NatStar

Version 5.00 Edition 1

NS-DK

Version 5.00 Edition 1

NS-TruncTrace User Manual

Ref. n° NSX500TRRACE00001

Contents

About this Manual

This is the NS-TruncTrace manual for Nat System‘s development tools. NS-TruncTrace offers a set of features that make it possible to investigate problems linked to string truncations when there is an exception of a program generated with NS-DK or NatStar.

Organization of the manual

This manual contains one chapter.

Chapter 1 **NS-TruncTrace operation**

This chapter introduces NS-TruncTrace and the way it is used.

Conventions

Typographic conventions

Important term	Important terms are printed in **bold**.
Interface component	The names of windows, dialog boxes, controls, buttons, menus and options are printed in *italics*.
[F9]	Function key names appear in square brackets.
FILENAME	Filenames are printed in UPPERCASE.
`syntax example`	Syntax examples are printed in a `fixed-width font.`

Notational conventions

•	A round bullet is used for lists
♦	A diamond is used for alternatives
1.	Numbers are used to mark the steps in a procedure to be carried out in sequence

Operating conventions

Choose *XXX* \ *YYY*	This means you need to open the *XXX* menu, then choose the *YYY* command (option) from this menu. You can perform this action using the mouse or mnemonic characters on the keyboard.
Click the *XXX* \ *YYY* button	This means you need to display the tool bar named *XXX*, then click the *YYY* button in this tool bar (the name of each button is shown by its help bubble). You can only perform this action with the mouse.
Choose the *XXX* button	This means you need to choose the *XXX* button in a dialog box. You can perform this action using the mouse or mnemonic characters on the keyboard.

Icon codes

	**Comment,** note, etc.
	**Reference** to another part of the documentation
	**Danger**: precaution to be taken, irreversible action, etc.
	**Suggestion**: helpful hints, etc.
	**To go a step further**: level of detail or expertise greater than the average level of the document
	Indicates specific information on using the software under DOS- Windows (all versions)

Indicates specific information on using the software under DOS-Windows 32 bits

Indicates specific information on using the software under DOS-Windows 32 bits

Indicates specific information on using the software under Unix systems

Chapter 1 NS-TruncTrace Operation

This chapter introduces NS-TruncTrace and the way it is used.

In this chapter you will find

- An introduction to NS-TruncTrace
- The two ways of investigating problems linked to string truncations.

Table of contents

Introduction to NS-TruncTrace

NS-TruncTrace offers a set of features that make it possible to investigate problems linked to string truncations when there is an exception of a program generated with NS-DK or NatStar.

It works with two investigation modes:

- Non-recompilation mode: When a string truncation has been detected, a warning window is displayed and/or an information message is generated in the trace file specified by the NS-TRACE environment variable.

- Recompilation mode: When a string truncation has been detected, an information message specifying the location of the truncation (file/row number) is generated in the file.

Non-recompilation mode

Trace of string truncations

To trace string truncations, run the program with the NSTRUNCSTR environment variable set to the TRACE value.

The truncation messages are stored in the trace file specified by the NS-TRACE environment variable.

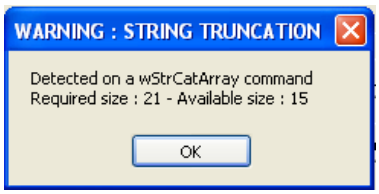
Example of a trace generated.

```
STRING TRUNCATION - Detected on a wStrCatArray command - Required size : 21 -
Available size : 15
```

Displaying a message box when there is a problem

To display a message box, run the program with the NSTRUNCSTR environment variable set to the WARNUSER value.

When detecting a problem, a message box similar to the one below is displayed:



Combining the trace and the display

To display problems in the message box and to trace the string truncations both at the same time, enter the following code:

```
SET NSTRUNCSTR=TRACE;WARNUSER
```

Recompilation mode

This mode is used to have the same information as the previous mode, while also to know the exact line of code where the problem occurred.

Trace of string truncations

To trace string truncations, run the program with the NSTRUNCSTR environment variable set to the TRACE value.

The truncation messages are stored in the trace file specified by the NS-TRACE environment variable.

Example of a trace generated.

```
STRING TRUNCATION - File 'C:\BUGS_NAT\BUGS_NATSTAR\TESTV5\WIN32\testchar.c' Line
85 ( Size 20 Maxsize 15 in wStrCatS )
```

Displaying a message box when there is a problem

To display a message box, run the program with the NSTRUNCSTR environment variable set to the WARNUSER value.

When detecting a problem, a message box similar to the one below is displayed:

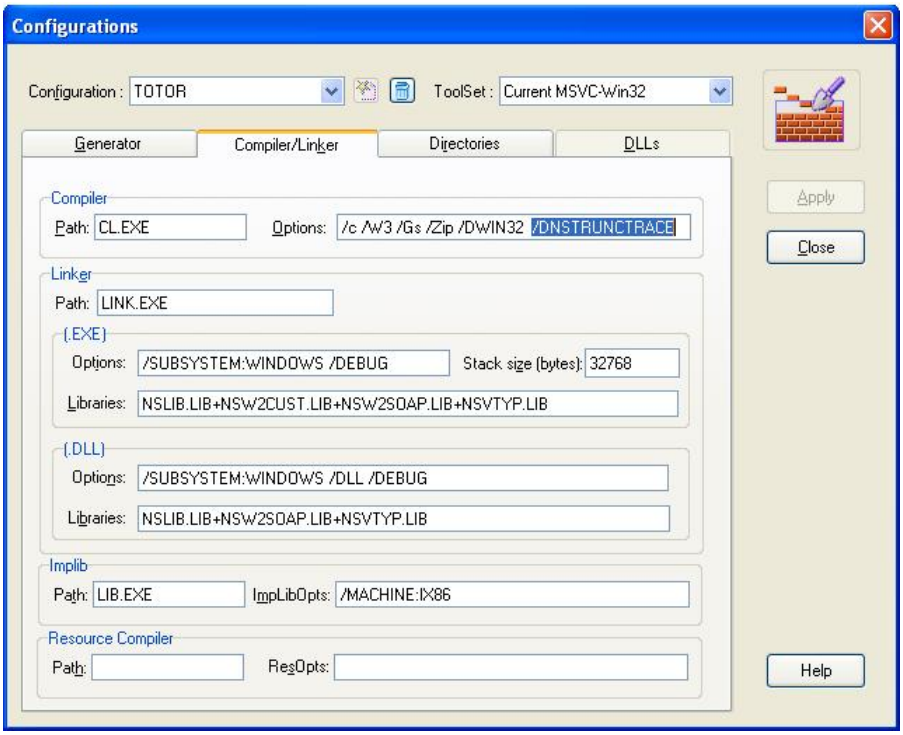
Combining the trace and the display

To display problems in the message box and to trace the string truncations both at the same time, enter the following code:

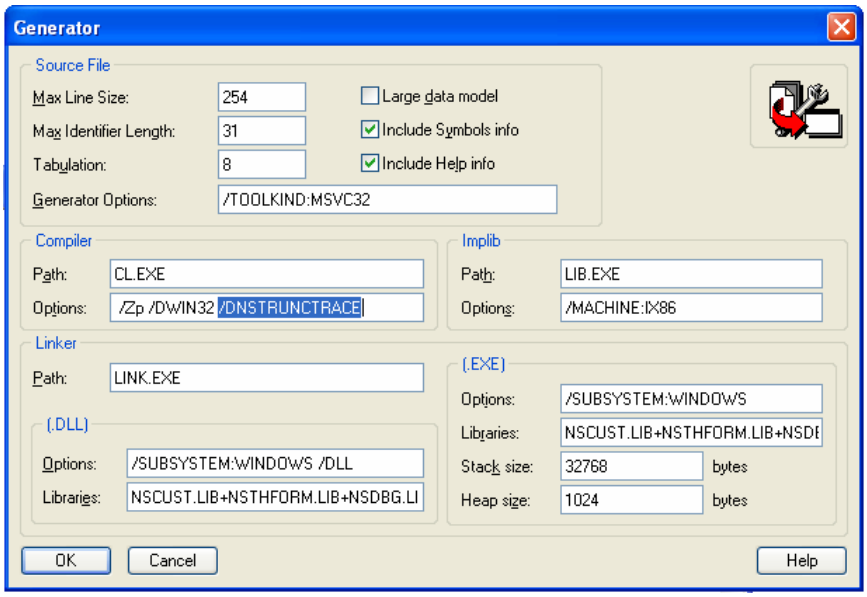
```
SET NSTRUNCSTR=TRACE;WARNUSER
```

Using compilation mode

1. When in NS-DESIGN, click the *Options/Build Configurations* menu and the *Configurations* window appears. Add the /DNSTRUNCTRACE compilation option to the *Compiler/Linker* tab.



2. In NatStar, click the *Build/Set Configuration* menu and the *Select Default Configuration* window appears. Click the *Gen…* button. The *Generator* window appears. Add the /DNSTRUNCTRACE compilation option.



3. Regenerate the program.

4. Launch the program with the NS-TRACE environmental variable, which specifies the output file.

Description of the messages

Information about the style is obtained in the file.

NSDK:19:13:30 STRING TRUNCATION - File <<*File_name*>> Line <<*Line_number*>> (Size <<*Requested_size*>> Maxsize <<*available_size*>> in <<*Internal_function_name*>>)

E.g.:

```
STRING TRUNCATION - File 'C:\BUGS_NAT\BUGS_NATSTAR\TESTV5\WIN32\testchar.c' Line
85 ( Size 20 Maxsize 15 in wStrCatS )
```

In this example, it can be seen that one string has been truncated at line 85 of the testchar.c file during a concatenation (wStrCatS), since the result needs 20 characters, but only 15 are available.

To find the corresponding NCL code easier, you should load with the “Runtime Trace” option. We will have a trace that specifies the caller NCL instruction.

E.g.:

```
NSDK:19:12:43 NSGENTRACE-funct   { TST_WLSTRCAT1$ in TOTOR.EXE/TESTCHAR.NCL(31)
NSDK:19:12:43 STRING TRUNCATION - File
'C:\BUGS_NAT\BUGS_NATSTAR\TESTV5\WIN32\testchar.c' Line 85 ( Size 20 Maxsize 15
in wStrCatS )
NSDK:19:12:43 NSGENTRACE-funct   } TST_WLSTRCAT1$ in TOTOR.EXE/TESTCHAR.NCL(34)
(CLOCK=0000.875)
```

Appendix

Steps to follow to find the origin of a C-code line generated in NCL

The trace files indicate the position of errors in the C code generated.

E.g.:

```
STRING TRUNCATION - File 'C:\BUGS_NAT\BUGS_NATSTAR\TESTV5\WIN32\testchar.c' Line
85 ( Size 20 Maxsize 15 in wStrCatS )
```

The name of the C file is the same as the resource's name. In this case, the resource in question thus has the name **TESTCHAR**.

The C code generated is:

```
/*                                                                    */
/* Function szTST_WLSTRCAT1                                           */
/*                                                                    */
NS_FN(NS_PCHAR) szTST_WLSTRCAT1(NS_PCHAR sz__RETURN)
{
    static const NS_CHAR NS_szCurId[] = "TST_WLSTRCAT1$";
    static const NS_GenTrace NS_traceEnter={1,NS_traceEnterFunc,NS_szCurMod,
      NS_szCurRsc,NS_szEmpty,NS_szCurId,NS_NULL};
    static const NS_GenTrace NS_traceLeave={1,NS_traceLeaveFunc,NS_szCurMod,
      NS_szCurRsc,NS_szEmpty,NS_szCurId,NS_NULL};
    NS_CHAR szVARIABLE[16];
    wNSGenTrace(&NS_traceEnter, 31, NS_NULL);
    wLStrCpy(szVARIABLE,"1234567890",16);
    wStrCatS(szVARIABLE,16,szVARIABLE,"ABCDEFGHIJ"); // LINE 85
    {
    wLStrCpy(sz__RETURN,szVARIABLE,21);
    wNSGenTrace(&NS_traceLeave, 34, NS_NULL);
    goto Return;
    }
Return:
    return sz__RETURN;
}}
```

From the function's name in C, it can be seen that the error is produced in the NCL function TST_WLSTRCAT1$ (the sz prefix in C corresponds to the $ suffix in NCL).

```
FUNCTION Tst_wLstrcat1$ RETURN CSTRING (20)
      LOCAL CSTRING VARIABLE$ (15)

      VARIABLE$ = "1234567890"
      VARIABLE$ = VARIABLE$ && "ABCDEFGHIJ" ; Troncation

      RETURN VARIABLE$
ENDFUNCTION                                          .
```