

TPNS - Teleprocessing
Network Simulator

SX75-0065-03

TPNS STL Reference Card

Version 3 Release 5



TPNS STL Reference Card

This reference card is a source of quick information. For a complete description of STL language elements, see *Using the TPNS Structured Translator Language (STL) and the STL Translator*.

Syntax Conventions

<i>italics</i>	Information to be supplied by user.
bold	Code exactly as shown (although upper case is not required).
[brackets]	Optional information, any information that is not bracketed is required.
{braces}	Used to designate a list of items to choose from.
vertical bar ()	Choose one of the items separated by vertical bars ().
underscore	Default option if not coded.

Statements

@EJECT

@GENERATE

TPNS_statements

@ENDGENERATE

@IFNUM=*starting_number*

@INCLUDE *member_name*

@NETWORK

TPNS_statements

@ENDNETWORK

@PROGRAM[=*program_name*]

STL_statements

@ENDPROGRAM

ALLOCATE *variable_name* {*counter*|*save_area*|*switch*}

assignment statement: *variable_name* = *expression*

BIT [{**SHARED**|**UNSHARED**}] *variable_list*

CALL *procedure_name*

CANCEL *event_tag*

CONSTANT *name* {*integer_constant_expression*|
string_constant_expression}

CURSOR ({*integer_expression*|*direction*}
[, *integer_expression*])

DEACT {*onin_onout_labels*|*event_names*|
ALL IO ONS|**ALL EVENT ONS**}

```

DELAY(({fixed_time|'RATE',rate_table_number|
        random_function}|[,uti_name])
DO
    STL_statements
END
DO WHILE condition
    STL_statements
END
DO FOREVER
    STL_statements
END
DO control_variable = initial_value TO exit_value
    [BY increment]
    STL_statements
END
IF condition THEN[:] statement[:]
    [ELSE[:] statement]
INITSELF(resource[,mode][[,user_data][,log_byte]])
INTEGER [{SHARED|UNSHARED}] variable_list
ITERATE
LEAVE
LIGHTPEN(integer_expression1[,integer_expression2])
LOG {string_expression|DISPLAY}
MONITOR
procedure_name: MSGTXT [string_constant]
    STL_statements
ENDTXT
msgutbl_name: MSGUTBL
    utbl_entries
ENDUTBL
NOP
ON SIGNALED(event_name) THEN[:]
    asynchronous_subset_statement
{ONINIONOUT} [asynchronous_condition] THEN[:]
    asynchronous_subset_statement
OPCMND operator_command
POST event_name [AFTER time [TAG event_tag]]
Q SIGNAL event_name [AFTER time [TAG event_tag]]
QUIESCE [UNTIL {ONINIONOUT} [async_condition]]|
    SIGNALED(event_name)}]
RESET event_name [AFTER time [TAG event_tag]]
RETURN
SAY message

```

```

SELECT
    WHEN condition THEN[;] statement
    [WHEN condition THEN[;] statement]
    OTHERWISE[;] statement
END
SETRH [TYPE request_type] [CHAIN chain_position]
    [ON(rh_settings)] [OFF(rh_settings)]
SETTH [SEQNO sequence_number]
    [ON(th_settings)] [OFF(th_settings)]
SHOW
SIGNAL event_name [AFTER time [TAG event_tag]]
SNACMND(type,[arg1...arg10])
STRING [{SHARED|UNSHARED}] variable_list
STRIPE stripe_data
SUSPEND([({fixed_time['RATE',rate_table_number|
    random_function]}[,uti_value]]])
TERMSELF([resource][[,log_byte]])
TRANSMIT [USING aid_key] [LOGGING log_byte]
    [EOMS([alcseom][[,twxseom][[,twxreom]]])]
    [AND wait_statement]
TYPE data
USEREXIT(module_name[,parm_list])
UTI uti_name
VERIFY simple_condition [FOR description]
WAIT [UNTIL {{ON|IN|ONOUT} [asynchronous_condition]}]
    POSTED(event_name)|
    SIGNED(event_name)}]

```

CPI-C Simulation Statements

Code each statement in the form:

CMxxxx(parameters) [**LOGGING** *log_byte*]

The parameters are not shown here. CMxxxx is one of the following:

CMACCP	CMINIT	CMSLD
CMALLC	CMPTR	CMSMN
CMCFM	CMRCV	CMSPLN
CMCFMD	CMRTS	CMSPTR
CMDEAL	CMST	CMSRC
CMECS	CMSDT	CMSRT
CMECT	CMSD	CMSL
CMEMN	CMSD	CMSST
CMEPLN	CMSERR	CMSTPN
CMESL	CMSF	CMTRTS
CMFLUS	CMSFM5	

Asynchronous Subset Statements

For ONIN, ONOUT, or ON SIGNALLED:

- ABORT
- CALL
- EXECUTE
- Any statement allowed within an EXECUTE procedure, except RETURN.

For ONIN or ONOUT only:

- CANCEL {DELAY|SUSPEND}
- NORESP

Key Statements

ATTR <i>string_constant</i>	FLDMINUS
BTAB	FLDPLUS
CHARSET <i>character_set</i>	FM
COLOR <i>color_attribute</i>	HIGHLITE <i>highlight_attribute</i>
CTAB	HOME
CURSRSEL	INSERT
DELETE [<i>delete_count</i>]	JUMP [<i>pid_number</i>]
DUP	LCLEAR
EREOF	NL
EREOS	RESET
ERIN	SCROLL {UP DOWN}
ESC <i>escape_key</i>	SYSREQ
FLDADV	TAB
FLDBKSP	

Operators

Operator	Meaning	Data Types
*	multiplication	integer
/	division	integer
//	remainder division	integer
+	addition	integer
-	subtraction	integer
	concatenation	string
&	logical and	bit
	logical or	bit
=	equality	integer, string, bit
≠,<>,><	inequality	integer, string, bit
<	less than	integer, string
>	greater than	integer, string
<=,<,>=	less than or equal	integer, string
>=,>,<=	greater than or equal	integer, string
&=	test under mask	string

STL Translator Execution Parameters

Parameter	Function
NOTPNS	No TPNS statements in printed listing
NOSEQOUT	No sequential output data set created
NOPDSOUT	No MSGDD data set members created
PROGRAM= <i>name</i>	Assign program name to STL source
PRTLNCNT= <i>nnn</i>	Specify maximum print page length
NOSOURCE	No STL source in MSGDD or SEQOUT data sets
NOLIST	No Preprocessor output listed
SUMMARY	Include network summary report
XREF	Include network cross-reference
NOXREF	Do not include network cross-reference
NOPREP	Do not preprocess network definition
NOTRAN	Only process network definition

STL Translator Return Codes

Code	Meaning
0	Execution completed successfully
4	An invalid execution parameter was specified
8	The SYSPRINT data set failed to open
12	The SYSIN data set failed to open
16	The MSGDD, INITDD, SEQOUT, or a work data set failed to open
20	An error occurred while writing to an output data set
24	An error occurred while translating the STL program or preprocessing the TPNS network definition
28	The trace correlation member of the MSGDD data set failed to open
32	Not enough storage was available to run
36	An error occurred while storing a partitioned data set member

Aid Keys

CLEAR	HELP	ROLLUP
CLEARPTN	PA1-PA3	SEND
CMD1-CMD24	PF1-PF24	SENDLINE
CURSRSEL	PRINT	SENDMSG
ENTER	ROLLDOWN	SYSREQ

Functions

Returns	Name
String	APPCLUID()
String	ATTR3270 (<i>screen_location</i> [, <i>length</i>])
String	BADGE()
String	CARD()
Integer	CCOL()
String	CHAR (<i>number</i> [, <i>length</i>])
String	CMONTH()
String	CNTRLRID()
Integer	COFF()
Bit	CPOS (<i>row_number</i> , <i>column_number</i>)
Integer	CROW()
Integer	C2D (<i>string</i> [, <i>length</i>])
String	C2X (<i>string</i>)
String	DATE ([<i>date_format</i>][, <i>days_offset</i>] [, <i>offset_direction</i>])
String	DAY()
String	DBCSADD (<i>string</i>)
String	DBCSADJ (<i>string</i>)
String	DBCSDEL (<i>string</i>)
String	DBCS2SB (<i>string</i>)
String	DELSTR (<i>string</i> , <i>n</i> [, <i>length</i>])
String	DEVID()
String	DIDO()
Integer	E2D (<i>numeric_string_expression</i> [, <i>length</i>])
String	FM()
String	FPORTID()
String	HEX (<i>number</i> [, <i>length</i>])
String	ID ([<i>length</i>])
Integer	INDEX (<i>source</i> , <i>target</i>)
String	INSERT (<i>new</i> , <i>target</i> [, <i>n</i>][, <i>length</i>][, <i>pad</i>])
String	LASTVERB()
String	LCHID()
String	LEFT (<i>string</i> , <i>length</i> [, <i>pad</i>])
Integer	LENGTH (<i>{string integer_expression}</i>)
String	LINID()
String	LINNAME()
String	LLSID()
String	LPORTID()
String	LUID()
String	MAG10()
String	MAG10S()
String	MAG63()
String	MAG63S()
String	MCHID()
String	MONTH()
String	MSGTXTID()
String	NETID()
String	NL()

Returns	Name
Integer	NUMCOLS()
Integer	NUMROWS()
Bit	POSTED(<i>event_name</i>)
String	PUID()
String	PU21ID()
Integer	RANDOM({ <i>low</i> , <i>high</i> 'RN', <i>rn_number</i> })
String	REPEAT(<i>character</i> , <i>count</i>)
String	RIGHT(<i>string</i> , <i>length</i> [, <i>pad</i>])
String	RNUM({ <i>low</i> , <i>high</i> 'RN', <i>rn_number</i> }[, <i>length</i>])
Integer	ROWCOL(<i>row</i> , <i>column</i>)
String	SB2DBCS(<i>string</i>)
String	SB2MDBCS(<i>string</i>)
String	SESSNO()
String	SSCPID()
String	SUBSTR(<i>source</i> , <i>starting_position</i> [, <i>length</i>])
String	TAB()
String	TCPIPID()
String	TERMID()
String	TOD([<i>length</i>])
String	TPID()
String	TPINSTNO()
String	TRANSLATE(<i>string</i> [,([<i>tableo</i>][[,([<i>tablei</i>][[, <i>pad</i>]]))])
String	UTBL(<i>utbl_name</i> ,({ <i>index_number</i> 'R'!'Rn'})
Integer	UTBLMAX(<i>user_table</i>)
Bit, Integer	UTBLSCAN(<i>source</i> , <i>utbl_name</i> [, <i>integer_variable</i>])
String	VTAMAPID()
String	YEAR()

Reserved Words

@EJECT	CMSEND	EREOF	MSGUTBL	SETRH
@ENDGEN[ERATE]	CMSERR	EREOS	NETID	SETTH
@ENDNET[WORK]	CMSF	ERIN	NL	SHARED
@ENDPROGRAM	CMSFM5	ESC	NONE	SHOW
@GEN[ERATE]	CMSLD	EVENT	NOP	SHUTD
@IFNUM	CMSMN	EXC	NORESP	SIGNAL
@INCLUDE	CMSPLN	EXECUTE	NTWKPRTY	SIGNALED
@NET[WORK]	CMSPTR	E2D	NUMCOLS	SNACMND
@PROGRAM	CMSRC	FLDADV	NUMROWS	SNI
ABORT	CMSRT	FLDBKSP	OFF	SSCPID
AFTER	CMSL	FLDMINUS	ON	STRING
ALL	CMSTT	FLDPLUS	ONIN	STRIPE
ALLOCATE	CMSTPN	FM	ONOUT	SUBSTR
AND	CMTRTS	FMI	ONS	SUSPEND
APPCLUID	CNTRLRID	FOR	OPCMND	SYSREQ
ATTR	COFF	FOREVER	OTHERWISE	TAB
ATTR3270	COLOR	FPORITD	PACE	TAG
BADGE	CONSTANT	HELP	PA1-3	TCPIPID
BB	CPOS	HEX	PF1-24	TERMIN
BID	CR	HIGHLITE	POST	TERMSELF
BIS	CROW	HOME	POSTED	TERMSLF1
BIT	CTAB	ID	PRINT	TGNFIFO
BTAB	CURSOR	IF	PUID	TGSWEEP
BUFFER	CURSSEL	INDEX	PU21ID	TH
BY	C2D	INITOTH1	QRI	THEN
CALL	C2X	INITOTH2	QSIGNAL	TO
CANCEL	DATA	INITSELF	QUEC	TOD
CARD	DATE	INITSLF1	QUIESCE	TPID
CCOL	DAY	INITSLF2	RANDOM	TPINSTNO
CDI	DBCSADD	INSERT	RELQ	TRANSLATE
CEB	DBCSADJ	INTEGER	REPEAT	TRANSMIT
CHAIN	DBCSDEL	IO	RESET	TYPE
CHAR	DBCS2SB	ITERATE	RESP	UNBIND
CHARSET	DEACT	JUMP	RETURN	UNSHARED
CHASE	DELAY	LASTVERB	RH	UNTIL
CLEAR	DELETE	LCHID	RIDISC	UP
CLEARPTN	DELSTR	LCLEAR	RIGHT	USEREXIT
CMACCP	DEVID	LEAVE	RNDISC	USING
CMALLC	DIDO	LEFT	RNUM	UTBL
CMCFM	DISPLAY	LENGTH	ROLLODOWN	UTBLMAX
CMCFMD	DO	LIGHTPEN	ROLLUP	UTBLSCAN
CMDEAL	DOWN	LINID	ROWCOL	UTI
CMD1-24	DR1	LINNAME	RSHUTD	VERIFY
CMECS	DR2	LLSID	RTR	VRCWI
CMECT	DUP	LOG	RU	VRCWRI
CMEMN	EB	LOGGING	SAY	VRPACCN
CMEPLN	EDI	LPORTID	SBI	VRPRQ
CMESL	EFI	LUID	SB2DBCS	VRPRS
CMFLUS	ELSE	LUSTAT	SB2MDBCS	VRRWI
CMINIT	END	MAG10	SCREEN	VTAMAPID
CMND	ENDTXT	MAG10S	SCROLL	WAIT
CMONTH	ENDUTBL	MAG63	SDT	WHEN
CMPTR	ENTER	MAG63S	SELECT	WHILE
CMRCV	EOMC	MCHID	SEND	WRU
CMRTS	EOMI	MONITOR	SENDLINE	XOFF
CMSC	EOMPB	MONTH	SENDMSG	XON
CMSDT	EOMS	MSGTXT	SEQNO	YEAR
CMSED	EOT	MSGTXID	SESSNO	

Invalid Asynchronous Conditions

The following expressions cannot be used for ONIN or ONOUT asynchronous conditions:

1. SUBSTR with string constant as first argument
(ex: onin substr('hello',1,1) = 'h' then...)
2. SUBSTR with nonconstant integer as second argument
(ex: onin substr(DATA,a,1) = 'h' then...
onin substr(SCREEN,a+1,1) = 'h' then...
onin substr(RU,index(RU,c),1) = 'x' then...)
EXCEPTION: COFF() and LENGTH(SCREEN) can be used as the second argument
(ex: onin substr(SCREEN,coff()+3,2) = 'Hi' then...
onin substr(SCREEN,length(SCREEN)) = c then...)
3. Integer comparisons involving the following functions: CCOL, COFF (except as in #2 above), CROW, C2D, E2D, NUMCOLS, NUMROWS, RANDOM, and RNUM
4. Arithmetic operations involving variables or functions
(ex: onin count + 5 = 20 then ...)
5. POSTED or SIGNED with an event name that is a concatenation of at least one string variable or a string expression involving a hexadecimal string constant or function
(ex: on signaled('hello'||mydata) then ...
onin posted('ffff'x) then ...)
6. UTBLMAX referencing a table before the table definition (MSGUTBL) has been encountered by STL
7. USEREXIT statements
8. LENGTH with an argument that is a variable string expression or a hexadecimal string constant or an integer variable
(ex: onin a = length(mydata) then ...
onin a = length('ffff'x) then ...
onin a = length(count) then ...)
9. Any string comparison involving the C2X, LEFT, RIGHT, INSERT, DELSTR, DBCSADD, DBCSADJ, DBCSDEL, DBCS2SB, SB2DBCS, SB2MBCS, and TRANSLATE functions
(ex: onin c = c2x('dfdf') then ...)
10. ATTR3270 with integer expression that contains one or more variables
(ex: onin atts = attr3270(offset) then ...
onin atts = attr3270(coff()-(offset+4)) then ...
onin atts = attr3270(rowcol(row+8,12)) then ...)



File Number: S370/4300/30XX-50
Program Number: 5688-121

Printed in U.S.A.

SX75-0065-03

