics attribute button again.

(Note: You MUST insert the ITALICS Attribute before the color code or pIRCh will strip the color code.)

### Examples:

If you click the italics attribute and text color 4 then type TEST, it will appear in the Editor Window as:

žž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

*TEST*

If you click the italics attribute and text color 4 then type TESTING, click the italics attribute again then type *AGAIN*, it will appear in the Editor Window as:

žž4TESTING žAGAIN

When you click the Send Button the text is sent to the channel, and appears as:

*TESTING* AGAIN

If you click the italics attribute and text color 4 then type TESTING, click the italics attribute again, click text color 3, then type *AGAIN*,   it will appear in the Editor Window as:

žž4TESTING žž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

*TESTING* AGAIN

See also: Underline, Control-K, Bold, Fixed Font, Symbol Font.

## Fixed Font Attribute

Located to the right of the Text Color Buttons, in the middle of the CTERM interface, the fixed font attribute button is used to apply a fixed font to your text. When clicking the fixed font attribute button, the fixed font character is inserted into the editor window. Any text entered after that code, will be displayed in the fixed font, when sent to the channel. To reset the fixed font attribute after your text, you can click the fixed font attribute button again.

(Note: You MUST insert the FIXED FONT Attribute before the color code or pIRCh will strip the color code.)

### Examples:

(Note: I could not properly represent the fixed system font in this section).

If you click the fixed font attribute and text color 4 then type TEST, it will appear in the Editor Window as:

žž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

TEST

If you click the fixed font attribute and text color 4 then type TESTING, click the fixed font attribute again then type AGAIN, it will appear in the Editor Window as:

žž4TESTING žAGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

If you click the fixed font attribute and text color 4 then type TESTING, click the fixed font attribute again, click text color 3, then type AGAIN, it will appear in the Editor Window as:

žž4TESTING žž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

See also: Underline, Control-K, Bold, Italics, Symbol Font.

## Symbol Font Attribute

Located to the right of the BackGround Color Buttons, in the middle of the CTERM interface, the symbol font attribute button is used to apply the symbol font attribute to your text.  When clicking the symbol font attribute button, the symbol font character is inserted into the Editor Window. Any text entered after that code, will be displayed  in the symbol font, when sent to the channel.  To reset the symbol font attribute after your text, you can click the symbol font attribute button again.

### Examples:

If you click the symbol font attribute and text color 4 then type TEST, it will appear in the Editor Window as:

žž4TEST

When you click the Send Button the text is sent to the channel, and appears as:

□□□□

If you click the symbol font attribute and text color 4 then type TESTING, click the symbol font attribute again, and type *AGAIN*, it will appear in the Editor Window as:

žž4TESTINGž AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING AGAIN

If you click the symbol font attribute and text color 4 then type TESTING, then click the bold attribute, and click text color 3 and type *AGAIN*,  it will appear in the Editor Window as:

žž4TESTING žž3AGAIN

When you click the Send Button the text is sent to the channel, and appears as:

TESTING **ΑΓΑΙΝ**

See also: Bold, Control-K, Italics, Underline, Fixed Font.

# PILs

Located on the right side of the CTERM interface, the PILs Window contains all of the preset definitions you have saved in your PILs INI file. To change which INI file is in use by CTERM, click the Settings Button. To the right of the PILs Window is the PILs Button. Clicking this button, inserts the selected PIL command (showing in the PILs Window) to the Editor Window before your text. Clicking the Send Button will then send the PIL to the channel.

CTERM will automatically insert the /runscript command along with the PIL name, before any existing text in the Editor Window.

(Note: In order for a PIL to work with CTERM you MUST install it in pIRCh like any other PIL, and you MUST edit the lines of the pil that send the text to the channel.)

Any line in a PIL that resembles the following line:

```
command('/msg ',$audience,' ',$target);
```

Must be replaced with:

```
command('/verbose PRIVMSG ',$activewin,' :',$target);
command('/display > ',$activewin,' ','<',$me,'> ','',$target);
```

in order for that PIL to work with CTERM 2.0 (PIL) for pIRCh.

See also: Adding PILs, Editing PILs.