# MyTerm 305

26. January 2006

FreeWare from GreenHouse

MyTerm retrieves and displays the IP address, or X.25 DTE number, of a GUARDIAN terminal device.

The result is:

- Displayed to the user (default)
- Written to any OUT type file
- Written to a MyTerm controlled key sequenced pair of files;
  this version is intended to be used in TACLLOCL to maintain a list of logged on terminal names and their related IP addresses.
  The terminal name in the file can be used as index into the PrCoSEEP log file to relate a process start action with an IP address.

MyTerm also can be used as a Telnet service to display the window IP address as well as GUARDIAN file name; it stays alive for a given time, before it stops running. While the time is running, the terminal name can be used as IN, OUT and/or TERM parameter for e.g. VIEWSYS. This allows a VIEWSYS in a dynamic Telnet Window WITHOUT a logged on TACL beneath it.

## Requirements

MyTerm runs in the GUARDIAN environment. To retrieve the correct IP address in a secured LAN, where the cryptographic engine is a PROXY based solution, MyTerm talks to the terminal emulator by using a GetMachineUser interface. Products from comForte as well as CAIL support this.

## Command syntax

Run time parameters can be supplied to MyTerm using two methods:

1. Start-up message
2. PARAMs

In case a start-up message as well as PARAMs are defined, parameters from the start-up message take precedence.

**Start-up message**

MyTerm takes a number of start-up parameters. These attributes are supplied at start time through the start-up message (the underlined parts of the command are mandatory).

```
  MYTERM [/OUT [<file>]/]  -H[ELP]
```

where

<file>
defines a file to which the help information is sent.
In case the file does NOT exist, it is created as an EDIT type file.
Default is the home terminal of MyTerm.

-H[ELP]
directs MyTerm to display its command syntax.

```
  MYTERM [/OUT [<file>]/]  [parameter]
```

Where **parameter** is one or more of:

SPI|ESC[nn]
IS <term>
LOGGEDON[nn]
[TERMINAL] <term>
WAITTIME <time>

<file>
defines a file to which the output is sent.
In case the file does NOT exist, it is created as an EDIT type file.
In case OUT is provided, no banner line will be displayed to the user.
Default is the home terminal of MyTerm.

SPI
SPI = Keyword.
When present causes MyTerm to use the SPI interface only to retrieve the IP address of a given terminal. This does make sense when the terminal in question does NOT support the GetMachineuser escape sequence.
Default is: ESC50 (see next key word).

ESC[nn]
ESC = keyword.
[nn] = optional I/O wait time in milliseconds.
When present causes MyTerm to use the ESC method first to get a terminal IP address, before the SPI interface is used.
This does make sense for all terminal emulators, supporting the GetMachineUser ESC sequence, such as products from comForte, or CAIL.
Default is: ESC50.

IS <term>
IS = keyword.
<term> = name of a terminal, that has to be processed.
Retrieves the IP address(es) of a given terminal from the database.

LOGGEDON[nn]
LOGGEDON = keyword.
[nn] = optional number of ancestors to check to find a logged on one.
MyTerm is not interested in its ancestors state. This can change, when started from a spawned TACL, e.g. within a TACL Macro. To suppress MyTerm, when started from a spawned TACL, a

value of: LOGGEDON1 has to be supplied.
Default is: LOGGEDON0 (LOGGEDONzero).

[TERMINAL] <term> TERMINAL = optional keyword.
<term> = name of a terminal, that has to be processed.
This may fail when running in ESC mode on a terminal other than MyTerms home terminal.
Default is: <term> = Home terminal of MyTerm.

WAITTIME <time> WAITTIME = keyword.
<time> = wait time in milliseconds, causing MyTerm to stay alive before it stops running.
Default is: WAITTIME 0.

**MYTERM PLAIN [parameter]**

Where **parameter** is one or more of: SPI|ESC[nn]
LOGGEDON[nn]

PLAIN directs MyTerm to write the IP address along with the terminal name into a key sequenced file.
The file has to be named: TERMIPA0.
In case it does not exist, it becomes created along with a second file named: TERMIPA1.
When TERMIP0 runs full, it is renamed to TERMIPA1, and the existing TERMIPA1 is renamed to TERMIPA0 and emptied by a PURGEDATA action.

The record structure of the TERMIPAn-files is:

```
Struct Terminal_IP_Address;
  Begin
    String TermName[0:25];     ! \system.$term.#subterm
    String TIPAddress[0:15];   ! 123.123.123.123
    String DatenTime[0:17];    ! ddmmmyyyy hh:mm:ss
    Fixed  TimeStamp;          ! GMT Julian Time Stamp
  End;
```

TermName and TIPAddress (= 42 Bytes) are the primary key.

SPI SPI = Keyword.
When present causes MyTerm to only use the SPI interface to retrieve the IP address of a given terminal. This does make sense when the terminal in question does NOT support the GetMachineUser escape sequence.
Default is: ESC50 (see next key word).

ESC[nn] ESC = keyword.
[nn] = optional I/O wait time in milliseconds.
When present causes MyTerm to use the ESC method first to get a terminal IP address, before the SPI interface is used. This does make sense for all comForte and CAIL 6530 type terminal emulator products.
Default is: ESC50.

LOGGEDON[nn] LOGGEDON = keyword.
[nn] = optional number of ancestors to check to find a logged on one.
MyTerm is not interested in its ancestors state. This can change, when started from a spawned TACL, e.g. within a TACL Macro. To suppress MyTerm, when started from a spawned TACL, a

```
                    value of: LOGGEDON1 has to be supplied.
                    Default is: LOGGEDON0
```

### Special version

Beside the 'normal' versions of MyTerm, a special version is delivered, which can be used in TACLLOCL to capture the terminal name and IP address WITHOUT the need to supply any parameters. This version is named: MyTermC.

MyTermC has these default settings enabled:

```
PLAIN           directs MyTerm to write the IP address along with the terminal
                name into a key sequenced file.

ESC100          ESC = keyword.
                100 = optional I/O wait time in milliseconds.

LOGGEDON1       LOGGEDON = keyword.
                1 = number of ancestors to check to find a logged on one.
                Makes MyTerm only runs when started from an already logged on
                process (process state bits <15> or <12> set). This prevents it
                from being started from a spawned TACL.
```

### General

Start-up parameters take precedence.
The order of start-up parameters can be any.
Start-up parameters are NOT case sensitive.

### PARAMs

Two of the attributes can be shipped as PARAMs:

1. The evaluation mode to be used:
   - ESC[nn] including a wait time
   - SPI

   The default wait time for ESC is: 50 milliseconds.
2. The check of the logged on state of the ancestor:
   ON[nn] may include the number of ancestors to be checked.
   The default number of ancestors to be checked is zero.

```
PARAM           MYTERM^MODE               ESC[nn]|SPI]
PARAM           MYTERM^CHECKLOGONSTATE    ON[nn]
```

### General

Start-up parameters take precedence.
The order of start-up parameters can be any.
Start-up parameters are NOT case sensitive.

## Evaluation methods

MyTerm uses two different methods to get to the IP address:

1. By default, it sends an ESC sequence to the terminal to retrieve the terminals IP address. This works OK for 6530 type emulators from comForte and CAIL.
   To ensure to get the address, MyTerm trashes the type ahead buffer of the emulator before sending the esc sequence!
   This method is used by default first. It can be bypassed, when PARAM MYTERM^MODE is set to SPI, or when the SPI keyword is present in the start-up message.
2. In case the ESC sequence fails to get the IP address, or PARAM MYTERM^MODE is set to SPI, or the SPI keyword is present in the start-up message, the ESC method is skipped and the SPI connection to the IP-Process is used directly to retrieve the address. This is meaningful when you know, that the terminal in question does NOT the GetMachineUser escape sequence.

## MyTerm and subtype 30 processes

In case the evaluated terminal is NOT a Telnet window, but a subtype 30 process (this might be the case with GreenHouse products, such as traced terminal sessions), MyTerm traces its chain of ancestors to find a valid IP address. This is meaningful to e.g. get the IP address of a MPWD session, or a traced or checked SECOM session.
**This does NOT work when running in PLAIN mode.**

## Restrictions

- Restriction (as of 24Jun1998): The X.25 DTE number of a SAFEGUARD started ICI is NOT shown. This is a known bug, and will be solved in D46.xx (but I have no chance to test this, because X.25 lines are no longer 'hip').

- Running MyTerm nowaited may cause unpredictable results: MyTerm may be unable to talk to its home terminal, because the process, creating MyTerm, owns it, and this causes MyTerm to stop running without retrieving the IP address.

## Evaluation result

The result, displayed by MyTerm, has this format:

```
MYTERM (300) - T7172G06 - (01Apr2005)   System \BEECH, running NSK G06
GreenHouse Software & Consulting 1998,2000-2002,2005
TELNET window \BEECH.$ZTNP0.#PT35UAA is IP-Address: 192.231.36.1  (spi)
```

1. The first and second line shows the product banner as well as the version of the currently running OS.
2. The third line shows the
   - telnet window name (`\BEECH.$ZTNP0.#PT35UAA`)
   - IP address `192.231.36.1`
   - the used method to get to the IP address (spi) or (esc)

## Getting help

```
$GHS1 SECOM 9> myterm -h
MYTERM displays the GUARDIAN name as well as the IP address of a
6530 type terminal.

Command syntax:

    MYTERM [/OUT [<file>]/] -H[ELP]

    MYTERM [/OUT [<file>]/] [SPI|ESC[nn]] [LOGGEDON[nn]]
                            [[TERMINAL] <term>] [WAITTIME <time>]
                            [IS] <term>]

    MYTERM                  PLAIN [SPI|ESC[nn]] [LOGGEDON[nn]]

where

 -H[ELP]       displays this screen.
 OUT           defines the OUT file to which the output should be
               written. An empty OUT suppresses any output.
 SPI|ESC[nn]   keywords;
               SPI causes MyTerm to use the SPI interface;
               ESC causes MyTerm to use an ESC sequence to get the
               IP address directly from the terminal, BEFORE it uses
               the SPI interface. This is meaningful for WIN6530 type
               emulators from comForte and CAIL.
               The [nn]-extension is a numeric value and optionally
               defines the I/O wait time to be used when sending the
               ESC sequence. The value is in milliseconds.
               SPI or ESC[nn] can be supplied as well as a PARAM:
               PARAM MYTERM^MODE SPI|ESC[nn]
               Default is: ESC with an I/O wait time of 50.
               The start-up value takes precedence.
 LOGGEDON[nn] keyword;
               directs MyTerm to check the logged on state of its
               creator. This is meaningful when MyTerm is started
               through TACLLOCL, where the ancestor TACL is a spawned
               one, e.g. in an OBEY file, or TACL Macro.
               The [nn]-extension is a numeric value and optionally
               defines the ancestor to be checked.
               0 = default = no ancestor check.
               1 = check ancestor/mom, 2 = check grand mom, etc.
               LOGGEDON[nn] can be supplied as well as a PARAM:
               MYTERM^CHECKLOGONSTATE  ON[nn]
               Default is: LOGGEDON0.
               The start-up value takes precedence.
 IS            keyword;
               causes MyTerm to read the record(s) of the given
               terminal from thee current data base, and to display it.
 term          is a 6530 type terminal name, e.g. $ZTN01.#PTVD5UA.
               The terminal name can be a 'fully blown' name, including
               a system name, e.g. \GINKGO.$ZTN04.#AGUZZT1
 time          defines the time in milliseconds, MYTERM stays alive
               before it stops itself. This allows it to become a
               Telnet service to display a telnet window name, and IP
               address.
 PLAIN         keyword; causes MyTerm to write the IP address into a
               key-sequenced file.
               TERMINAL and WAITTIME are NOT taken into account.
               For details please consult the documentation.
e.g.:
  MYTERM
  MYTERM ESC300 TERMINAL \FOXII.$ZTN05.#OAGHIUP
  MYTERM WAITTIME 50000
  MYTERM PLAIN SPI
  MYTERM IS $ZTN1300.#PTONM2AX
$GHS1 SECOM 10>
```

## Retrieving the IP address of a terminal using both evaluation methods

When MyTerm is started without any additional parameter, it automatically first uses the ESC sequence, before it tries the SPI interface. This is the most common and practical method.

```
$GHS1 MYTERM 61> myterm
MYTERM (300) - T7172G06 - (01Apr2005)   System \BEECH, running NSK G06
Copyright (c) GreenHouse Software & Consulting 1998,2000-2002,2005
TELNET window \BEECH.$ZTN01.#PTXAUVA is IP-Address: 192.231.36.1  (esc)
$GHS1 MYTERM 62>
```

## Retrieving the IP address of a terminal with the SPI interface only

In case you know, that the terminal emulator being used does NOT support the GetMachineUser escape sequence, the SPI start-up parameter does make sense: It directs MyTerm, to use the SPI interface right away without first trying to send an ESC sequence to the terminal.

```
$GHS1 MYTERM 19> object spi
MYTERM (300) - T7172G06 - (01Apr2005)   System \BEECH, running NSK G06
Copyright (c) GreenHouse Software & Consulting 1998,2000-2002,2005
TELNET window \BEECH.$ZTNP0.#PT35UAA is IP-Address: 192.231.36.1  (spi)
$GHS1 MYTERM 20>
```

## Translating a terminal name to its IP address

MyTerm allows a quick translation of a terminal name to the IP address(es). To make this work successful, MyTerm has to be executed whenever a terminal window is opened.

```
$GHS1 MYTERM 94> myterm is \BEECH.$ZTN00.#PTNM2AA
      Terminal  Name           IP Address       Creation  Time
--------------------------  ----------------  ------------------
$ZTN00.#PTNM2AA             192.231.36.1      10Oct2005 08:30:25
$GHS1 MYTERM 95>
```

## Special Version

A special version named MyTermC is available, which has these attributes hard coded:

1. PLAIN mode is set to ON
2. The evaluation mode is set to: ESC100
3. Ancestor check is set to: LOGGEDON1
   It only runs when started from an already logged on process
   (process state bits <15> or <12> set)

This version can be used in TACL Macros WITHOUT the need to supply any start-up parameter, where MyTerm logs the IP address to a file.

The same could be achieved by running MyTerm with these parameters:

```
[run] MYTERM PLAIN WAIT 100 LOGGEDON1
```

All pre-set attributes can be overwritten by PARAMs as well as start-up parameters, but the special versions **ALWAYS** run in PLAIN mode!

## Delivery

The interactive version of MyTerm comes in four flavours:

`MYTERM.100` accelerated TAL object
`MYTERMN.700` native pTAL object

`MYTERML.100`[1] accelerated object, that allows the display of the IP address INDEPENDENT of ACLs on $ZNET.#ZSPI.
This object has to be licensed.
`MYTERMNL.700` native object, that allows the display of the IP address INDEPENDENT of ACLs on $ZNET.#ZSPI
This object has to be licensed.

In addition, these four special versions are available with these hard coded defaults:

- The evaluation mode is set to ESC
- The WAIT time is set to 100 milliseconds
- The access mode is set to PLAIN
- In case the starting instance is NOT logged on, which is the case when started from a spawned TACL, MyTerm refuses from running.

This makes it easier to use MyTerm in TACL Macros to capture TACLs IP addresses and to record them, e.g. to have a reference for terminal names and the related IP addresses.

`MYTERMC.100` accelerated object with hard coded defaults
`MYTERMC.700` native object with hard coded defaults

`MYTERMCL.100` accelerated object with hard coded defaults, that allows the display of the IP address INDEPENDENT of ACLs on $ZNET.#ZSPI.
This object has to be licensed.
`MYTERMCL.700` native object with hard coded defaults, that allows the display of the IP address INDEPENDENT of ACLs on $ZNET.#ZSPI
This object has to be licensed.

## Listing TERMIPA files

When MyTerm runs in PLAIN mode, the evaluation result is written into a file named TERMIPA0. The structure of this file is defined in the DDL file MTDDL. To list its contents, please perform these steps:

1. Compile MTDDL with the DDL compiler:
   ```
   DDL/IN MTDDL/NOLIST,DICT !
   ```
2. Run ENFORM by using the supplied query file LISTADDR:
   ```
   ENFORM/IN LISTADDR/
   ```

## MyTerm in TACLLOCL

To maintain a file, in which all terminal names are related to an IP address, the special version MyTermC can be executed from TACLLOCL.

[1] ACL's on system processes are not really meaningful and may reduce the availability of the NonStop system!

## FAQs

Q: MyTerm is defined in TACLLOCL to ensure, that all terminal names get their related IP address recorded. But when logging to an already logged on TACL, the special version of MyTerm does not update the timestamp of the terminal name/IP address entry in TERMIPA0.

A: A re-logon does not cause the execution of TACLLOCL, and does NOT change a terminal name and its IP address.

## Feel free

In case you like this tool, feel free to use it.
In case you feel insecure – do NOT use it.
In case you use it and stumble into a glitch or error: Please let me know and I'll fix it!

Carl Weber
GreenHouse Software & Consulting
26. January 2006