ExtraPuTTY User Manual

ExtraPuTTY is a fork (develop under Earth-licensed) from 'Putty 0.64' Win32 Telnet and SSH client. This manual documents ExtraPuTTY.

Chapter 1 : Introduction to ExtraPuTTY

ExtraPuTTY is a fork of Putty.

1.1 ExtraPutty and PuTTY

The ExtraPuTTY V0.29 is based on PuTTY release 0.64,and has all the features from the original soft and adds others as described bellow.

1.2 ExtraPuTTY features

ExtraPuTTY needs VC2005 redistributable package SP1 for lua sripting or put "Microsoft.VC80.CRT" on the same folder of the executable

Principal features

- DLL frontend (win32, Teststand API)
- Shortcuts for pre-defined command
- Automatic sequencing of commands
- Automatic logon script
- URL hyperlinks
- Portability (save session in file instead of registry)
- PuTTY Session Manager
- Timestamps in logs and/or in Terminal
- Report Generation
- ftploader utilities
- Specific commands
  - Predifined commands
  - Virtual key codes
  - Unprintable characters

Graphical features:

- Menu bar in the PuTTY window
- Configuration window : resize of session list box
- Serial port settings : display all available serial ports
- Startup in full screen mode

Technical features:

- Scripting a session with lua
- TestStand (3.x , 4.x) steps
- ZModem integration(experimental)

Other features:

- The PuTTYCyg patch
- New command-line options

Chapter 2: Getting started with ExtraPuTTY

This chapter gives a quick guide to the simplest types of interactive login session using ExtraPuTTY.

2.1 Starting a session

When you start PuTTY, you will see a dialog box. This dialog box allows you to control everything ExtraPuTTY and PuTTY can do. See chapter 4 for details of all the things you can control.

You don't usually need to change most of the configuration options. To start the simplest kind of session, all you need to do is to enter a few basic PuTTY parameters (PuTTY basic parameters is without ExtraPuTTY).

For more details of this dilaog box see Chapter 2.1 Starting a session of PuTTY

2.2 Logging in

After you have connected, and perhaps verified the server's host key, you will be asked to log in, probably using a username and a password. Your system administrator should have provided you with these. Enter the username and the password, and the server should grant you access and begin your session. If you have mistyped your password, most servers will give you several chances to get it right.

2.3 After logging in

After you log in to the server, what happens next is up to the server! Most servers will print some sort of login message and then present a prompt, at which you can type commands which the server will carry out. At this time you can used ExtraPuTTY to send command on server or manually from PuTTY terminal.

2.4 Logging out

When you have finished your session, you should log out by typing the server's own logout command. This might vary between servers; if in doubt, try `logout` or `exit`, or consult a manual or your system administrator. When the server processes your logout command, the ExtraPuTTY and PuTTY windows should close itself automatically.

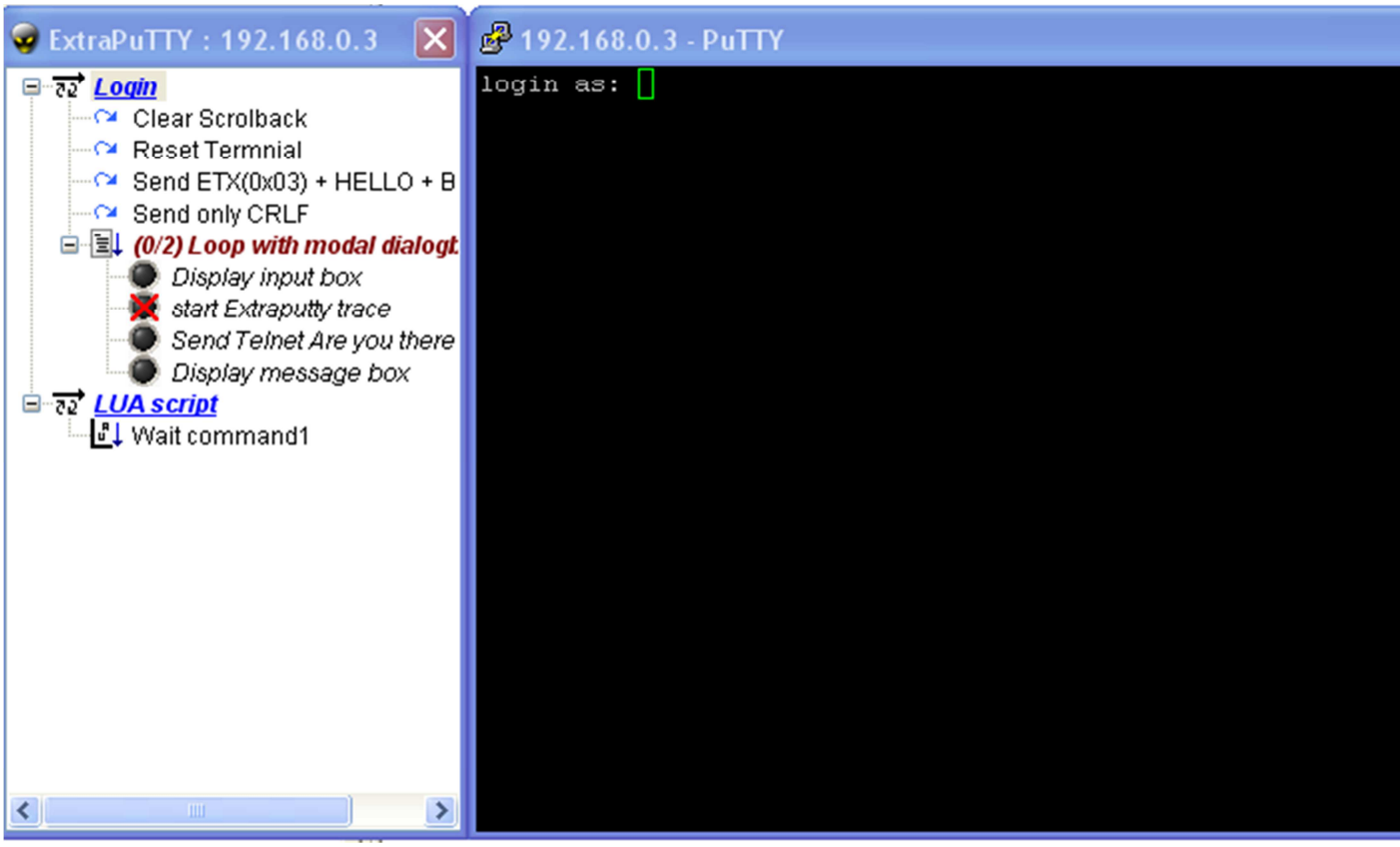
You *can* close a PuTTY session using the Close button in the window border (from ExtraPuTTY or PuTTY), but this might confuse the server - a bit like hanging up a telephone unexpectedly in the middle of a conversation. We recommend you do not do this unless the server has stopped responding to you and you cannot close the window any other way.

Chapter 3: Using ExtraPuTTY

This chapter provides a general introduction to some more advanced features of ExtraPuTTY. For extreme detail and reference purposes, chapter 4 is likely to contain more information.

3.1 : During your session

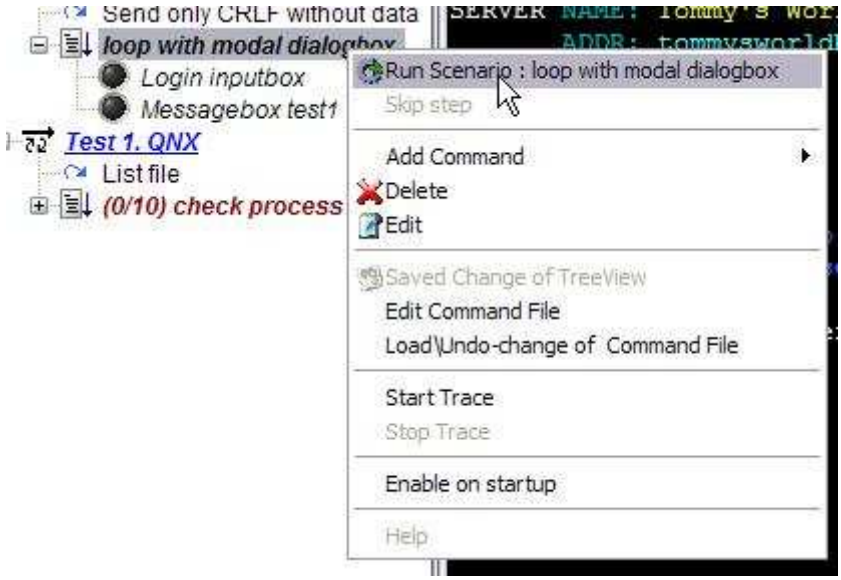
A lot of ExtraPuTTY's complexity and features are in the configuration panel. Once you have worked your way through that and started a session, things should be reasonably simple after that. Nevertheless, there are a few more useful features available.



3.1.1 Send simple command, or sequencing of commands

On ExtraPuTTY window you can send data from treeview by double-click into your terminal session.

It's possible to start a sequence of few commands.For do that you must click on the right mouse button and click run scenario on the popup menu.

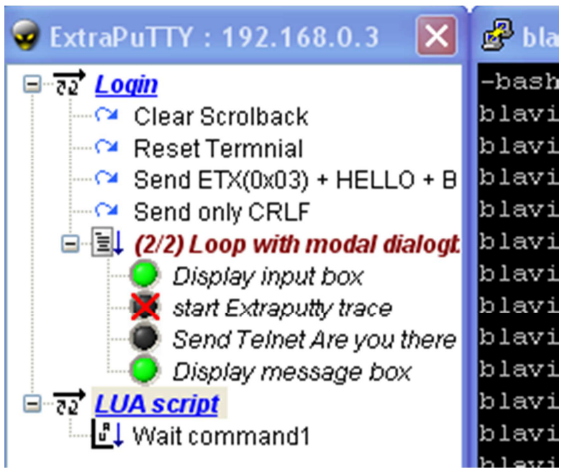


3.1.2 Scripting a session with lua

During a session you can start a lua script from ExtraPutty window or win32 API.

You can translates (compile) your programs written in the Lua programming language into binary files that can be used instead of lua script.
The main advantages of precompiling chunks are: faster loading, protecting source code from user changes, and off-line syntax error detection, BUT
does not imply faster execution. To do that used luac 5.1.4 compiler. (luac man page)

Example from ExtraPuTTY window :



3.1.2.1 Some lua functions are available to sciptring your putty session

- lua_send_login_password

  Function used to send login and password.

  Syntax

```
int lua_send_login_password(char *login,char *password,int timeout);
```

Parameters

Parameter	Description
login	Login to send.
password	Password to send.
timeout	Timeout in ms on log or password prompt.

Return Values

Return Value	Description
0	No error.
1	Function failed , timeout on login prompt.
2	Function failed , timeout on password prompt.

- lua_get_log_file

Function used to get log file name.

Syntax

```
char *lua_get_log_file();
```

Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
Path+FileName	No error.

- lua_log_start

  Function used to control putty logging (Session\logging from config box).

  Syntax

```
int lua_log_start(int LogType,Char *Logfile);
```

  Parameters

Parameter	Description
LogType	LogType (1: Printable Output, 2: All traffic, 3: SSH data packets, 4: SSH raw data) .
Logfile	Path+logging file name.

  Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_log_stop

  Function used to stop, save and close file putty logging (Session\logging from config box).

  Syntax

```
int lua_log_stop();
```

  Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_stop_trace

Function used to de-activate the trace and to reset the trace buffer :

Syntax

```
int lua_stop_trace();
```

Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_reset_trace

Function used to reset the trace buffer :

Syntax

```
int lua_reset_trace();
```

Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_getdata

Function used to get all inputs buffer :

Syntax

```
int lua_getdata();
```

Parameters

Parameter	Description
NA	No parameters.

Return Value for first return element

Return Value	Description
0	No error.

1	Function failed.

Return Value for second return element

Return Value	Description
String	Contents of trace buffer, only if the first return element is set to 0 otherwise nil.

- lua_senddata

Function used to send a command :

Syntax

```
int lua_senddata(char *Command,bool EOL);
```

Parameters

Parameter	Description
Command	Command to send to the target.
EOL	End of Line.

Return Values

Return Value	Description
0	No error.
1	Function failed.

Field "Command" can be a predifned ExtraPuTTY commands EXT_xxx (See FAQ to get all commands).

Examples :

- Sending basic command with EOF : lua_senddata("ls",true);

- Sending Predifined command Display inputbox : lua_senddata("EXT_MSGBOX_INPUTBOX_TEST1",true);
- Sending Predifined command Edit ExtraPuTTY commands file : lua_senddata("EXT_SYS_CMD_EDIT_XML_FILE",true);

- lua_log_start

Function used to enable the putty session logging.

Syntax

```
int lua_log_start(int SessionLogging);
```

Parameters

Parameter	Description
SessionLogging	Type of logging to activate. (1 : Printable Output, 2 : All traffic, 3 : SSH data packets, 4 : SSH raw data)

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_log_stoplua_log_stoplua_log_stoplua_log_stoplua_log_stoplua_log_stoplua_log_stoplua_log_stoplua_log_stop

Function used to disable the putty session logging.

Syntax

```
int lua_log_stop();
```

Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_write_log

  Function used to append comments in putty log file :

  Syntax

  ```
  int lua_write_log(char *comment,int size);
  ```

  Parameters

Parameter	Description
Text	Comment to append in log file.
size	Size of comment.

  Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_def_EOF

  Function used to change end of line :

  Syntax

  ```
  int lua_def_EOF(char *EndOfLine);
  ```

Parameters

Parameter	Description
EndOfLine	New End of Line. (<CR><LF> or <LF><CR> or <CR> or <LF> else set to default value <CR><LF>)

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_sleep

Function used to make a pause like "sleep" C function :

Syntax

```
int lua_sleep(unsigned long Time);
```

Parameters

Parameter	Description
Time	During of sleep in ms.

Return Values

Return Value	Description
0	No error.
1	Function failed.

- lua_currentdir

  Function used to get path of current directory :

  Syntax

  ```
  int lua_currentdir();
  ```

  Parameters

Parameter	Description
NA	No parameters.

  Return Value for first return element

Return Value	Description
0	No error.
1	Function failed.

  Return Value for second return element

Return Value	Description
String	Path of putty executable

- lua_msgbox

  Function used to display message box :

  Syntax

  ```
  int lua_msgbox(char *Text,char *Title,unsigned long Button,unsigned long Icon,unsigned long DefButton);
  ```

  Parameters

Parameter	Description

Text	Text of the message box.
Title	Title of the message box.
Button	To indicate the buttons displayed in the message box, specify one of the following values : o **0x00000002L** : The message box contains three push buttons: Abort, Retry, and Ignore. o **0x00000006L** : The message box contains three push buttons: Cancel, Try Again, Continue. Use this message box type instead of MB_ABORTRETRYIGNORE. o **0x00004000L** : Adds a Help button to the message box. When the user clicks the Help button or presses F1, the system sends a WM_HELP message to the owner. o **0x00000000L** : The message box contains one push button: OK. This is the default. o **0x00000001L** : The message box contains two push buttons: OK and Cancel. o **0x00000005L** : The message box contains two push buttons: Retry and Cancel. o **0x00000004L** : The message box contains two push buttons: Yes and No. o **0x00000003L** : The message box contains three push buttons: Yes, No, and Cancel.
Icon	To display an icon in the message box, specify one of the following value : o **0x00000030L** : An exclamation-point icon appears in the message box. o **0x00000040L** : An icon consisting of a lowercase letter i in a circle appears in the message box. o **0x00000020L** : A question-mark icon appears in the message box. The question-mark message icon is no longer recommended because it does not clearly represent a specific type of message and because the phrasing of a message as a question could apply to any message type. In addition, users can confuse the message symbol question mark with Help information. Therefore, do not use this question mark message symbol in your message boxes. o **0x00000010L** : A stop-sign icon appears in the message box.
DefaultButton	To indicate the default button, specify one of the following values : o **0x00000000L** : The first button is the default button. o **0x00000100L** : The second button is the default button. o **0x00000200L** : The third button is the default button. o **0x00000300L** : The fourth button is the default button.

Return Values

Return Value	Description
0	Function failed.
1	The OK button was selected.
2	The Cancel button was selected.
3	The Abort button was selected.
4	The Retry button was selected.
5	The Ignore button was selected.
6	The Yes button was selected.
7	The No button was selected.
10	The Try Again button was selected.
11	The Continue button was selected.

- lua_inputbox

  Function used to display input message box :

  Syntax

```
int lua_inputbox(char *Text,char *Title);
```

  Parameters

Parameter	Description

Text	Text of the message box.
Title	Title of the message box.

2 Return Values

Return Value 1	Description
0	The OK button was selected.
1	The CANCEL button was selected.
1	Function failed.

Return Value 2	Description
char *	The text of the input box.

- lua_tftp_start

  Function used to start TFTP server.

  Syntax

```
int lua_tftp_start();
```

  Parameters

Parameter	Description
NA	No parameters.

  Return Values

Return Value	Description
0	No error.
1	Function failed .

- lua_ftp_ServIp

  Function used to change dynamically ip of FTP server for DragAndDrop.

  Syntax

  ```
  int lua_ftp_ServIp(char *ip);
  ```

  Parameters

Parameter	Description
ip	IP of FTP server.

  Return Values

Return Value	Description
0	No error.

- lua_ftp_open

  Function used to start FTP Client.

  Syntax

  ```
  int lua_ftp_open(hostname,login,password);
  ```

  Parameters

hostname	Hostname or ip of FTP server.

login	Login.
password	Password.

Return Values

Return Value	Description
0	No error.
1	Function failed .

- lua_ftp_close

  Function used to stop FTP Client.

  Syntax

  ```
  int lua_ftp_close();
  ```

  Parameters

Parameter	Description
NA	No parameters.

  Return Values

Return Value	Description
0	No error.
1	Function failed .

- lua_ftp_put

Function used to put files.

Syntax

```
int lua_ftp_put(HostPath,RemotePath);
```

Parameters

HostPath	Directory or folder to upload.
RemotePath	remote path to upload files.

Return Values

Return Value	Description
0	No error.
1	Function failed .

- lua_ftp_get

Function used to get files.

Syntax

```
int lua_ftp_get(HostPath,RemotePath,Files);
```

Parameters

HostPath	Directory to put remote files.
RemotePath	remote path of files to download.
Files	Files to download.

Return Values

Return Value	Description

0	No error.
1	Function failed .

- lua_ymodem_rcv

  Function used to receive file with ymodem 1K protocol.

  Syntax

  ```
  int lua_ymodem_rcv(char *DestPath);
  ```

  Parameters

Parameter	Description
DestPath	Local path to save file, if DestPath == nil file save in putty folder.

  Return Values

Return Value	Description
0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_ymodem_snd

  Function used to send file with ymodem 1K protocol.

  Syntax

  ```
  int lua_ymodem_snd(char *PathFile);
  ```

  Parameters

Parameter	Description

PathFile	Path + FileName to transfert.

Return Values

Return Value	Description
0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_xmodem_rcv

Function used to receive file with xmodem protocol.

Syntax

```
int lua_xmodem_rcv(char *DestPath,char *FileName);
```

Parameters

Parameter	Description
DestPath	Local path to save file.
FileName	Name of file to receive.

Return Values

Return Value	Description
0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_xmodem1K_rcv

Function used to receive file with xmodem 1K protocol.

Syntax

```
int lua_xmodem1K_rcv(char *DestPath,char *FileName);
```

Parameters

Parameter	Description
DestPath	Local path to save file.
FileName	Name of file to receive.

Return Values

Return Value	Description
0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_xmodem_snd

Function used to send file with xmodem protocol.

Syntax

```
int lua_xmodem_snd(char *PathFile);
```

Parameters

Parameter	Description
PathFile	Path + FileName to transfert.

Return Values

Return Value	Description

0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_xmodem1K_snd

  Function used to send file with xmodem 1K protocol.

  Syntax

```
int lua_xmodem1K_snd(char *PathFile);
```

  Parameters

Parameter	Description
PathFile	Path + FileName to transfert.

  Return Values

Return Value	Description
0	No error.
1	Function failed , transfert aborted or protocol is Cygwin.

- lua_status_bar

  Function used to write text in third part of status bar.

  Syntax

```
int lua_status_bar(int StatusBarItem,char *Text);
```

  Parameters

Parameter	Description

Item	Status bar item (1to8) to set text.
Text	Text to display in status bar.

Return Values

Return Value	Description
0	No error.
1	Function failed (no error if item is disabled).

- lua_get_protocol

  Function used to get protocol of current session.

  Syntax

  ```
  int lua_get_protocol();
  ```

  Parameters

Parameter	Description
NA	No parameters.

  Return Values

Return Value	Description
Protocol	0: RAW, 1: TELNET, 2: RLOGIN, 3: SSH, 4: CYGTERM, 5: SERIAL
1	Function failed.

- lua_get_hostname

Function used to get hostname of current session.

Syntax

```
int lua_get_hostname();
```

Parameters

Parameter	Description
NA	No parameters.

Return Value for first return element

Return Value	Description
0	No error.
1	Function failed.

Return Value for second return element

Return Value	Description
String	Hostname of current session.

- lua_get_session_name

Function used to get session name.

Syntax

```
int lua_get_session_name();
```

Parameters

Parameter	Description
NA	No parameters.

Return Value for first return element

Return Value	Description
0	No error.
1	Function failed.

Return Value for second return element

Return Value	Description
String	Session Name.

- lua_close_window

Function used to close extraputty window :

Syntax

```
int lua_close_window();
```

Parameters

Parameter	Description
NA	No parameters.

Return Values

Return Value	Description
0	No error.
1	Function failed.

Example of using:

- lua script.

3.1.2 Different items of Treeview

ExtraPuTTY is composed of one treeview which contained 4 different icons :

3.1.4 System menu

If you click the left mouse button on the icon in the top left corner of ExtraPuTTY's terminal window, or click the right mouse button on the title bar, you will see the standard Windows system menu containing items like Minimise, Move, Size and Close.

ExtraPuTTY's system menu contains extra program features in addition to the Windows standard options. These extra menu commands are described below.

(These options are also available in a context menu brought up by holding Ctrl and clicking with the right mouse button anywhere in the ExtraPuTTY window.)

Stop Scenario

Used to stop the current scneario.

Detach from PuTTY Window

The ExtraPuTTY position is free.

Attach with PuTTY Window, resize on PuTTY size

The ExtraPuTTY window is attach on the left side of PuTTY window.

Attach with PuTTY Window, no-resize

The ExtraPuTTY window is attach on the left side and resize of PuTTY window.

Lock position and size

Memorise the ExtraPuTTY and PuTTY windows position in XML commans file, in order to restart the session with the same position of windows.

Unlock position and size

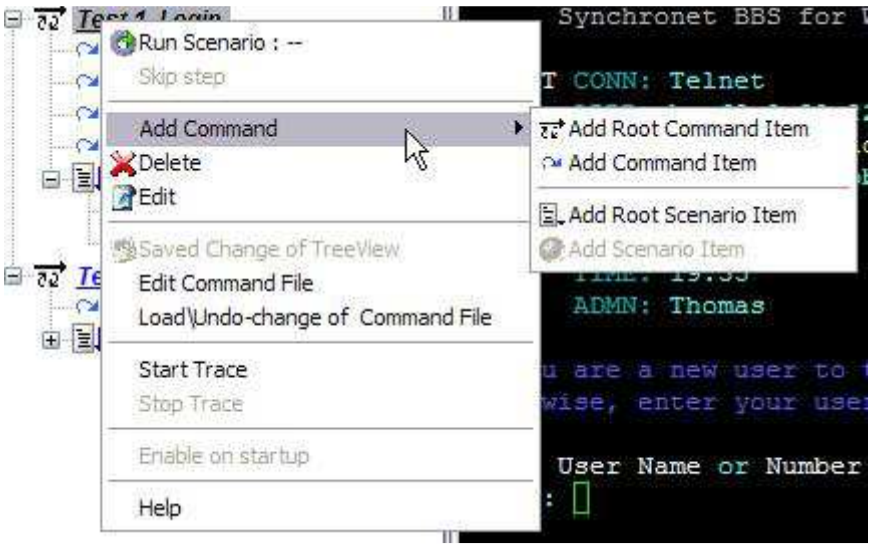
The ExtraPuTTY and PuTTY windows are load with default position.

About ExtraPuTTY

Display the About box.

3.1.5 Popup menu

If you click the right mouse button on ExtraPuTTY's editor window, you will see the Popup commands which are described below.



Run Scenario : ScenarioName_Select

Used to start sequence of command select.

skip or execute step

Used to skip or execute step in sequence of commands.

Add command

Used to add root item,commands,scenario item,scenario commands.

Delete

Used to delete the selected item in Treeview.

Edit

Edit the parameters of item selected in Treeview.

Saved Change of TreeView

Saved all changes.

Edit command file

Edit the XML commands file with the default XML application on your system.

Load\Undo_change of command file

Load the XML file during the session in order to take into account the modifications of files or to undo all changes.

Start trace

Start the generation of HTML report.

Stop trace

Stop and load the HTML report.

Enable on startup

Run a scenario automatically on startup of putty session.(the scenario are execute fom up to down of the extraputty window)

Disable on startup

Unset the scenario in automatic start on startup of putty session.

Help

Open associate html help page of selected item.

3.2 : The ExtraPuTTY command line

ExtraPuTTY can be made to do various things without user intervention by supplying command-line arguments (e.g., from a command prompt window, or a Windows shortcut).

Specific to serial line (missing of putty help) : putty.exe -serial COMx

3.2.1 Run scenario or lua script on startup of putty session

In order to run a scenario or lua script on startup called "name1" use the -startup option. (Refer to command line help file for complete description of putty options)

putty.exe putty_options -startup "name1"(putty_options for serial line (missing of putty help) : -serial COMx)

Note : You need double quotes around the session name if it contains spaces.

3.2.2 To copy all the configuration from the registry to files

It is possible to copy all the configuration from the registry for users who already created sessions with "REG_MODE" mode of ExtraPuTTY and wants to used ExtraPuTTY in "DIR_MODE".To do that use the *-sessions-reg-to-file* option.

This option will :
- create 2 sub-directories: Sessions SshHostKeys which containing all the configuration,
- **delete all registry keys**.

putty.exe -sessions-reg-to-file

3.2.3 To copy all the configuration from files to registry

It is possible to copy all the configuration from the Files for users who already created sessions with "DIR_MODE" mode of ExtraPuTTY and wants to used ExtraPuTTY in "REG_MODE".
To do that use the *-sessions-file-to-reg* option. This option will delete all configuration files (sessions and SshHostKeys).

putty.exe -sessions-file-to-reg

3.2.4 Display licence dialog box

Display the licence in window dialogbox.To do that use the *-license* option.

putty.exe -license

3.2.5 Display about dialog box

Display the About in window dialogbox.To do that use the *-about* option.

putty.exe -about

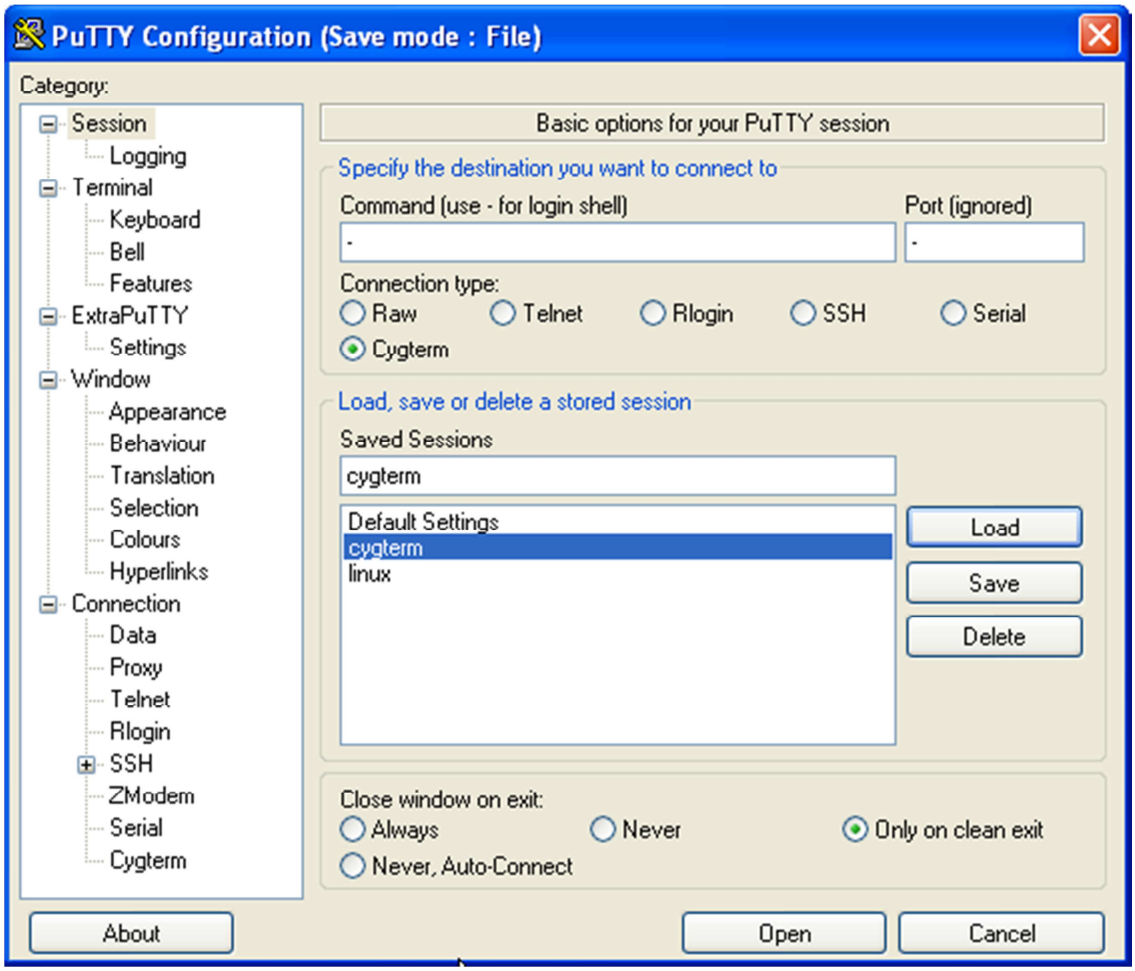
Chapter 4: Using ExtraPuTTY

This chapter describes all the configuration options in ExtraPuTTY.

ExtraPuTTY is configured using the control panel that comes up before you start a session. Some options can also be changed in the middle of a session, by selecting 'Change Settings' from the window menu of PuTTY.

4.1 The session panel

The Session configuration panel contains the basic options you need to specify in order to open a session at all, and also allows you to save your settings to be reloaded later.



4.1.1 ExtraPuTTY Options

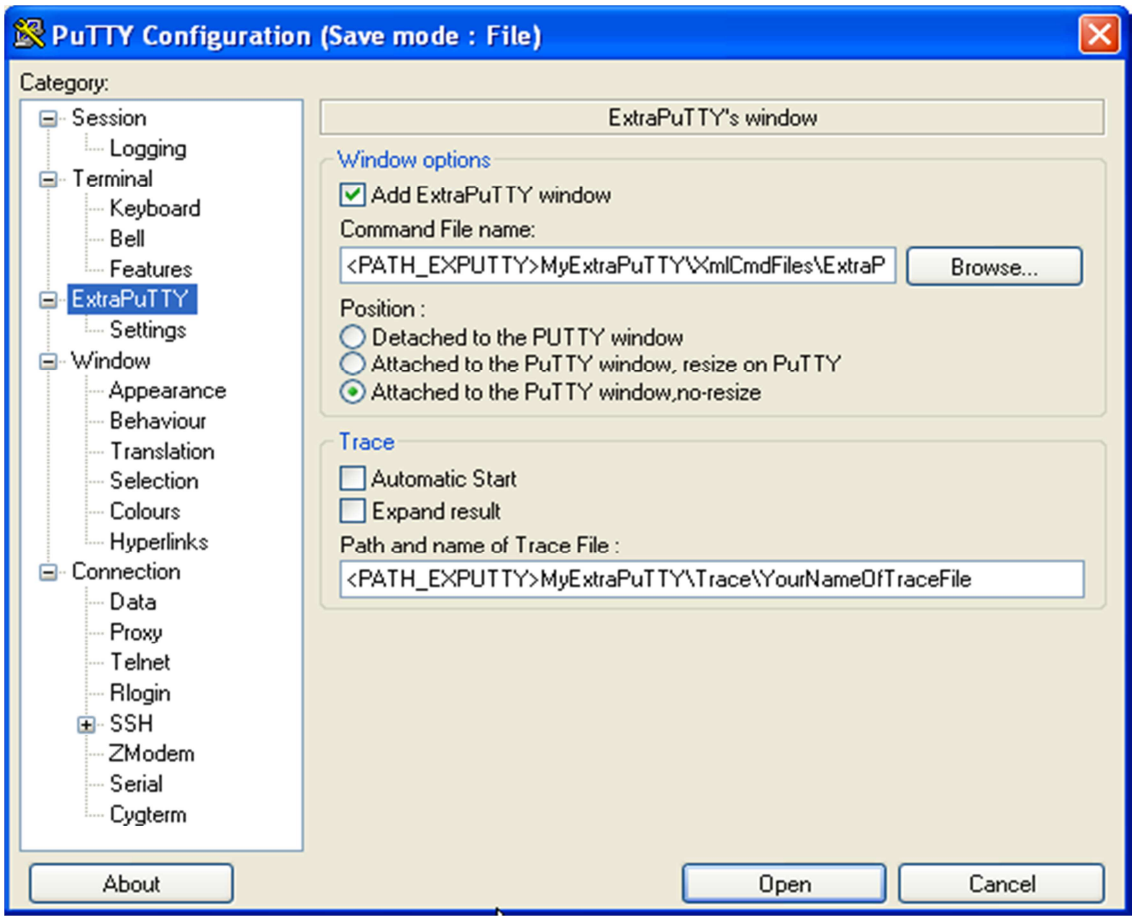
Never, Auto-Connect

Used to automatically restart the session every x ms after it has terminated (x ms can be defined in ExtraPuTTY/Settings panel)
When this option is enabled, the keepalive option of PuTTY connection panel is set

to 1 second, only if the keepalive is turn off.(see '4.13.1 of putty help document for a complet description)
The keepalive option is used to detect that the connection was closed unexpectedly or not properly.

4.2 The ExtraPuTTY panel

The Session configuration panel contains the basic options you need to specify in order to open a session at all, and also allows you to save your settings to be reloaded later.



4.2.1 Window 0ptions

Add ExtraPuTTY window

Used to display the ExtraPuTTY window

Command File Name

*Select your XML commands file. You can used **<PATH_EXPUTTY>** for relative path of putty.exe.*
You can have one file by session and select a empty file in order to complet it with ExtraPuTTY editor.
The command file can be edit,change and reload during a session.If this parameter is not set and the tools box activate empty XML commands file is create : C:\\ExtraPuTTY_Settings.xml

Position

Detached from PuTTY Window : the ExtraPuTTY position is free.
Attached to the PuTTY Window, resize on PuTTY size : The ExtraPuTTY window is

attach on the left side of PuTTY window.
Attached to the PuTTY Window, no-resize : The ExtraPuTTY window is attach on the left side and resize of PuTTY window.

4.2.2 Trace

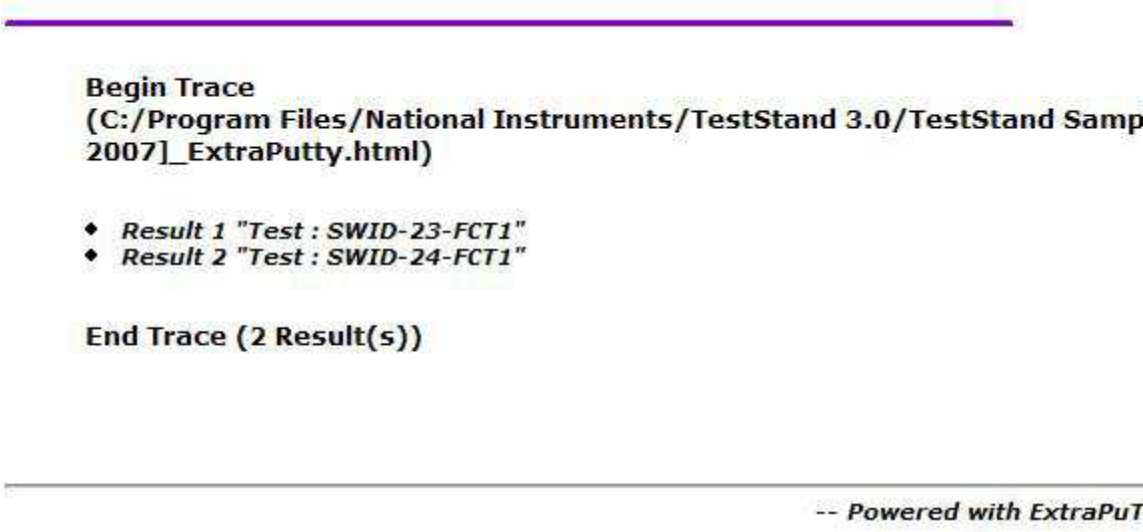
Automatic Start

Used to start automaticly ExtraPuTTY report when the session is open.

Expand Result

Default sate of the result.

Not Expand :



Expand :

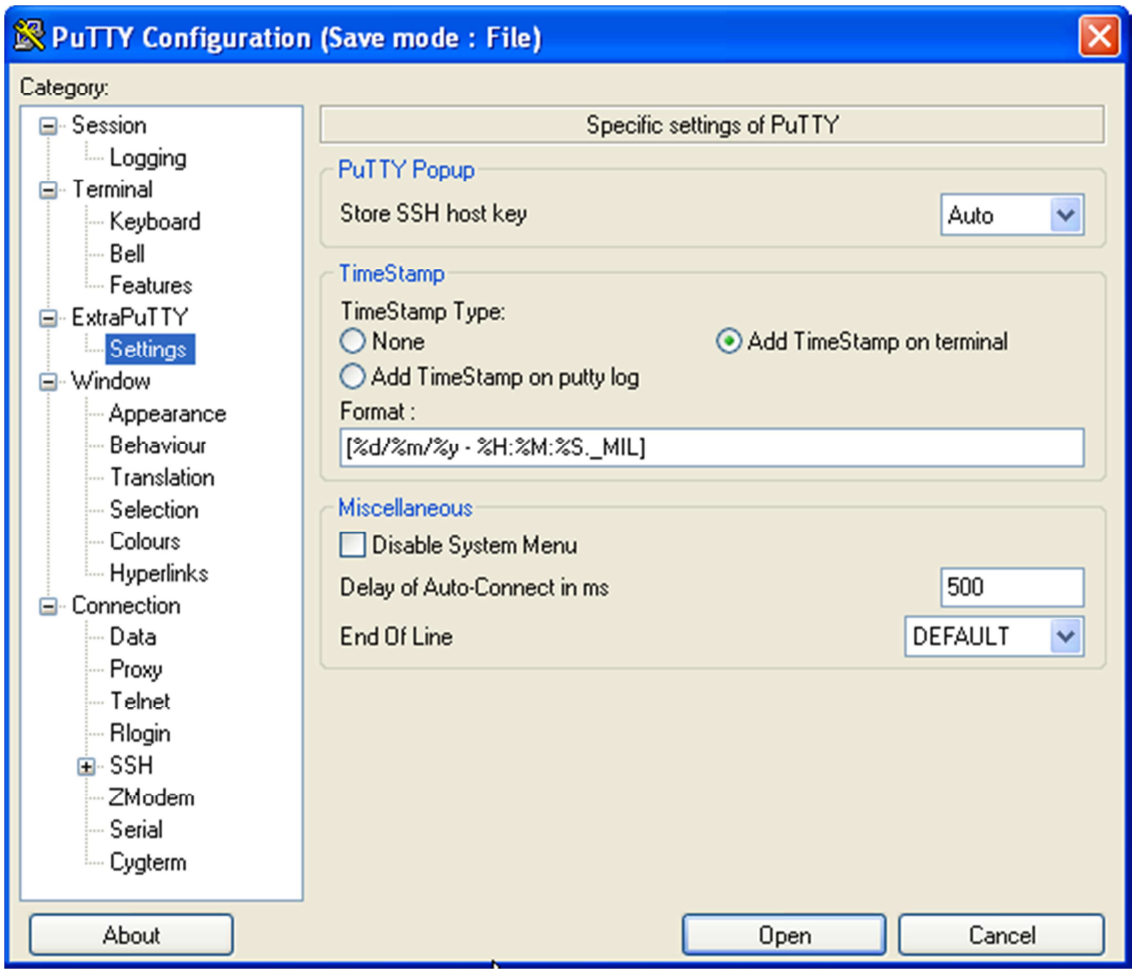

Begin Trace
(C:/Program Files/National Instruments/TestStand 3.0/TestStand Sample s
2007]_ExtraPutty.html)
Result 1 "Test : SWID-23-FCT1"
Test : SWID-23-FCT1
Command:
Date and Time:
Fri Feb 02 04:31:29 PM
Check that the data is received
Target Reply :
[Lobby] 4 messages, 4 new Jun 15, 2007 19:11 from Devil Lady
Have you ever wanted to give a little back? ISCA is accepting donations via PayPal t
operating costs of the BBS. It will not be for recognition, but it will be very appreciate
payments to "ISCA@gailbunny.com". Funds are non-refundable and become the pro
NPO. Currently ISCA has almost $500 debt for hosting that has come out of pocket
and this hosting is where the funds will be applied. Comments or questions to Mail> o
Treasurer of ISCA NPO
[Lobby> msg #2275 (3 remaining)] Read cmd ->**

Path and Name of trace file

The name and path of Report. if this parameter is not set a default file is create : c:\\ExtraPuTTYTrace[Date][Time].html
You can used **<PATH_EXPUTTY>** for relative path of putty.exe.

4.3 Specific settings

The specific settings configuration panel contains :

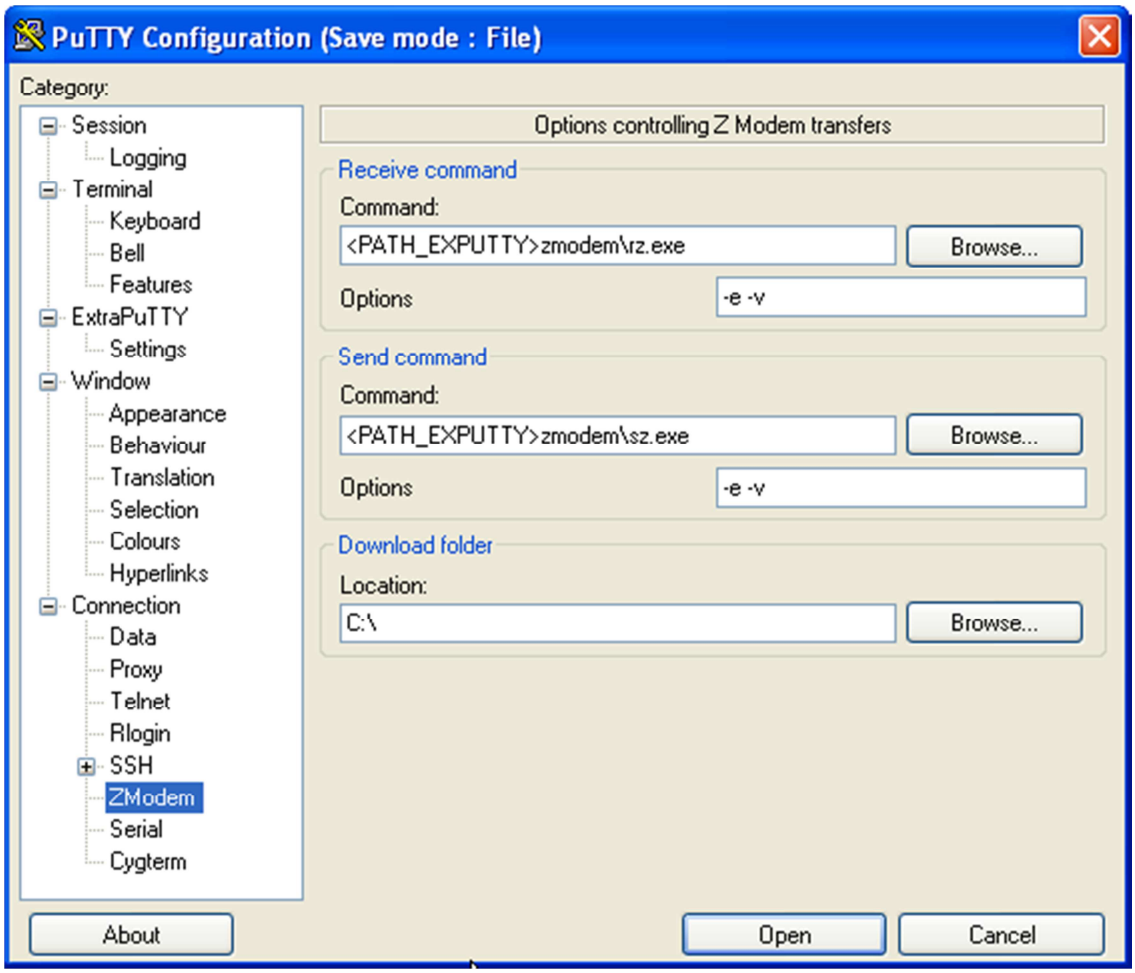


TimeStamp Format

specifier	Replaced by	Example
%a	Abbreviated weekday name *	Thu
%A	Full weekday name *	Thursday
%b	Abbreviated month name *	Aug
%B	Full month name *	August

%c	Date and time representation *	Thu Aug 23 14:55:02 2001
%d	Day of the month (01-31)	23
%G	GPS Week seconds	23
%H	Hour in 24h format (00-23)	14
%I	Hour in 12h format (01-12)	02
%j	Day of the year (001-366)	235
%m	Month as a decimal number (01-12)	08
%M	Minute (00-59)	55
%p	AM or PM designation	PM
%S	Second (00-61)	02
_MIL	Milliseconds (000-999)	678
%U	Week number with the first Sunday as the first day of week one (00-53)	33
%w	Weekday as a decimal number with Sunday as 0 (0-6)	4
%W	Week number with the first Monday as the first day of week one (00-53)	34

%x	Date representation *	**08/23/01**
%X	Time representation *	**14:55:02**
%y	Year, last two digits (**00-99**)	**01**
%Y	Year	**2001**
%Z	Timezone name or abbreviation	**CDT**
%%	A **%** sign	**%**



4.5 Environment Variables

ExtraPuTTY

The ExtraPuTTY DLL need the path of PuTTY. In order to do that you must create the environment variable "ExtraPuTTY" with the path of ExtraPuTTY.
(This environment variable is automatically set by the installer)

ExtraPuTTY_BOXSIZE

This variable can be used to change the size of the list box which contains saved sessions name of putty configuration window.(Default value AUTO)

Possible values :
- NORMAL : Display 7 saved sessions,
- AUTO : Display all saved sessions (min 7, max 30),
- DISPLAY_xy : Display xy sessions (with x = 1,2,3 , and y = 0,5)

ExtraPuTTY_MODE

By default, ExtraPuTTY uses tree directories structure to save its configuration (sessions, host keys, parameters). It's possible to save it into Windows registry database. To do that you just have to create the variable environment ExtraPuTTY_MODE :

Possible values :
- REG_MODE : uses Windows registry database,
- DIR_MODE : uses tree directories structure,

It is possible to copy all the configuration from the registry to file or file to registry.See chapter command line

Chapter 5: ExtraPuTTY Files

This chapter describes all the ExtraPuTTY files.

5.1 XML commands file

The XML commands file is used to perform the function "Automatic sequencing of commands".
This file can be create or update by two ways : ExtraPuTTY editor or manualy.

5.1.1 XML file description

This file used 8 XML TAGS, which are describe below :

ExtraPuTTYTreeCmd

Main TAG

```
<ExtraPuTTYTreeCmd>
</ExtraPuTTYTreeCmd>
```

Position

TAG which contains ExtraPuTTY or PuTTY window position

```
<Position name="ExtraPuTTY or PuTTY" use="0 or 1">
<btlr name="bottom" value="Coordonate"/>
<btlr name="top" value="Coordonate"/>
<btlr name="left" value="Coordonate"/>
<btlr name="right" value="Coordonate"/>
</Position>
```

Options

TAG which allows or not the edition of file.

```
<Options Locked="0 : not locked, 1 : locked"/>
```

File

TAG which contains the path of an other XML commands file.

```
<File name="PATH of other XML commands file (c:\File1.xml)"/>
```

Help

TAG to defined help file.

```
<Help Type="type of help file (chm,html help)" name="Path of help
file (c:\file.chm)"/>
```

To used help file you shall add the optional balise "help", described below.

root

TAG which contains a categorie name of commands

```
<root name="Categorie name" help="Used to display associate html help
page of command">
</root>
```

childroot

TAG which contains command description

```
<childroot name="Command description"  help="Used to display
associate html help page of command" type="1" cmd="Command" CRLF="0
or 1"/>
```

TAG which contains sequence commands name

```
<childroot name="Sequence commands name"  help="Used to display
associate html help page of command" type="2" loop="number of loop"
auto="run on startup or not">
</childroot>
```

childscenarii

TAG which contains sequence commands description

```
<childscenarii name="Command description"  help="Used to display
associate html help page of command" cmd="command" tempo="value"
mode="execute or skip" CRLF="0 or 1"/>
```

TAG which contains lua script link

```
<childroot name="Lua script name"  help="Used to display associate
html help page of command" type="3" path="Path and name of lua
script"/>
```

The CRLF field is used to create dynamic command. (CRLF = 0,the "enter" key is not send to allow to ending the command manually else yes)

The TAG "root,childroot,childscenarii" are composed of POLICE balises which are not defined here in order to keep a simple and clearly document.
If there TAG are not set in the file the default value are used. You can modify the police and saved it from the ExtraPuTTY window.

You can used **<PATH_EXPUTTY>** shortcut for relative path of putty.exe in File,Help and childroot (Type 3 ,lua cript) TAGS.

Note the following about parameters:

- The user parameters are represent in black.
- The fixe parameters are represent in orange.

5.1.2 Predefined commands

From ExtraPuTTY XML command file some predefined commands are available :

Virtual Key codes :

Commands	ExtraPutty Virtual Key codes
ADD	<VK_ADD>
APPS	<VK_APPS>
BACK	<VK_BACK>
CAPITAL	<VK_CAPITAL>
CLEAR	<VK_CLEAR>

DECIMAL	<VK_DECIMAL>
DELETE	<VK_DELETE>
DOWN	<VK_DOWN>
END	<VK_END>
RETURN	<VK_RETURN>
ESCAPE	<VK_ESCAPE>
F1	<VK_F1>
F2	<VK_F2>
F3	<VK_F3>
F4	<VK_F4>
F5	<VK_F5>
F6	<VK_F6>
F7	<VK_F7>
F8	<VK_F8>
F9	<VK_F9>

F10	<VK_F10>
F11	<VK_F11>
F12	<VK_F12>
HELP	<VK_HELP>
HOME	<VK_HOME>
INSERT	<VK_INSERT>
LEFT	<VK_LEFT>
NUMLOCK	<VK_NUMLOCK>
PRINT	<VK_PRINT>
RIGHT	<VK_RIGHT>
SCROLL	<VK_SCROLL>
SNAPSHOT	<VK_SNAPSHOT>
SUBTRACT	<VK_SUBTRACT>
TAB	<VK_TAB>
UP	<VK_UP>

CTRL	<CTRL>
ALT	<ALT>
ALTGR	<ALTGR>

Example :

- Send CTRL+C : <CTRL>C
- Send F1 TEST F11 ALT+F1 : <VK_F1>TEST<VK_F11><ALT><VK_F1>

PuTTY system commands :

System Commands	ExtraPutty predefined commands
Event log	EXT_SYS_CMD_SHOWLOG
New Session ...	EXT_SYS_CMD_NEWSESSION
Restart Session	EXT_SYS_CMD_RESTART
Duplicate Session	EXT_SYS_CMD_DUPSESSION
Saved Sessions	EXT_SYS_CMD_SAVEDSESSION
Change Settings ...	EXT_SYS_CMD_RECONF
Copy All to Clipboard	EXT_SYS_CMD_COPYALL
Clear Scrollback	EXT_SYS_CMD_CLEAR_SB
Reset Terminal	EXT_SYS_CMD_RESET_TERM

Full Screen	EXT_SYS_CMD_FULLSCREEN
Help	EXT_SYS_CMD_HELP
About PuTTY	EXT_SYS_CMD_ABOUT

ExtraPuTTY system commands :

System Commands	ExtraPutty predefined commands
Start Trace	EXT_SYS_CMD_START_TRACE
Stop Trace	EXT_SYS_CMD_STOP_TRACE
Edit Command File	EXT_SYS_CMD_EDIT_XML_FILE
About ExtraPuTTY	EXT_SYS_CMD_ABOUT_EXT
Close ExtraPuTTY window	EXT_SYS_CMD_CLOSE_WND

Extra commands :

Commands	ExtraPutty predefined commands
Display message box	EXT_MSGBOX_ + Text (EXT_MSGBOX_HELLO World => display HELLO World, the title of the message box it's the name of command)
Display input box	EXT_MSGBOX_INPUTBOX_ + Text (EXT_MSGBOX_INPUTBOX_Enter your password => display Enter your password, the title of the message box it's the name of command)

Telnet special commands :

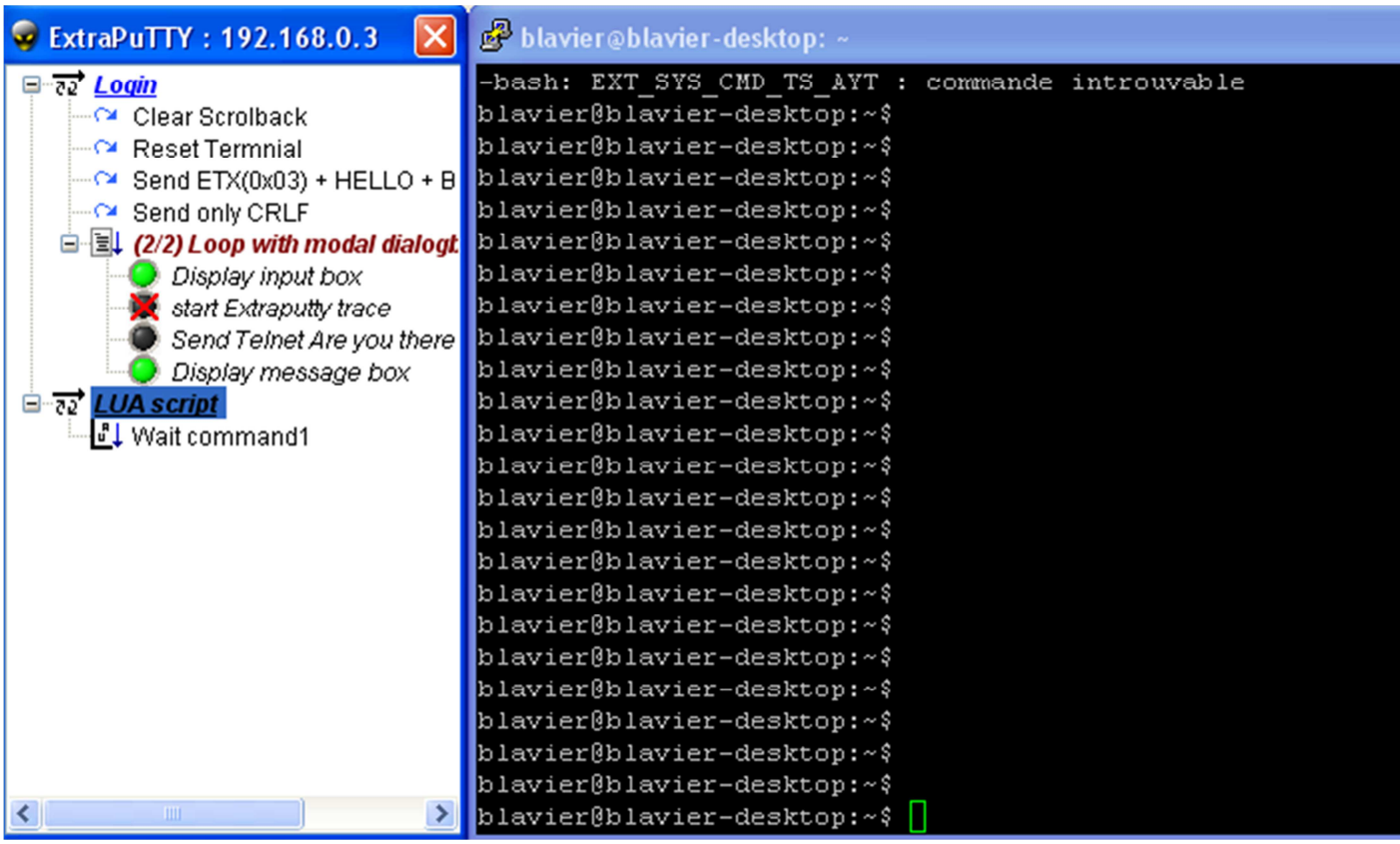
Commands	ExtraPutty predefined commands
Are you there	EXT_SYS_CMD_TS_AYT
Break	EXT_SYS_CMD_TS_BRK
Synch	EXT_SYS_CMD_TS_SYNCH
Erase Character	EXT_SYS_CMD_TS_EC
Erase Line	EXT_SYS_CMD_TS_EL
Go Ahead	EXT_SYS_CMD_TS_GA
No Operation	EXT_SYS_CMD_TS_NOP
Abort Process	EXT_SYS_CMD_TS_ABORT
Abort Output	EXT_SYS_CMD_TS_AO
Interrupt Process	EXT_SYS_CMD_TS_IP
Suspend Process	EXT_SYS_CMD_TS_SUSP
End Of Record	EXT_SYS_CMD_TS_EOR
End Of File	EXT_SYS_CMD_TS_EOF
End Of Line	EXT_SYS_CMD_TS_EOL

5.1.3 XML file example

XML File

```
<?xml version="1.0"?>
<!--Registry file for ExtraPuTTY, ExtraPuTTY Copyright 2010-->
<ExtraPuTTYTreeCmd>
      <Position name="ExtraPuTTY" use="1">
          <btlr name="bottom" value="639"/>
          <btlr name="top" value="81"/>
          <btlr name="left" value="105"/>
          <btlr name="right" value="341"/>
      </Position>
      <Position name="PuTTY" use="1">
          <btlr name="bottom" value="501"/>
          <btlr name="top" value="81"/>
          <btlr name="left" value="341"/>
          <btlr name="right" value="1008"/>
      </Position>
      <Options Locked="0">
      <Help Type="chm" name="<PATH_EXPUTTY>ExtraPutty.chm"/>
      <root name="Login"  help="ExtraPuTTY_Start.html">
          <childroot name="Clear Scrolback" type="1"
cmd="EXT_SYS_CMD_CLEAR_SB" CRLF="1"/>
          <childroot name="Reset Termnial"
help="ExtraPuTTY_Using.html" type="1" cmd="EXT_SYS_CMD_RESET_TERM"
CRLF="1"/>
          <childroot name="Send ETX(0x03) + HELLO + BEL(0x07)"
type="1" cmd="EXT_SPECHAR_ETX_HELLO_BEL" CRLF="1"/>
          <childroot name="Send only CRLF" type="1"
cmd="EXT_SPECHAR_CR_LF" CRLF="0"/>
          <childroot name="Loop with modal dialogbox" type="2"
loop="2" auto="1">
              <childscenarii name="Display input box"
cmd="EXT_MSGBOX_INPUTBOX_Enter your login" tempo="500" mode="execute"
CRLF="1"/>
              <childscenarii name="start Extraputty trace"
cmd="EXT_SYS_CMD_START_TRACE" tempo="500" mode="skip" CRLF="1"/>
              <childscenarii name="Send Telnet special command Are you
there" cmd="EXT_SYS_CMD_TS_AYT" tempo="1000" mode="execute"
CRLF="1"/>
              <childscenarii name="Display message box"
cmd="EXT_MSGBOX_Tests MessageBox" tempo="1000" mode="execute"
CRLF="1"/>
          </childroot>
      </root>
      <root name="LUA script"  help="ExtraPuTTY_Using_Step_lua.html">
      <childroot name="Wait command1"
help="ExtraPuTTY_Using_Step_lua.html" type="3"
path="<PATH_EXPUTTY>Examples\script.lua"/>
      </root>
      <File name="C:\File1.xml">
      <File name="<PATH_EXPUTTY>Examples\ExtraPuTTYCmd.xml">
</ExtraPuTTYTreeCmd>
```

Result with ExtraPuTTY editor



5.2 Report files

The ExtraPuTTY report file can be activate from PuTTY panel or during session with system menu of ExtraPuTTY window or by changing the settings of connection.
This report can trace all exchanges (Command,Reply) with the distant target.This report is available for ExtraPuTTY and PuTTY.
For extreme detail and reference purposes, chapter 4 is likely to contain more information.

5.3 TestStand Report files

This report is generate by TestStand. Note : The Data received from distant target is added in the report by the ExtraPuTTY TestStand Step. For extreme detail and reference purposes, chapter 6.1.2 is likely to contain more information.

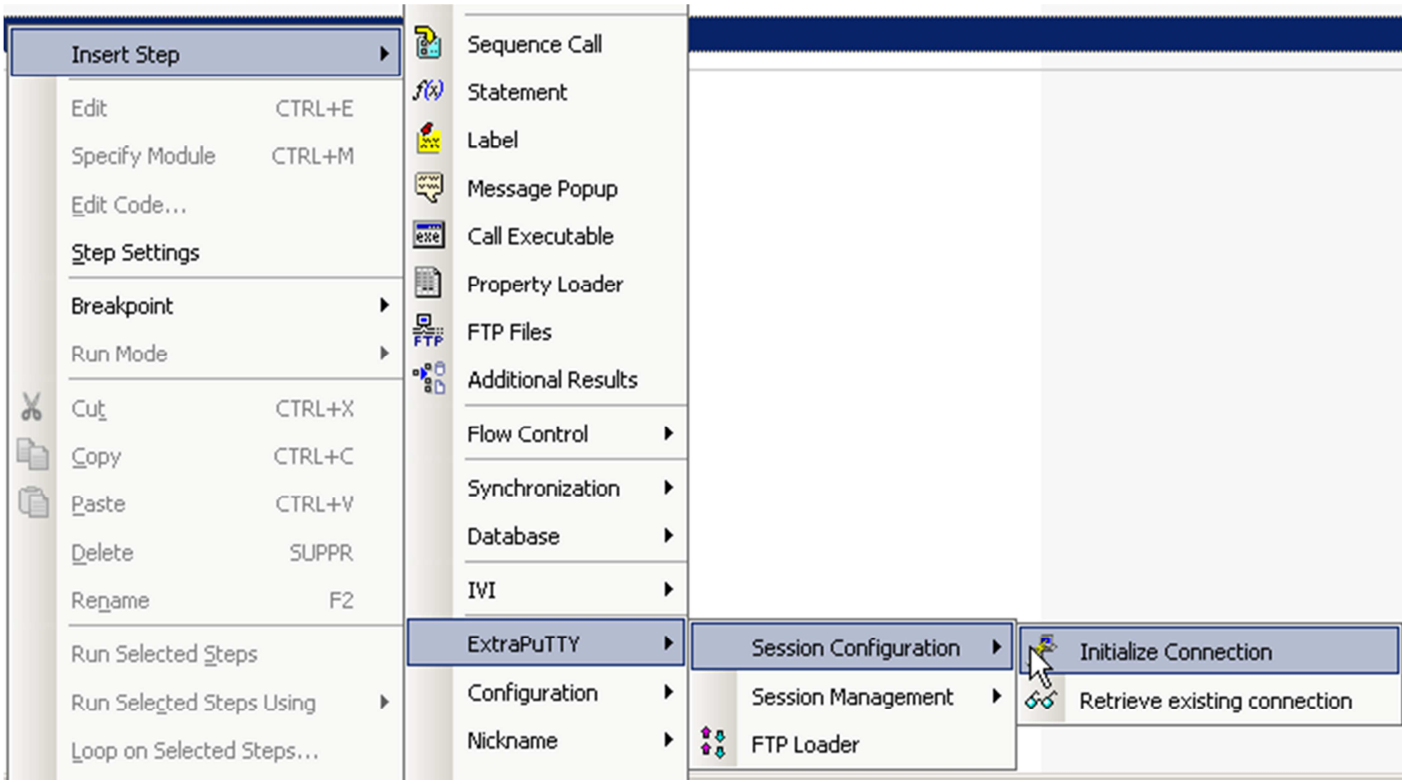
Chapter 6: How to use with TestStand

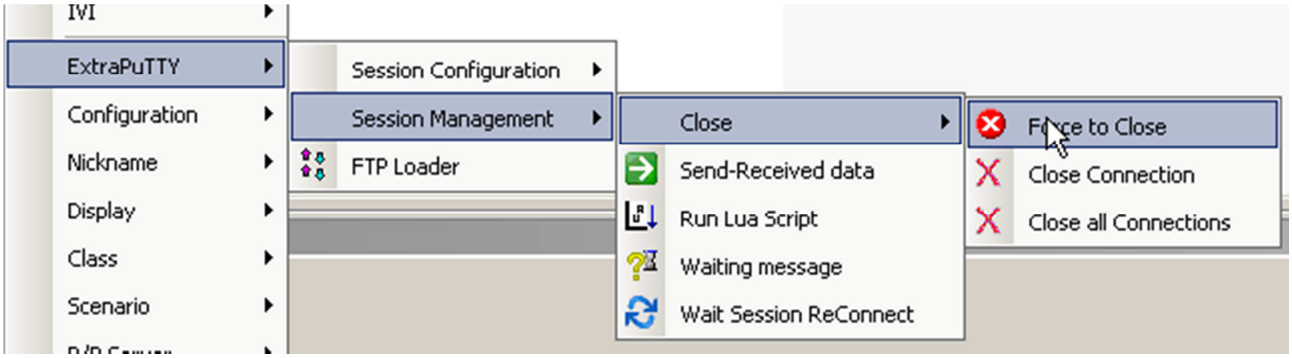
This chapter provides a general introduction to some more advanced TestStand features of ExtraPuTTY.

6.1 : TestStand Steps description and examples

To manage exchange with remote device used thes steps for :

1. Initialize Connection.
2. Send-Received Data.
3. Waiting message.
4. Close connection.
5. Close all connections.
6. Transfert files.
7. Waiting restart of remote device.
8. Run lua script.
9. Retrieve existing connection and keep a live connection.
10. Force to close connection.





6.1.1 : "Initialize Connection" Step

"Intialisation Connection" Step is used to establish the connection, (used "Connexion_F" function of extraputty DLL) :

Parameters	Description
TargetName	TargetName or serial line name or PuttySession Name (in this case Protocol must be equal to 4).
ConnexionId	Connection ID.
Login	optinonal parameter.
Password	optinonal parameter.
ShowPuTTY	TRUE: Putty Terminal is display, FALSE: not display.
Protocol	0:Telnet,1:SSH,2:Rlogin,3:Raw,4:LoadPutty Session,5:Serial.
PortNumber	If the parameter is set to 0, the value of the default settings of putty is used. This field correspond to the speed for serial line protocol.
GenerateReport	1:extraputty report activate,0:Not activate.

CRLF	1:0A0D,2:0D0A,3:0A,4:0D.
NewCRLF	String used to replace the CRLF parameter find in data (for example CRLF = 1,NewCRLF :)
ReportFileData	Name and path of report generate by TestStand in order to generate extraputty report.
arg10	State of the step (Pass or failed).
arg11	Step result report.
arg12	Error occured during step (True or false).
arg13	Error code.
arg14	Error Message.

6.1.2 : "Send-Received data" Step

"Send-Received data" Step is used to send and/or received data, (used "SendRcvData_F" function of extraputty DLL) :

Parameters	Description
ConnexionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
Command	Data to send to the target.
Title	Title of your command,used only if extraputty report is activate.

Comments	Comments of your command,used only if extraputty report is activate.
TimeCapture	Time used to capture the reply data in ms.
DataRcv	Buffer which contain your data if TimerCapture is > 0.
MaxSizeofData	Size of DataRcv Buffer or maximum data size in DataRcv
Settings	Bit fields of settings (2^0 : CRLF (0 send,1 not send),...reserved)
arg7	State of the step (Pass or failed).
arg8	Step result report.
arg9	Error occured during step (True or false).
arg10	Error code.
arg11	Error Message.

6.1.3 : "Waiting Message" Step

"Waiting Message" Step is used to wait one message on putty terminal, (used "WaitingMessage_F" function of extraputty DLL) :

Parameters	Description
ConnexionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
Message	Message to wait on putty terminal.

TimeCapture	Timeout value of wait in ms.
arg6	State of the step (Pass or failed).
arg7	Step result report.
arg8	Error occured during step (True or false).
arg9	Error code.
arg10	Error Message.

6.1.4 : "Close Connection" Step

"Close Connection" Step is used to close the current connection, (used "CloseConnexion_F" function of extraputty DLL) :

Parameters	Description
ConnexionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
arg2	State of the step (Pass or failed).
arg3	Step result report.
arg4	Error occured during step (True or false).
arg5	Error code.
arg6	Error Message.

6.1.5 : "Close All Connections" Step

"Close all Connections" Step is used to close all connections, (used "CloseAllConnexion_F" function of extraputty DLL) :

Parameters	Description
arg1	State of the step (Pass or failed).
arg2	Step result report.
arg3	Error occured during step (True or false).
arg4	Error code.
arg5	Error Message.

6.1.6 : "FTPLoader" Step

"FTP Loader" Step is used to transferring files with FTP protocol, (used "FtpLoader_F" function of extraputty DLL) :

Parameters	Description
Target	Hostname of target.
FilePath	path of the files to upload.
DestPath	destination of the files on target.
Show	Display FTPLoader terminal.

Login	login (size 1 - 99).
Password	password (size 1 - 99).
TransfertMode	transfert mode (BINARY or ASCII).
verbose	explain what is being done (verbose) (param show shall be set to TRUE).
arg9	State of the step (Pass or failed).
arg10	Step result report.
arg11	Error occured during step (True or false).
arg12	Error code.
arg13	Error Message.

6.1.7 : "Wait Session ReConnect" Step

"Wait Session ReConnect" Step used to wait automatic session ReConnect only when "Close window on exit" parameter of session is configured to "Never,Auto-Connect", (used "WaitReConnect_F" function of extraputty DLL) :

Parameters	Description
ConnectionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
TimeOut	Timeout value of wait in ms.

arg6	State of the step (Pass or failed).
arg7	Step result report.
arg8	Error occured during step (True or false).
arg9	Error code.
arg10	Error Message.

6.1.8 : "Run Lua script" Step

"Run Lua script" Step used to run lua script, see Lua description to get all ExtraPuTTY lua APIs available.(used "lua_dofile_F" function of extraputty DLL)

Parameters	Description
ConnectionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
PathFile	Path and name of lua script file.
arg6	State of the step (Pass or failed).
arg7	Step result report.
arg8	Error occured during step (True or false).
arg9	Error code.
arg10	Error Message.

6.1.9 : "Retrieve Existing Connection" Step

"Retrieve Existing Connection" step used to reconnect to a existing session and free AND to keep a live the connection if the DLL is unloaded or if the Close functions are called.(used "RetrieveExistingConnection_F" function of extraputty DLL) The only way to close an existing session is to call the step Force To Close.

Parameter	Description
ConnectionHandle	Handle of the connection (value : 1 to 255). Be careful the connection handle and connection id (used with all other functions) are not the same, the connection handle is used for a global identification of a connection where as the connection id is local at the DLL.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

Important notes :

- Only one client by connection.
- If more than one existing connections have the same handle, the connection is established with the first one.
- This function do not replace the Connexion function which must be called each time after.

6.1.10 : "Force To Close Connection" Step

"Force To Close" step used to force the closure of the connection even if extraputty is configured to keep alive the connection (ie..html">"Retrieve Existing Connection" step has been called). (used "ForceToClose_F" function of extraputty DLL).

Parameter	Description

ConnectionId	ConnectionId set by "Initialize Connection" step function (shall be > 0).
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

Example of using :

7.2.2.11 Example of using

- TestStand sequence example.

Chapter 7: How to use with Win32 applications

This chapter provides a general description of all functions available with ExtraPuTTY DLL in order to communicate on Telnet,Serial Link,SSH,Rlogin, Raw protocols and Cygterm terminal.

7.1 ExtraPuTTY DLL

The ExtraPuTTY DLL is composed with two groups of functions :

- One goup of functions with basic parameters
- Second group of functions with complexes parameters (Design for TestStand using)

7.1.1 DLL compile options

The DLL is compiled with calling convention "__stdcall" and with struct member alignment "16 bytes"

7.1.2 How to used ExtraPuTTY without ExtraPuTTY installer

The ExtraPuTTY DLL need the path of *Extra*PuTTY.
In order to do that you have two choices :
1.Create the environment variable "ExtraPuTTY" with the path of ExtraPuTTY.
2.Put putty.exe in the same folder that the DLL, in this case do not forget lua DLL.

7.2.1 Functions with basic parameters

7 functions are available :

1. Using "Connexion" function to intialized the connexion.
2. Using "SendRcvData" function to send\received data.
3. Using "WaitingMessage" function to wait one message on putty terminal.
4. Using "WaitReConnect" function to wait automatic session ReConnect.
5. Using "CloseConnexion" function to closed the connexion.
6. Using "CloseAllConnexion" function to closed all connections.

7. Using "FtpLoader" function to transferring files.
8. Using "lua_dofile" function to run lua script.
9. Using "UploadFiles" function to upload file through xmodem,ymodem or FTP protocols.
10. Using "RetrieveExistingConnection" function to to retrieve an existing connection and to keep a live a connection.
11. Using "ForceToClose" function to force the closure of the connection.

7.2.1.1 "Connexion" function

Function used to establish the connection on remote device.

Syntax

```
int Connexion(char *TargetName,unsigned long *ConnectionId,char
*Login,char *Password,bool ShowPuTTY,long Protocol,
     unsigned long PortNumber,long GenerateReport,int
     *CallBackRcvData,unsigned long SpecSettings);
```

Parameters

Parameter	Description
TargetName	TargetName or PuttySession Name (in this case Protocol must be equalt to 4).
ConnectionId	Connection ID.
Login	Optional parameter.
Password	Optional parameter.
ShowPuTTY	TRUE: Putty Terminal is display, FALSE: not display.
Protocol	0:Telnet,1:SSH,2:Rlogin,3:Raw,4:LoadPutty Session,5:Serial Link,6:Cygterm.
PortNumber	If the parameter is set to 0, the value of the default settings of putty is used. This field correspond to the speed for serial line protocol.
GenerateReport	1:extraputty report activate,0:Not activate.

CallBackRcvData	Optional parameter,callback used to receive data from putty terminal, Syntax : **int CallBackRcvData(unsigned long ConnectionId,char *buf, DWORD size,DWORD ConnectionStatus)** ConnectionStatus : • 0 : Connection open • 1 : Connection close by host
SpecSettings	Specific settings, bit field : • 2^0 : Do not wait login prompt, connection has no prompt • 2^1 : Dynamically starting putty log • 2^2 : SSH V1 otherwise SSH V2 • 2^3 ....: Reserved

Return Values

Return Value	Description
0	Indicates that the function successfully to establish the connection.
<> 0	Indicates that the function was unable to establish the connection.
1	ExtraPutty environment vairable doesn't exist.
2	Wrong value of parameter Protocol (0-6).
3	Putty.exe doesn't exist at the place defined in ExtraPutty environment variable.
4	the path set in ExtraPutty environment variable doesn't exist.
5	putty is stop before the connection is established.
6	Connection failed with the target (host doesn't exist,timeout on connection,connection refused by the host...)

7	Timeout to send password
8	Internal error (memory allocation etc ....),impossible to perform the function.

7.2.1.2 "SendRcvData" function

Function used to send\received data with the distant target on Telnet,SSH,Rlogin and Raw protocols.

Syntax

```
int SendRcvData(unsigned long ConnectionId,char *Command,char
*Title,char *Comments,long TimeCapture,char **DataRcv,long
MaxSizeofData,unsigned long settings);
```

The old version of "SendRcvData" function is also available, named "SendRcvData_O". The difference is on "DataRcv" parameter type :

```
int SendRcvData_O(unsigned long ConnexionId,char *Command,char
*Title,char *Comments,long TimeCapture,char *DataRcv,long
MaxSizeofData,unsigned long settings);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
Command	Data to send on the target, if the parameter is null the function is configured only in reception way.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.
TimeCapture	Time used to capture the reply data in ms.
DataRcv	Buffer which contains the data received if TimerCapture is > 0.

MaxSizeofData	Size of DataRcv Buffer or maximum data size in DataRcv (1 < MaxSizeofData < 20 000 000)
Settings	Bit fields of settings (2^0 : CRLF (0 send,1 not send),2^1 : Virtual key codes (0 no virtual key codes in command,1 yes)...reserved) See FAQ page for a description of all virtual keys codes.

Return Values

Return Value	Description
0	Indicates that the function successfully to Send\Received data.
<> 0	Indicates that the function was unable to Send-Received data.
1	Connection not established with the target.
2	The field 'RcvTelnetData' is NULL.
3	The field 'MaxSizeofData' is invalid (< 0 or > 20 000 000).
4	The function is used in reception but timeCapture is no set.
5	Internal error (memory allocation etc ....),impossible to perform the function.

7.2.1.3 "WaitingMessage" function

Function used to wait one message on putty terminal.

Syntax

```
int WaitingMessage(unsigned long ConnectionId,char *Message,char *Title,char *Comments,long TimeCapture);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
Message	Message to wait on putty terminal.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.
TimeCapture	Timeout value of wait in ms.

Return Values

Return Value	Description
0	Indicates that the indicate message was displayed on putty terminal.
<> 0	Indicates that the function was unable to wait or that the timeout is reached.
1	Connection not established with the target.
2	The field "WaitMessage" is not set.
3	The field "TimeCapture" is invalid (shall be > 0).
4	Indicates that the timeout is reached without received corresponding data defined in "WaitMessage" field.
5	Internal error (memory allocation etc ....),impossible to perform the function.

7.2.1.4 "CloseConnexion" function

Function used to close the connection with the distant target.

Syntax

```
int CloseConnexion(unsigned long ConnectionId);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).

Return Values

Return Value	Description
0	Indicates that the function successfully to close the connection.
<> 0	Indicates that the function was unable to close the connection.
1	Connection not established with the target.
2	Impossible to stop putty process.
3	Internal error (memory allocation etc ....),impossible to perform the function.

7.2.1.5 "CloseAllConnexion" function

Function used to close all connections.

Syntax

```
int CloseAllConnexion();
```

Parameters

Parameter	Description
None	NA.

Return Values

Return Value	Description
0	Indicates that the function successfully to close all connections.
<> 0	Indicates that the function was unable to close all connections.
1	Internal error (memory allocation etc ....),impossible to perform the function.

7.2.1.6 "FtpLoader" function

Function is used to transferring files with FTP protocol.

Syntax

```
void FtpLoader(char *TargetName,char *FilePath,char *DestPath,bool Showterm,char *User,char *Pass,unsigned long TransfertMode,bool verbose);
```

Parameters

Parameters	Description
Target	Hostname of target.
FilePath	path of the files to upload.
DestPath	destination of the files on target.

Show	Display FTPLoader terminal.
Login	login (size 1 - 99).
Password	password (size 1 - 99).
TransfertMode	transfert mode (BINARY or ASCII).
verbose	explain what is being done (verbose) (param show shall be set to TRUE).

Return Values

Return Value	Description
0	All files are uploaded on the target.
<> 0	Impossible to transfert the files due to a bad parameters or at least 1 files is KO
1	Bad parameter : TargetName not set.
2	Bad parameter : File path not set.
3	Bad parameter : Destination path not set.
4	Bad parameter : The FTPLoader.exe is not found.
5	Bad parameter : Path of FTPLoader.exe file not found.
6	At least one file has not been correctly transfert.

6	ExtraPuTTY environment variable is missing.

7.2.1.7 "WaitReConnect" function

Function used to wait automatic session ReConnect only when "Close window on exit" parameter of session is configured to "Never,Auto-Connect".
This function can be used after a shutdown command instead of a waiting tempo in order to improve waiting time performance.

Syntax

```
int WaitReConnect(unsigned long ConnectionId,long TimeOut);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
TimeOut	Timeout value of wait in ms.

Return Values

Return Value	Description
-1	Indicates that the target has not been reseted.
0	Indicates that the target is restarted.
<> 0	Indicates error or that the timeout is reached.
1	Connection not established with the target.
2	The field "TimeCapture" is invalid (shall be > 0).

3	Internal error (memory allocation etc ....),impossible to perform the function.
4	Indicates that the timeout is reached before the target has been restarted.
5	Indicates that the parameter "Close window on exit" parameter of session is not configured to "Never,Auto-Connect".

7.2.1.8 "lua_dofile" function

Function used to run lua script.

Syntax

```
int lua_dofile(unsigned long ConnectionId,char *PathFile,char *Title,char *Comments);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
PathFile	Path and name of lua script file.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.

Return Values

Return Value	Description
0	No error.

<> 0	Error.
1	Connection not established with the target.
2	The field "PathFile" is invalid.
3	Internal error (memory allocation etc ....),impossible to perform the function.
4	Lua scripting error.

7.2.1.9 "RetrieveExistingConnection" function

Function used to reconnect to a existing session AND to keep a live the connection if the DLL is unloaded or if the Close functions are called. The only way to close an existing session is to call the function ForceToClose.

Syntax

```
int RetrieveExistingConnection(unsigned char ConnectionHandle);
```

Parameters

Parameter	Description
ConnectionHandle	Handle of the connection (value : 1 to 255). Be careful the connection handle and connection id (used with all other functions) are not the same, the connection handle is used for a global identification of a connection where as the connection id is local at the DLL.

Return Values

Return Value	Description
0	Connection establish. In this case a call of Connexion function is used to get a connection ID and to reconfigure the callback all other field are not taken into account.

1	Connection is not free (one client by connection) or do not exist. In this case when the Connexion function is called, a new connection is made with ConnectionHandle.
< 0	Error.
-1	The field "ConnectionHandle" is invalid.

Important notes :

- Only one client by connection.
- If more than one existing connections have the same handle, the connection is established with the first one.
- This function do not replace the Connexion function which must be called each time after.

7.2.1.10 "ForceToClose" function

Function used to force the closure of the connection even if extraputty is configured to keep alive the connection (ie.RetrieveExistingConnection has been called).

Syntax

```
int ForceToClose(unsigned long ConnectionId);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).

Return Values

Return Value	Description
0	Connection establish. In this case a call of Connexion function is used to get a connection ID and to reconfigure the callback all other field are not taken into account.
1	The Handle point to a connection which is not open or not free (one client by connection). In this case when the Connexion function is called, a new connection is made

	with ConnectionHandle.
< 0	Error.
-1	The field "ConnectionHandle" is invalid.

7.2.1.11 "UploadFiles" function

Function used to upload file through xmodem,ymodem or FTP protocols.

Syntax

```
int UploadFiles(unsigned long ConnectionId,int ProtocolType,char
*PathFile,char *Title,char *Comments);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
ProtocolType	Protocol to use for upload (0 : xmodem, 1 : xmodem-1k, 2 : ymodem).
PathFile	Path and name of file to upload.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.

7.2.1.11 Examples of using (MFC,.NET(VB,C#),VB6,Labview)

Example of using:

- MFC Example.
- VB.NET Example.
- CSharp.NET Example.
- VB6 Example.
- LabView Example.

7.2.2 Functions with complexes parameters (Design for TestStand)

7 functions are available :

1. Using "Connexion_F" function to intialized the connection.
2. Using "SendRcvData_F" function to send\received data.
3. Using "WaitingMessage_F" function to wait one message on putty terminal.
4. Using "WaitReConnect_F" function to wait automatic session ReConnect.
5. Using "CloseConnexion_F" function to closed the connection.
6. Using "CloseAllConnexion_F" function to closed all connections.
7. Using "FtpLoader_F" function to transferring files.
8. Using "lua_dofile_F" function to run lua script.
9. Using "UploadFiles_F" function to upload file through xmodem,ymodem or FTP protocols.
10. Using "RetrieveExistingConnection_F" function to to retrieve an existing connection and to keep a live a connection.
11. Using "ForceToClose_F" function to force the closure of the connection.

7.2.2.1 "Connexion_F" function

Function used to establish the connection with remote device.

Syntax

```
void Connexion_F(char *TargetName,unsigned long *ConnectionId,char
*Login,char *Password,bool ShowPuTTY,long Protocol,unsigned long
PortNumber,long GenerateReport,
                long  TypeCRLF,char *NewCRLF,char
*ReportFileData,unsigned long SpecSettings, bool &result, char
reportText[1024],
```

```
                bool &errorOccurred, long &errorCode, char
errorMsg[1024]);
```

Parameters

Parameter	Description
TargetName	TargetName or PuttySession Name (in this case Protocol must be equalt to 4).
ConnectionId	Connection ID.
Login	optinonal parameter.
Password	optinonal parameter.
ShowPuTTY	TRUE: Putty Terminal is display, FALSE: not display.
Protocol	0:Telnet,1:SSH,2:Rlogin,3:Raw,4:LoadPutty Session,5:Serial Link,6:Cygterm.
PortNumber	If the parameter is set to 0, the value of the default settings of putty is used. This field correspond to the speed for serial line protocol.
GenerateReport	1:extraputty report activate,0:Not activate.
CRLF	1:0A0D,2:0DOA,3:0A,4:0D.
NewCRLF	String used to replace the CRLF parameter find in data (for example CRLF = 1,NewCRLF :)
ReportFileData	Name and path of report generate by TestStand in order to generate extraputty report.
SpecSettings	Specific settings, bit field :

	• 2^0 : Do not wait login prompt, connection has no prompt • 2^1 : Dynamically starting putty log • 2^2 : SSH V1 otherwise SSH V2 • 2^3 ....: Reserved
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Parameter already set in step. Error code.
errorMsg	Parameter already set in step. Error Message.

7.2.2.2 "SendRcvData_F" function

Function used to send\receive data with the distant target on Telnet,SSH,Rlogin and Raw protocols.

Syntax

```
void SendRcvData_F(unsigned long ConnexionId,char *Command,char
*Title,char *Comments,long TimeCapture,char DataRcv[],long
MaxSizeofData,unsigned long settings,
                   bool &result, char reportText[1024],bool
&errorOccurred, long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnexionId	Connection ID set by Connection Function (shall be > 0).
Command	Data to send on the target, if the parameter is null the function is configured only in reception way.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.
TimeCapture	Time used to capture the reply data in ms.

DataRcv	Buffer which contains the data received if TimerCapture is > 0.
MaxSizeofData	Size of DataRcv Buffer or maximum data size in DataRcv (1 < MaxSizeofData < 20 000 000)
Settings	Bit fields of settings (2^0 : CRLF (0 send,1 not send),2^1 : Virtual key codes (0 no virtual key codes in command,1 yes)...reserved) See FAQ page for a description of all virtual keys codes.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.3 "WaitingMessage_F" function

Function used to wait one message on putty terminal.

Syntax

```
void WaitingMessage_F(unsigned long ConnectionId,char *Message,char
*Title,char *Comments,long TimeCapture,
                      bool &result, char reportText[1024],bool
&errorOccurred, long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).

Message	Message to wait on putty terminal.
Title	Title of your command,used only if extraputty report is activate.
Comments	Comments of your command,used only if extraputty report is activate.
TimeCapture	Timeout value of wait in ms.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.4 "CloseConnexion_F" function

Function used to close the connection.

Syntax

```
void CloseConnexion_F(unsigned long ConnectionId,bool &result, char
reportText[1024],bool &errorOccurred, long &errorCode,
                      char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).

result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.5 "CloseAllConnexion_F" function

Function used to close all connections.

Syntax

```
void CloseAllConnexion_F(bool &result, char reportText[1024],bool &errorOccurred, long &errorCode,char errorMsg[1024]);
```

Parameters

Parameter	Description
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.6 "FtpLoader_F" function

Function is used to transferring files with FTP protocol.

Syntax

```
void FtpLoader_F(char *TargetName,char *FilePath,char *DestPath,bool
Showterm,char *User,char *Pass,unsigned long TransfertMode,bool
verbose,
                 bool &result, char reportText[1024],bool
&errorOccurred, long &errorCode, char errorMsg[1024]);
```

Parameters

Parameters	Description
Target	Hostname of target.
FilePath	path of the files to upload.
DestPath	destination of the files on target.
Show	Display FTPLoader terminal.
Login	login (size 1 - 99).
Password	password (size 1 - 99).
TransfertMode	transfert mode (BINARY or ASCII).
verbose	explain what is being done (verbose) (param show shall be set to TRUE).
arg9	State of the step (Pass or failed).
arg10	Step result report.

arg11	Error occured during step (True or false).
arg12	Error code.
arg13	Error Message.

7.2.2.7 "WaitReConnect_F" function

Function used to wait automatic session ReConnect only when "Close window on exit" parameter of session is configured to "Never,Auto-Connect".
This function can be used after a shutdown command instead of a waiting tempo in order to improve waiting time performance.

Syntax

```
void WaitReConnect_F(unsigned long ConnectionId,long TimeOut,
                     bool &result, char reportText[1024],bool
&errorOccurred, long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
TimeOut	Timeout value of wait in ms.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.8 "lua_dofile_F" function

Function used to run lua script.

Syntax

```
int lua_dofile_F(unsigned long ConnectionId,char *PathFile,bool
&result, char reportText[1024],bool &errorOccurred,
  long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnectionId	ConnectionId set by Connexion function (shall be > 0).
PathFile	Path and name of lua script file.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

7.2.2.9 "RetrieveExistingConnection_F" function

Function used to reconnect to a existing session and free AND to keep a live the connection if the DLL is unloaded or if the Close functions are called. The only way to close an existing session is to call the function ForceToClose_F.

Syntax

```
int RetrieveExistingConnection_F(unsigned char ConnectionHandle,bool
&result, char reportText[1024],bool &errorOccurred,
  long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description
ConnectionHandle	Handle of the connection (value : 1 to 255). Be careful the connection handle and connection id (used with all other functions) are not the same, the connection handle is used for a global identification of a connection where as the connection id is local at the DLL.
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

Important notes :

- Only one client by connection.
- If more than one existing connections have the same handle, the connection is established with the first one.
- This function do not replace the Connexion function which must be called each time after.

7.2.2.10 "ForceToClose_F" function

Function used to force the closure of the connection even if extraputty is configured to keep alive the connection (ie.RetrieveExistingConnection_F has been called).

Syntax

```
int ForceToClose_F(unsigned long ConnectionId,bool &result, char
reportText[1024],bool &errorOccurred,
  long &errorCode, char errorMsg[1024]);
```

Parameters

Parameter	Description

ConnectionId	ConnectionId set by Connexion function (shall be > 0).
result	State of the step (Pass or failed).
reportText	Step result report.
errorOccurred	Error occured during step (True or false).
errorCode	Error code.
errorMsg	Error Message.

Chapter 8: Using FTPLoader to transfer files

This chapter provides a general introduction to some more advanced features of FTPLoader.
FTPLoader, the ExtraPuTTY FTP client, is a tool for transferring files between computers without using an SSH connection

Chapter 8.1 : Starting FTPLoader

The usual way to start FTPLoader is from a command prompt. To do this, it will need either to be on your PATH or in your current directory. To add the directory containing PSFTP to your PATH environment variable, type into the console window:

set PATH=C:\path\to\extraputty\directory;%PATH%

FTPLoader has no complex command-line syntax; you just specify a host name and perhaps a user name and path of file to transfert and path on target:

ftploader -tserver.com -fc:\transfert -d/ (creates all subfolder and copy all files to /)

ftploader -tserver.com -fc:\transfert\test.c -d/ (copy test.c to /)

ftploader -tserver.com -gc:\download* -d/ (download all files present in /)

Alternatively, if you just type ftploader on its own (or double-click the ftploader icon in the Windows GUI), you will see the ftploader prompt, and a message telling you an error occured due to a bad arguments :

C:\>FTPLoader
ftploader: missing operand
Try 'ftploader --help' for more information

FTPLoader Usage : ftploader <Commands> <Switches>[data]

<Commands>	Description
-v	explain what is being done (verbose)
--help	display this help and exit
--version	output version information and exit
<Switches>	**Description**
-t[target]	Set the Target name (size 1 - 99)
-m[mode]	Set the transfert mode (BINARY or ASCII)
-f[path]	Set file or path of files to upload
-g[path]	Set local current directory for download
-d[destination]	Set the destination of files on target, with option /g dertmine the path of file to download (not recursif)
-u[user]	Set the login (size 1 - 99)
-p[password]	Set the password (size 1 - 99)

By default, Bynary transfert mode and Anonymous connection is used.
Limitation :
* number of folders : 10000
* number of files : 10000
* size of path : 2048

Report bugs to bugs-ftploader@extraputty.com

Chapter 9: ExtraPuTTY installer

The ExtraPuTTY installer is make with NSIS for everything. (executables,dll,teststand,etc..)

9.1 Personalize ExtraPuTTY installer

The ExtraPuTTY's installer offers the possibility to add some files like your ExtraPuTTY commands file or other.
To do that the file "install.ini" shall be present in the same folder that the installer, with the syntax describe below :

Syntax

```
[NB_FEATURES]
Number=x
[FEATURE_y]
Path = "the output path(*)"
LocName = "file(s) to be extracted to the current output path, shall be in the same folder
          that the installer"
```

x : represents the total file number
y : represents the number of files (1,2 ...), shall not exceed x

(*): the installer creates the output path (recursively if necessary), if it does not exist.
Must be a full pathname, usually its can be a predifined path INSTDIR - DESKTOP - WINDIR - SYSDIR - TEMP - DOCUMENTS.
INSTDIR = The installation directory selected with ExtraPuTTY installer.
DESKTOP = The windows desktop directory (usually C:\windows\desktop but detected at runtime).
WINDIR = The windows directory (usually C:\windows or C:\winnt but detected at runtime)
SYSDIR = The windows system directory (usually C:\windows\system or C:\winnt\system32 but detected at runtime)
TEMP = The system temporary directory (usually C:\windows\temp but detected at runtime)
DOCUMENTS = The "My Documents" directory (usually C:\Documents and Settings\Users\My Documents)

Example

```
[NB_FEATURES]
Number=2
[FEATURE_1]
Path = "WINDIR"
LocName = "test_install\test_file.txt"
[FEATURE_2]
Path = "c:\testFolder"
LocName = "test_install\test_file1.txt"
```

Appendix A: ExtraPuTTY FAQ

What is ExtraPuTTY ?

ExtraPuTTY is a fork of PuTTY which is client program for the SSH, Telnet and Rlogin network protocols.
Principals features :

- allow win32 application to send,receive data on Telnet, SSH, RLogin and Raw protocols.
- TestStand steps to send,receive data on Telnet, SSH, RLogin and Raw protocols.(installed by the windows installer)
- automatic sequencing of commands.
- generate report of all exchanges (Command,reply) in HTML or Text format.

Features supported in ExtraPuTTY

In general, if you want to know if ExtraPuTTY or PuTTY supports a particular feature, you should look for it on the ExtraPuTTY web site or PuTTY web site. In particular:

- try the changes page, and see if you can find the feature on there. If a feature is listed there, it's been implemented. If it's listed as a change made *since* the latest version, it should be available in the development snapshots, in which case testing will be very welcome.
- try the Wishlist page, and see if you can find the feature there. If it's on there, and not in the 'Recently fixed' section, it probably *hasn't* been implemented.

Used ExtraPuTTY DLL wihtout windows installer

The ExtraPuTTY DLL need the path of PuTTY.
In order to do that you must create the environment variable "ExtraPuTTY" with the path of ExtraPuTTY.

how to send unprintable characters

From XML commands file used syntaxe "EXT_SPECHAR_" + characters.
The unprintable charaters managed are NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SUB, ESC, FS, GS, RS, US, Sp, DEL. (value 0x00 to 0x20 and 0x7F)
For example to sending BEL + HELLO + BEL, used the command EXT_SPECHAR_BEL_HELLO_BEL or for sending CR LF without data used the command EXT_SPECHAR_CR_LF (in this case don't forget to unset the implicit CRLF)

XML syntax

The five "predefined entities" representing special characters are managed (",,;,').

TestStand using

The ExtraPuTTY TestStand steps are automaticly install with the ExtraPuTTY windows installer.

Note : Closed TestStand during the installation else re-start TestStand application at the end

BUGS : TestStand Steps Upgrade

On ExtraPuTTY upgrade if you change the ExtraPuTTY install directory, the path of the DDL in your teststand sequence will not be update.

This major bug is not correct with TestStand V3.0,for bypassing the problem, edit your sequence file and change the old DDL path of the keyword "LibPath" by the new install directory.

ExtraPuTTY DLL with .NET

The Extraputty dll is not an assembly or COM object. To accessing and wrapping unmanaged API from managed code you shall used Interoperability ("interop"). See VB.NET example.

ExtraPuTTY DLL compile options

The ExtraPuTTY DLL is compile with calling convention "__stdcall" and with struct member alignment "16 bytes"

ExtraPuTTY Command Line

The ExtraPuTTY snapshot version alow to used command line options to run a scenario automatically on startup of putty session.

Syntax : [Path_of_putty] [putty_command_line] [-startup] [scenario_name]

Example start a putty session name "test1" and start scenario "autolog" :
c:\PUTTY.exe -load test1 -startup autolog

Note : You need double quotes around the session name if it contains spaces.

If you want to provide feedback on this manual or on the ExtraPuTTY tools themselves, see the Feedback page.

ExtraPuTTY predefined variable

Variable	Definition
<PATH_EXPUTTY>	Shortcut for relative path of putty.exe which can be used in File,Help and childroot (Type 3 ,lua cript) TAGS of XML commands file (see Chapter 5: ExtraPuTTY Files) or in ExtraPuTTY settings panel (see Chapter 4.2: The ExtraPuTTY panel).

ExtraPuTTY predefined commands

From ExtraPuTTY XML command file some predefined commands are available :

Virtual Key codes :

Commands	ExtraPutty Virtual Key codes
ADD	<VK_ADD>
APPS	<VK_APPS>
BACK	<VK_BACK>
CAPITAL	<VK_CAPITAL>
CLEAR	<VK_CLEAR>

DECIMAL	<VK_DECIMAL>
DELETE	<VK_DELETE>
DOWN	<VK_DOWN>
END	<VK_END>
RETURN	<VK_RETURN>
ESCAPE	<VK_ESCAPE>
F1	<VK_F1>
F2	<VK_F2>
F3	<VK_F3>
F4	<VK_F4>
F5	<VK_F5>
F6	<VK_F6>
F7	<VK_F7>
F8	<VK_F8>
F9	<VK_F9>

F10	<VK_F10>
F11	<VK_F11>
F12	<VK_F12>
HELP	<VK_HELP>
HOME	<VK_HOME>
INSERT	<VK_INSERT>
LEFT	<VK_LEFT>
NUMLOCK	<VK_NUMLOCK>
PRINT	<VK_PRINT>
RIGHT	<VK_RIGHT>
SCROLL	<VK_SCROLL>
SNAPSHOT	<VK_SNAPSHOT>
SUBTRACT	<VK_SUBTRACT>
TAB	<VK_TAB>
UP	<VK_UP>

CTRL	<CTRL>
ALT	<ALT>
ALTGR	<ALTGR>

Example :

- Send CTRL+C : <CTRL>C
- Send F1 TEST F11 ALT+F1 : <VK_F1>TEST<VK_F11><ALT><VK_F1>

PuTTY system commands :

System Commands	ExtraPutty predefined commands
Event log	EXT_SYS_CMD_SHOWLOG
New Session ...	EXT_SYS_CMD_NEWSESSION
Restart Session	EXT_SYS_CMD_RESTART
Duplicate Session	EXT_SYS_CMD_DUPSESSION
Saved Sessions	EXT_SYS_CMD_SAVEDSESSION
Change Settings ...	EXT_SYS_CMD_RECONF
Copy All to Clipboard	EXT_SYS_CMD_COPYALL
Clear Scrollback	EXT_SYS_CMD_CLEAR_SB
Reset Terminal	EXT_SYS_CMD_RESET_TERM

Full Screen	EXT_SYS_CMD_FULLSCREEN
Help	EXT_SYS_CMD_HELP
Close PuTTY	EXT_SYS_CMD_CLOSE_PY
Minimize PuTTY	EXT_SYS_CMD_MINIMIZE
Maximize PuTTY	EXT_SYS_CMD_MAXIMIZE
About PuTTY	EXT_SYS_CMD_ABOUT

ExtraPuTTY system commands :

System Commands	ExtraPutty predefined commands
Start Trace	EXT_SYS_CMD_START_TRACE
Stop Trace	EXT_SYS_CMD_STOP_TRACE
Edit Command File	EXT_SYS_CMD_EDIT_XML_FILE
About ExtraPuTTY	EXT_SYS_CMD_ABOUT_EXT
Close ExtraPuTTY window	EXT_SYS_CMD_CLOSE_WND

Extra commands :

Commands	ExtraPutty predefined commands

Display message box	EXT_MSGBOX_ + Text (EXT_MSGBOX_HELLO World => display HELLO World, the title of the message box it's the name of command)
Display input box	EXT_MSGBOX_INPUTBOX_ + Text (EXT_MSGBOX_INPUTBOX_Enter your password => display Enter your password, the title of the message box it's the name of command)

Telnet special commands :

Commands	ExtraPutty predefined commands
Are you there	EXT_SYS_CMD_TS_AYT
Break	EXT_SYS_CMD_TS_BRK
Synch	EXT_SYS_CMD_TS_SYNCH
Erase Character	EXT_SYS_CMD_TS_EC
Erase Line	EXT_SYS_CMD_TS_EL
Go Ahead	EXT_SYS_CMD_TS_GA
No Operation	EXT_SYS_CMD_TS_NOP
Abort Process	EXT_SYS_CMD_TS_ABORT
Abort Output	EXT_SYS_CMD_TS_AO
Interrupt Process	EXT_SYS_CMD_TS_IP

Suspend Process	EXT_SYS_CMD_TS_SUSP
End Of Record	EXT_SYS_CMD_TS_EOR
End Of File	EXT_SYS_CMD_TS_EOF
End Of Line	EXT_SYS_CMD_TS_EOL

Appendix B : ExtraPuTTY License

ExtraPuTTY is based on PuTTY software which is developped by Simon Tatham team's under MIT license (PuTTY's license).

The ExtraPuTTY Licensing Model **Version 2.3**

ExtraPuTTY is a *free software* and copyright 2007-2015 *Sebastien Blavier.*

The ExtraPuTTY executables are distributed under the Earth license as described below :

Do your possible :

- *to respect children*
- *to reduce your consumption of energy*
- *to respect animals*
- *to respect environement*

Allowed Activities.

- *anybody (even companies) can use ExtraPuTTY without restriction (even for commercial purposes) and owe nothing to me or anybody else.*
- *apart from having to maintain the copyright notice and the licence text in derivative products, anybody (even companies) can use into their own programs and products (even commercial products) and owe nothing to me or anybody else.*

Prohibited Activities. You may not:

- *use the Software in any application in which death, personal injury, or severe physical or property damage is a foreseeable consequence of Software use or failure including.*
- *download or transfer the Software to: any country prohibited by french laws*

and regulations; any person or entity prohibited from receiving French exports; or any country which requires an import or use permit for encryption technology.

- *sublicense*
- *modify ExtraPuTTY guideline*
- *sell ExtraPuTTY*

The precise license text, as given in the About box and in the file license in the distribution and affect all ExtraPuTTY version since the 0.26 released.

In particular, there is no warranty and if ExtraPuTTY causes you damage you're on your own, so don't use it if you're unhappy with that.