

# Natural under UTM - Part 2

This part of the Natural UTM Interface documentation deals with the macro parameters.

It covers the following topics:

- NATUTM Macro Parameters
- NATUTM Macro Entries
- NURENT Macro Parameters

## NATUTM Macro Parameters

The following parameters are available:

ADACALL | ADACOM | ADAPRI | ADAUTM | APPLNAM | APRISTD | ASAPPLI | ASYNTAC | BADTAC | CDYNAM | CLRKEY | CURPRO | ICONTRL | INITPRG | KB | KBSAVE | KBUSEXT | LFH | LINK | LINK2/LINK3/LINK4 | LOFFMAP | NATMON | NUAADDR | NUCNAME | PARMOD | PENDPR | PFK | PRKEY | REFRKEY | ROLLACC | ROLLTSZ | RSTCNT | RSTWARM | SCRNOPT | SHUTALL | SHUTLST | SPOOL | STRTALL | STRTFST | SVDYPRM | SWAMODE | SWDPAGE | SWPUSID | SYAPPLI | SYNTAC | SYSLST | TACEND | TCLA1 | TCLA2, TCLA3, TCLA4 | TCLS1 | TCLS2, TCLS3, TCLS4 | TERMTAB | TID | TRACE | TTYLS | TTYP5 | UMODE

### ADACALL - Access to Adabas

This parameter defines an entry in the Natural UTM Interface for the subroutine ADACALL. This subroutine must be called each time a non-Natural program accesses Adabas. ADACALL generates a valid Adabas user ID and subsequently invokes the Adabas interface module ADALNN. Possible values are:

Value:	Explanation:
<i>name</i>	<i>name</i> of the entry.
NO	Subroutine ADACALL is not generated. This is the default value.

### ADACOM - Adabas Link Module Usage

This parameter determines which Adabas link module is to be used. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
ADABAS	The modules ADAUSER and SSFB2C are linked to the front-end part (Adabas Version 7.1 and higher).
ADALINK	The modules ADALNK, ADAL2P and SSFB2C are linked to the front-end part (Adabas Version 7.1 and higher).
, (comma)	The module ADALNN is linked to the front-end part (Adabas versions lower than 7.1). This is the default value.



## ADAPRI - Activation of Adabas Priority Control for UTM Application

See *Adabas Priority Control* for details. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	Activates Adabas priority control for a Natural UTM application.
NO	The Adabas priority for all UTM transactions is the same. This is the default value.

## ADAUTM - Synchronization of Async UTM/Adabas Transactions

This parameter enables you to realize synchronized processing and coordinated restart of asynchronous transactions between UTM and Adabas. This requires that the module ADAUTM is available; this module must be linked to the front-end part of the Natural UTM application. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	Synchronized processing and coordinated restart of asynchronous transactions between UTM and Adabas are enabled.   <b>Warning:</b> <b>If ADAUTM=YES is specified, the ADACALL parameter must not be set to ADABAS.</b>
NO	This is the default value.   <b>Warning:</b> <b>Do not change the default value of this parameter without prior consultation of Software AG support.</b>

## APPLNAM - Name of Natural UTM Application

With this parameter, you specify the name of the Natural UTM application. The value of this parameter must be identical with the value of parameter APPLINAME in KDCDEF. This name is used to create a name for a task-specific SYSLST file (see also SYSLST parameter below). Possible values are:

Value:	Explanation:
<i>name</i>	Up to 8 characters long. No default value is provided.

The specified name is also used to construct a serialization marker for the initialization routine in the Natural UTM Interface; an S is inserted in the first free character position (for example, if APPLNAM=NATUTM, the name of the serialization marker is NATUTMS).

Furthermore, this name is used to create an Adabas user ID if TID=N is specified.

A defined character position of the operand of APPLNAM can be used for constructing the Adabas user ID; see parameter TID.

## APRISTD - Adabas Priority for Standard UTM TAC


This parameter can be used to define the Adabas priority *nnn* for the standard UTM TAC (default NAT). Possible values are:

Value:	Explanation:
<i>nnn</i>	Adabas priority <i>nnn</i> for the standard UTM TAC (default NAT).
144	This is the default value.

The APRISTD parameter is only in effect if the ADAPRI parameter is set to YES. For individual TACs, individual priorities can be defined with the parameters TCLSn and TCLAn; see also *Adabas Priority Control*.

## ASAPPLI - Name of Logical UTM Communications Partner

This parameter specifies the name of the logical UTM communications partner (as defined in KDCDEF) of the asynchronous UTM application. This name is only relevant in the case of asynchronous transaction processing between two UTM applications. Possible values are:

Value:	Explanation:
<i>name</i>	<i>name</i> specifies the name of the logical UTM communications partner.   <b>Warning:</b> <b>If ASAPPLI=<i>name</i> is specified, the operand of the parameter SYAPPLI must also be defined.</b>
NO	This is the default value.

## ASyntac - UTM Transaction Code for Asynchronous UTM Task or Application

With this parameter you define the UTM transaction code (TAC) for the UTM task or application that runs asynchronously. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
<i>tac</i>	UTM TAC for the UTM task or application that runs asynchronously.
NATAS	This is the default value.

The specified UTM TAC must be distinct from the "standard" Natural TAC and also from the TAC used for the synchronous UTM application (if asynchronous transaction processing is used between two UTM applications).

The first five characters determine the unique identifier for asynchronous UTM TACs.

## **BADTAC - Activation of UTM Function BADTACS**

This parameter enables you to activate the UTM function BADTACS, which means that in the assembled program of macro NATUTM, the startup program AUTOTAC is generated for undefined UTM transaction codes. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	Activates the UTM function BADTACS.
NO	This is the default value.

### **Note:**

BADTAC=YES requires that the following additional definitions must be supplied when defining KDCDEF and generating KDCROOT:

### **KDCDEF:**

```
PROGRAM AUTOTAC , COMP=ASSEMB
TAC KDCBADTC , CALL=FIRST , PROGRAM=AUTOTAC , EXIT=NUERROR , TYPE=D
TAC AUTOCONN , TYPE=D , PROGRAM=NATUTM , EXIT=NUERROR , CALL=BOTH
```

## **CDYNAM - Maximum Number of Programs to be Dynamically Loaded**

This parameter specifies the maximum number of programs to be dynamically loaded (for example, COBOL or Assembler subroutines) and/or the number of programs which have been linked with the front-end part and declared with parameters LINK to LINK4. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
<i>nn</i>	<i>nn</i> defines the number of programs.
15	This is the default value.

### **Note:**

The programs to be dynamically loaded must be either in the load library specified in the Natural parameter module or in the BLSLIB library or libraries specified in the start job.

## CLRKEY - Activation/Deactivation of CLEAR Key

This parameter activates or deactivates the CLEAR key. Possible values are:

Value:	Explanation:
ON	Activates the CLEAR key (keys LSP and ENTER).  This is the default value.
OFF	Deactivates the CLEAR key, which means that after pressing CLEAR, the entire last Natural screen is displayed again.

## CURPRO - Cursor Positioning to Protected Field

This parameter controls whether the cursor can be positioned to a protected field. Possible values are:

Value:	Explanation:
ON	The cursor <i>cannot</i> be positioned to a protected field.  This is the default value.
OFF	The cursor can also be placed in a protected field (for example, for field-specific help functions).

## ICONTRL - UTM Input Exit for Messages in Minus Format

This parameter allows you to generate an UTM input exit for messages in minus (-) format; that is, messages from a Natural screen. Such an input exit controls the allowed (or not-allowed) user KDC commands. Possible values are:

Value:	Explanation:
( YES , KDC xxxx ( , KDC xxxx , . . . ) )  ( YES )	Any KDC command not allowed must be defined with this parameter by specifying YES and the name of the KDC command. See examples below.
( NO )	This is the default value.

### Examples:

ICONTRL= ( NO )	This example does not generate an input exit and allows all UTM commands.
ICONTRL= ( YES )	This example generates an input exit with the name ICONTRL and prohibits usage of all UTM commands.
ICONTRL= ( YES , KDCOUT , KDCOFF )	This example generates an input exit with the name ICONTRL and prohibits usage of the commands KDCOUT and KDCOFF.

If YES is specified as first operand, the generated input exit must be defined in KDCDEF and KDCROOT as follows:

```
EXIT PROGRAM=ICONTRL,USAGE=(INPUT,USERFORM)
PROGRAM ICONTRL,COMP=ASSEMB
```

## INITPRG - Value for Natural Variable \*INIT-PROGRAM

This parameter defines the value for the Natural variable \*INIT-PROGRAM. Possible values are:

Value:	Explanation:
APPLNAM	The Natural variable *INIT-PROGRAM contains the value of the parameter APPLNAM. This is the default value.
KCTACVG	The Natural variable *INIT-PROGRAM contains the value of the UTM KB field KCTACVG (UTM start TAC).

## KB - Pass KB Address as First Parameter

This parameter specifies whether the address of the UTM communication area KB (*Kommunikationsbereich*) is passed as the first parameter address each time Natural calls a non-Natural program. This has been taken account of in the subroutines and utility programs of the Natural UTM Interface. Possible values are:

Value:	Explanation:
YES	The address of the UTM communication area (KB) is passed as the first parameter address each time Natural calls a non-Natural program.
NO	This is the default value.

## KBSAVE - Saving of UTM KB via SPUT

This parameter specifies whether the UTM KB will be saved via SPUT or not. Possible values are:

Value:	Explanation:
YES	The UTM KB will be saved via SPUT, starting from the end of the KB header plus twelve bytes. This information will be saved in the LSSB before a PEND PR is executed for a user-specific partial UTM program.
NO	The UTM KB will not be saved. This is the default value.

To be able to use this parameter, you must set the following KDCDEF definition:

```
MAX LSSBS=1
```

If the user-specific partial UTM program resumes, the original communication area will be refreshed via SGET. This allows the partial UTM program to use the KB from the end of the UTM communication area header plus twelve bytes. Therefore, the program must not destroy these twelve bytes. If a KB user

extension is defined, this area will not be saved.

## KBUSEXT - Length of UTM KB User Extension

This parameter specifies the length of a UTM KB user extension. Possible values are:

Value:	Explanation:
<i>nnnnn</i>	<i>nnnnn</i> specifies the length of a UTM KB user extension. The maximum length allowed is 30720 bytes.
0	This is the default value.

Length and address of a user extension are stored in the KB:

USEREXTL DS	H	<i>length in bytes</i>
USEREXTA DS	F	<i>address</i>

For more information, see the DSECT macro CMBS2TP.

## LFH - Use of Adabas LFH

This parameter specifies that the Adabas large file handler (LFH) is to be used. Possible values are:

Value:	Explanation:
YES	Specifies that you are using the Adabas LFH.
NO	This is the default value.

If you specify YES, you also must define the buffer size for the Adabas LFH in the Natural parameter module (parameter VSIZE ).

## LINK - Programs and Modules Called from Natural

This parameter enables you to specify the names of programs and modules that are called from Natural programs and linked with the non-reentrant part. Possible values are:

Value:	Explanation:
<i>name</i> ( <i>name</i> , <i>name</i> , . . . )	The <i>names</i> of programs and modules that are called from Natural programs and linked with the non-reentrant part must be specified in the operand of this parameter.  Conversely, the programs and modules whose names are specified must be linked with the non-reentrant part, otherwise the application is put into status SYSTEMERROR and all users are rejected with an error message.

A TABLE macro call is performed for the specified programs and modules, which enters their load addresses into the dynamic loader's link table. It is therefore not necessary to dynamically load these programs when they are called by Natural programs.

Example:

```
LINK=PROG1
LINK=(PROG1,PROG2,MODUL111)
```

## LINK2/LINK3/LINK4 - Extensions of Parameter LINK

These parameters are an extension of the parameter LINK. Possible values are:

Value:	Explanation:
<i>name</i> ( <i>name</i> , <i>name</i> , . . . )	The syntax is in analogy to that of LINK. See examples below.

No default value is provided.

Since an operand definition must not be longer than 127 characters (including parentheses), the parameters LINK2 to LINK4 are provided for cases where the operand of parameter LINK would be too long.

### Examples:

```
NATUTM LINK=(PROG1,PROG2,...),
        LINK2=(PROG54,...)
NATUTM LINK=(PROG1,PROG2,PROG3,PROG4)
```

## LOFFMAP - Format Name for Logoff Message

With this parameter, a format name for the logoff message can be specified. Possible values are:



Value:	Explanation:
' -END '	<p>The message defined in the format exit module FREXIT is output:</p> <pre>NAT9994 - YOUR SESSION WAS SUCCESSFULLY FINISHED. PLEASE GIVE "KDCOFF" (LEAVE THE APPLICATION) OR "UTM-TAC".</pre> <p>The message is output in the language specified by parameter UMODE; if required, it can be modified in the program FREXIT.</p> <p>This is the default value.</p>
' '	<p>The following message is output in line mode:</p> <pre>NAT9994 - Natural TERMINATED NORMALLY</pre>
' name '	<p>The user-defined message is output.</p> <p>The message is defined with in minus (-) format in FREXIT or in asterisk (*) format with IFG and FHS.</p>
' KDCOFF '	<p>An automatic KDCOFF is performed for the user when a FIN system command or TERMINATE statement is executed.</p>

In any case, the operand specified with the LOFFMAP parameter is used as the format name for UTM. The operand is therefore restricted to a maximum of 8 characters.

## NATMON - Automatic Activation of Natural Monitor during Application Startup

This parameter specifies whether the Natural monitor is activated automatically during application startup or not. Possible values are:

Value:	Explanation:
ON	The Natural monitor is activated automatically during application startup.
OFF	<p>The Natural monitor is not activated automatically during application startup.</p> <p>This is the default value.</p>

## NUAADDR - Natural User Thread Address

With this parameter, you specify a Natural user thread address. The following happens if you specify a particular value:

Value:	Explanation:
, (comma)	Comma means no value. The Natural user thread will be allocated in the next free address below the 16-MB line.  This is the default value.
XXXXX	The Natural user thread will be allocated on the hexadecimal address in the class 6 memory below the 16-MB line. This address must be aligned to the 4-KB segment limit. The result of address plus Natural user thread's length in bytes (MAXSIZE) must not be greater than address H'DF0000'. The highest possible address is H'DFFFFFF'.
ABOVE	The Natural user thread will be allocated above the 16-MB line.
(ABOVE, NNNNN)	The Natural user thread will be allocated above the 16-MB line where NNNNN denotes the decimal number of megabytes above the 16-MB line.

### Examples:

NUAADDR=ABOVE	The Natural user thread will be allocated in the next free address above the 16-MB line.
NUAADDR=(ABOVE, 258)	The hexadecimal address of the Natural user thread is H'10200000' (above the 16-MB line).
NUAADDR=6E000I	The hexadecimal address of the Natural user thread is H'6E000' (below the 16-MB line).

When the Natural user thread is allocated above the 16-MB line, the asynchronous write buffer and the thread for asynchronous transactions will also be allocated above the 16-MB line. In this case, the 31-bit address mode will not be switched back to 24-bit address mode before a 3GL program is called. This means the called 3GL program must be able to run in 31-bit address mode.

## NUCNAME - Name of Bounded Reentrant Natural Module

This parameter specifies the name of the bounded, reentrant Natural module. Possible values are:

Value:	Explanation:
<i>name</i>	The name of the bounded, reentrant Natural module.

No default value is provided.

You must use the name of the bounded, reentrant Natural module for the Natural pool and load information in macro ADDON (macro ADDON assembles BS2STUB) and for program CMPSTART when a shared nucleus is to be used.

## PARMOD - Generation of Front-End and Reentrant Parts

This parameter applies to the generation of both the front-end and reentrant parts. Possible values are:

Value:	Explanation:		
<i>nn, loc</i>	<i>nn</i>	24 / 31	The first value of this parameter ( <i>nn</i> ) is used to define an addressing mode (24-bit or 31-bit mode) for the Natural UTM application.
	<i>loc</i>	BELOW/ABOVE	The second value of this parameter ( <i>loc</i> ) is used to define the front part location of the Natural UTM application.
( 31 , ABOVE )	This is the default value.		

If you load the front part of the application above 16 MB, this must be defined in the front part's link procedure as follows:

```
LOADPT=*XS
```

or

```
LOADPT=X'address'
```

**Example:**

```
/EXEC TSOLINK
PROG NATvrs, FILENAM=NATvrs, LOADPT=*XS, ...
TRAITS RMODE=ANY, AMODE=31
INCLUDE...
/* PARMOD=(nn, loc) MUST ALSO BE DEFINED FOR ASSEMBLING MACRO NURENT, WHICH
/* BELONGS TO THE REENTRANT PART OF Natural UTM; OPERANDS MUST BE IDENTICAL FOR
/* THE FRONT-END AND REENTRANT PARTS.
```

**PENDPR - Define UTM TAC for PEND PR**

This parameter defines a UTM TAC for a PEND PR. Possible values are:

Value:	Explanation:
' zzzzzzzz '	zzzzzzzz (maximum 8 characters) defines the UTM TAC.
' '	This is the default value (no TAC for PEND PR).

When PENDPR=' zzzzzzzz ' is specified, a PEND PR (OGRAM) is executed instead of a PEND FI (NISH) when the FIN system command is entered or a TERMINATE statement is executed or the PEND PR function key is pressed. The UTM partial program that has been associated with the specified UTM TAC is started after the PEND PR.

**PFK - Function Key Modes**

This parameter is used to set one of the following function-key modes:

Value:	Explanation:
(KN, Y)	The literals %K1 to %K20 and send-key code DÜ1 are loaded to the function keys.
(KO, Y)	The literals 01 to 20 and send-key code F5 are loaded to the function keys.
(KS, Y)	The literals A to T and send-key code F5 are loaded to the function keys; in addition, with every output message a dummy field is generated at the last two positions of the screen, which is used to receive and pass the key value.
OFF	No function key mode is generated.
KS, L	This is the default value.

Where *y* can be:

L	function keys are loaded
N	function keys are not loaded

## PRKEY - UTM Return Code for Function Key

This parameter is used to define a UTM return code for a function key (F1 to F5 or K1 to K14). Possible values are:

Value:	Explanation:
<i>nnZ</i>	Possible values are 20Z to 39Z.
35Z	Default value for K10 (keys ESC + >).

Whenever a function key defined with this parameter is activated in the Natural dialog, the Natural session is suspended and if a UTM TAC for another UTM partial program is available, a `PEND PR(OGRAM)` is executed.

This UTM TAC can be defined in several ways:

- with the Natural profile parameter `PROGRAM=tac`,
- with the parameter `PENDPR=tac`,
- with the utility program `TACSWTCH`.

On return from the called UTM partial program via the `PEND PR(OGRAM)` to the Natural UTM Interface, the Natural session is continued at the point where it has been suspended.

The same return code as specified with the `PRKEY` parameter must also be defined with an `SFUNC` statement in `KDCDEF`.

## REFRKEY - Definition of UTM Function Key

This parameter can be used to define a UTM function key. Possible values are:

REFRKEY= <i>nnZ</i>	Possible values for <i>nn</i> are in the range from 26 to 39 (K1 to K14).
REFRKEY=NO	No UTM function key defined.
REFRKEY=39Z	Default value for K14 (keys ESC + :).

If the defined function key is pressed, the last full Natural screen is refreshed. Thus it is possible to continue the dialog with Natural after the screen has been overwritten by messages from the operator or the operating system. The send key code is not passed to the Natural application. The interface sets the Natural key code to ENTER.

## ROLLACC - Access Method for Natural Roll File

This parameter defines the access method for the Natural roll file. Possible values are:

Value:	Explanation:
UPAM-SY	The access method for the Natural roll file is UPAM with synchronous roll file I/Os. This access method is not allowed with global swap pools.
UPAM-AS	The access method for the Natural roll file is UPAM with P1-eventing for asynchronous writes.  This is the default value.
(UPAM-AS , PAMWAIT)	The Natural UTM interface waits with a VPASS SVC from the completed asynchronous write before a PEND RE is executed. This option is needed because a UTM task which is inactive (P2 wait) cannot be posted via P1-eventing. Instead, the user session must be terminated with the error message Timeout for asynchronous write.
FASTPAM	The access method for the Natural roll file is FASTPAM with Forward Eventing for asynchronous writes (high performance). See prerequisites described below.

### Prerequisites for ROLLACC=FASTPAM

To use the FASTPAM option, the following prerequisites apply:

BS2000/OSD Version 1.0 or above.

- Parameter TERMTAB must be defined as SWP.
- The class II definition in the batch job for starting the resident FASTPAM environment and the FASTPAM I/O pool must be:

```
/EXEC NATUTM,CLASSII=(nnn, yy)
```

- The FASTPAM authorization in the user catalog must be:

```

/SHOW-USER-ATTRIBUTES
                FIELD:DMS-TUNING-RESOURCES=*EXCLUSIVE
/*OR ALTERNATIVELY:
/MODIFY-USER
                FIELD: DMS-TUNING-RESOURCES=EXCLUSIVE-USE>
    
```

- The BIAS for the BS2000/OSD operating system must be defined as follows:

```

/MODIFY-SYSTEM-BIAS MAX-RESIDENT-PAGES=nnn
    
```

To calculate the necessary number of resident core pages, use the following formula (ignore all rest values):

$$\text{ROLLTSZ} + 3 / 4 * 2 = N1 \text{ (FASTPAM I/O areas)}$$

$$\text{ROLLTSZ} + 31 / 32 * 36 + 4095 / 4096 * 2 = N2 \text{ (FASTPAM access lists)}$$

$N1 + N2 =$  number of resident pages for one Natural UTM task

### ROLLTSZ - Maximum Roll Thread Size

This parameter determines the maximum roll thread size *nnn* (in KB); that is, the maximum size of a compressed user thread on the Natural roll file. Possible values are:

Value:	Explanation:
<i>nnn</i>	<i>nnn</i> must be a multiple of 4 (roll file block size).
160	This is the default value.

If ROLLACC=UPAM-AS, valid values for ROLLTSZ are 4 to 1600 (KB).

If ROLLACC=UPAM-SY or ROLLACC=FASTPAM, valid values for ROLLTSZ are 4 to 3200 (KB).

To calculate the size of the Natural roll file, use the following formula:

$$\text{ROLLTSZ} / 2 * \text{maximum number of users} = nnn$$

*nnn* is the number of PAM pages for the Natural roll file.

As user threads are generally written to the roll file in compressed form, an optimum roll thread size contributes considerably to saving disc storage.

The optimum value for *nnn* can be ascertained with the Natural Swap Pool Statistics; see the SYSLST parameter.

### RSTCNT - Control of Restart Situations

This parameter can be used to control restart situations in which the "lifetime" of a user results from an old Natural UTM session. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	In such a restart situation a message is displayed to the user and the UTM task is finished with PENDING (NISH); the user must restart his/her UTM task by entering the UTM TAC.
NO	In such a restart situation the Natural session is newly initialized without a message being displayed.  This is the default value.

## RSTWARM - Control of Restart Situations

This parameter can be used to control restart situations. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	There will be a warm start of a Natural session if there is a UTM restart situation. The last terminal screen will be displayed, prerequisite for this function is a global Natural swap pool.  This is the default value.
NO	There will be a restart of a Natural session if there is a UTM restart situation.

## SCRNOPT - Terminal Types with Deactivated Natural Screen Optimization

This parameter can be used to define (one or two) terminal types for which Natural screen optimization is to be de-activated. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
(yy=zz)	yy must be a valid terminal name TERMN as defined in KDCDEF.  zz is a synonym for yy.  For terminal types defined in KDCDEF with TERMN= zz, screen optimization is then de-activated.
(yy=zz ,yy=zz)	Same as above, but two terminals defined.
NO	Screen optimization is active for all terminal types.  This is the default value.

### Example:

```
SCRNOPT=(FL=Z9)
```

where:

FL	is a valid TERMN name for IBM 3270-type terminals
Z9	is a synonym for 3270-type terminals

This example would deactivate screen optimization for those 3270-type terminals which are defined as TERMN=Z9 in KDCDEF.

## SHUTALL - Name of User Exit

With this parameter, you can specify the name of a user exit. Possible values are:

Value:	Explanation:
<i>name</i>	Specifies the name of a user exit.
SHUTEX1	This is the default value.

This user exit is invoked by the Natural UTM interface whenever a UTM task is terminated with KDCSHUT, provided that the UTM SHUTDOWN function has been defined in KDCDEF.

## SHUTLST - Name of User Exit

With this parameter, you can specify the name of a user exit. Possible values are:

Value:	Explanation:
<i>name</i>	Specifies the name of a user exit.
SHUTEX2	This is the default value.

This user exit is invoked by the Natural under UTM when the last UTM task is terminated with KDCSHUT, provided that the UTM SHUTDOWN function has been defined in KDCROOT.

## SPOOL - Automatic Start and Termination of Printer Task

This parameter enables you to specify a spooling system. Possible values are:

Value:	Explanation:
(NATSPOOL, ' <i>enter-parms</i> ', <i>n</i> )	For use with NATSPOOL (Natural Advanced Facilities), see <i>Using NATSPOOL</i> .
REPRO-2000	For use with a remote spooling system, see <i>Using REPRO-2000 Remote Spooling System</i> .
RMSPOOL	For use with your own user exit program, see <i>Using RMSPOOL User Exit</i> .

No default value is provided

The following topics are covered below:

- Using NATSPOOL
- NATSPOOL Processing Logic
- Using REPRO-2000 Remote Spooling System
- Using RMSPOOL User Exit



## Using NATSPOOL

When using NATSPOOL (Natural Advanced Facilities), the SPOOL parameter can be used to indicate that the printer task(s) required by NATSPOOL are to be started up automatically by means of ENTER calls whenever the Natural UTM application is started, and terminated whenever the application is shut down. In this case, the operands of the parameter must be:

```
SPOOL=(NATSPOOL,'enter-parm',n)
```

where:

<i>'enter-params'</i>	are the parameters for the ENTER call (in apostrophes)
<i>n</i>	is the number of printer tasks to be started (in the range 1 to 30)

### Example:

The following ENTER job is to be automatically started and terminated. The file name is AF . E . PRINT:

```
/LOGON
/OPTION MSG=FHL
/SYSFILE FILE=SYSLST
/EXEC NAFPTTSK
/LOGOFF
```

Operand definition for the parameter SPOOL:

```
SPOOL=(NATSPOOL,'AF.E.PRINT,TIME=999',2)
```

In this example, NATSPOOL is the name of the Natural spooling system; AF . E . PRINT is the file name of the ENTER job to be started and terminated; TIME=999 is an additional, optional parameter for the ENTER call (see the description of the BS2000/OSD ENTER macro); and 2 means that two NATSPOOL printer tasks are to be started/terminated.

The second suboperand can contain any valid operands (enclosed in apostrophes) for the ENTER macro call.

The operand of parameter SPOOL in macro NURENT must be NATSPOOL.

### NATSPOOL Processing Logic

The specified number of NATSPOOL printer tasks according to the operand definition in the parameter SPOOL is started when the application is started up. Interprocess communication is then used to check that at least one printer task is running. If this condition is not satisfied, the application is set to status SYSTEMERROR, an error message is output on the console and users who attempt to logon are rejected with the message:

```
NUI0036 - SYSTEMERROR ... PLEASE GIVE KDCOFF
```

For more information on this system error, see error message NUI0036.

## Using REPRO-2000 Remote Spooling System

If a remote spooling system is used (for example, TD-SPOOL or REPRO-2000), set SPOOL=REPRO-2000 in the macros NATUTM and NURENT. This function is not supported by Software AG.

The logic used by Natural offline reports must be considered when implementing the interface module for a remote spooling system (see macro NURENT, label CMWHC). When an offline report is activated, Natural transfers output a record at a time. The logic for sending and accepting print records, the layout of the print record, etc., are in macro NURENT, subroutine CMWHC.

## Using RMSPOOL User Exit

If you use your own user exit program named RMSPOOL as remote spooling interface, set SPOOL=RMSPOOL in the macros NATUTM and NURENT. See *User Exits* for details on the user exit RMSPOOL.

## STRTALL - Name of User Exit for All UTM Tasks

With this parameter, you can specify the name of a user exit. This user exit is invoked by Natural under UTM whenever a UTM task is started. Possible values are:

Value:	Explanation:
<i>name</i>	Specifies the name of a user exit.
STARTEX	This is the default value.

## STRTFST - Name of User Exit for First UTM Task

With this parameter, you can specify the name of a user exit. This user exit is invoked by Natural under UTM when the first UTM task is started, provided that the UTM STARTUP function has been defined in KDCDEF. Possible values are:

Value:	Explanation:
<i>name</i>	Specifies the name of a user exit.
STAPPLX	This is the default value.

## SVDYPRM - Save Area Length for Dynamic Natural Parameters

This parameter determines the length in bytes of a save area for dynamic Natural parameters in the UTM KB. These parameters are used when a Natural UTM session is restarted. Possible values are:

Value:	Explanation:
<i>nnnn</i>	Specifies the length in bytes of a save area for dynamic Natural parameters in the UTM KB. Possible values are 0 / 8 . . . 2048 (bytes).
0	This is the default value.

## SWAMODE - Switching from 31 to 24-Bit Address Mode

This parameter determines whether a 31-bit address mode is switched to 24-bit mode or not before a `PEND PR` is executed. What you must set depends on whether the partial UTM program can run in 31-bit address mode (NO) or not (YES). Possible values are:

Value:	Explanation:
YES	31-bit address mode is switched to 24-bit mode.
NO	This is the default value.

## SWDPAGE - Pageability of Swap Pool Main Directory

This parameter determines whether the swap pool main directory is pageable or not. Possible values are:

Value:	Explanation:
NO	Specifies that the swap pool main directory is not pageable.
YES	This is the default value.

A swap pool directory that is not pageable improves performance considerably. In that case, the BS2000/OSD macro `CSTAT` will be used to declare the swap pool directory as not pageable. To be able to specify `SWDPAGE=NO`, you must define the maximum and minimum of resident core pages in the startup job.

### Example:

```
/EXEC E.NATvrs,CLASSII=(4,2)
```

For more information, see the description of BS2000/OSD macro `CSTAT` or the description of BS2000/OSD command `EXECUTE`, operand `CLASSII` or, when `SDF` is used, the description of BS2000/OSD command `START-PROGRAM`, operand `RESIDENT-PAGES=PARAMETERS . . .`

If the call to macro `CSTAT` fails, the application is still able to run.

## SWPUSID - Swap Pool User Identification

This parameter determines the swap pool user identification. Possible values are:

Value:	Explanation:
KCLOGTER	This is the UTM KB's logical terminal name.  This is the default value.
KCBENID	This is the UTM KB's user name.
INTERNID	This is the internal terminal ID (serial number).

## SYAPPLI - Name of Logical UTM Communications Partner

With this parameter, you can specify the name of the logical UTM communications partner (as defined in KDCDEF) of the synchronous UTM application. Possible values are:

Value:	Explanation:
<i>name</i>	The operand of the parameter ASAPPLI must also be defined.
NO	This is the default value.

The operand is only significant in the case of asynchronous transaction processing between two UTM applications.

## SYNTAC - UTM TAC for Sending Messages from Async to Sync UTM Applications

This parameter defines the UTM transaction code used to send free messages for a terminal from the asynchronous to the synchronous UTM application. Possible values are:

Value:	Explanation:
<i>tac</i>	Specifies the UTM transaction code.
NATSY	This is the default value (synchronous TAC).

The UTM TAC specified in this parameter must be distinct from the "standard" Natural TAC and also from the TAC used for the asynchronous UTM application.

## SYSLST - SYSLIST File Generation for UTM Task

This parameter defines whether a SYSLST file is generated for each UTM task or not. The SYSLST file contains statistics data and error information (if a UTM task ends abnormally). Possible values are:

Value:	Explanation:
YES	A SYSLST file is generated for each UTM task.  This is the default value.
NO	No SYSLST file is generated.

The name of a SYSLST file is `LST.name.tsn`, which is generated from the following components:

LST	prefix
<i>name</i>	the value of parameter APPLNAM
<i>tsn</i>	the 4-digit task sequence number of the UTM task

## TACEND - Action at PEND

This parameter defines the action to be taken in conjunction with the UTM operation key `PEND`. Possible values are:

Value:	Explanation:
KP	Each dialog step is terminated with a <code>PEND KP (KEEP)</code> . The UTM-KB is written to the page pool of <code>KDCFILE</code> only if no additional space in UTM cache storage is available. It is to be noted that no synchronized processing between UTM(s) and Adabas can be performed.
RE	Each dialog step is terminated with a <code>PEND RE (RETURN)</code> ; that is, the end of a UTM transaction. The UTM-KBs in the page pool of <code>KDCFILE</code> are saved with each dialog step. This processing mode is required when a synchronized processing between UTM(s) and Adabas is to take place.  This is the default value.

## TCLA1 - UTM TACs for Async Transaction w. Priority Level 1

This parameter allocates UTM TACs for asynchronous transactions with priority level 1 using the UTM TACCLASS concept. A TAC table is constructed that can be accessed from Natural programs by means of the subroutine `NATTAC`, passing a priority level as parameter; see *UTM TACCLASS Concept (Priority Control)*.

Possible values are:

Value:	Explanation:
<code>tac (tac , nn)</code>	<code>nn</code> can be specified to control Adabas priority for the corresponding UTM TAC (TACCLASS); see <i>Adabas Priority Control</i> .
<code>-</code> <code>(- , 0)</code>	Specifying <code>TCLA1=-</code> (note that the dash is not enclosed in apostrophes) denotes that no UTM TAC is to be allocated.
<code>(NATAS1 , 64)</code>	This is the default value.

## TCLA2, TCLA3, TCLA4 - UTM TACs for Async Transaction w. Priority Levels 2, 3, 4

These parameters allocate UTM TACs for asynchronous transactions with priority levels 2, 3 and 4 using the UTM TACCLASS concept. Their values are used analogous to `TCLA1` (see above). Possible values are:

Value:	Explanation:
TCLA $n = tac$ TCLA $n =(tac , nn)$ TCLA $n =-$	Analogous to TCLA1, but for priority levels $n=2, 3, 4$ .
TCLA2=(NATAS2, 48) TCLA3=(NATAS3, 32) TCLA4=(NATAS4, 16)	These are the default values.

## TCLS1 - UTM TACs for Async Transaction w. Priority Level 1

This parameter allocates UTM TACs for synchronous transactions with priority level 1 using the UTM TACCLASS concept. A TAC table is constructed that can be accessed from Natural programs by means of the subroutine NATTAC, passing a priority level as parameter; see *UTM TACCLASS Concept (Priority Control)*.

Possible values are:

Value:	Explanation:
$tac (tac , nn)$ - (-, 0)	$nn$ can be specified to control Adabas priority for the corresponding UTM TAC (TACCLASS); see <i>Adabas Priority Control</i> .  Specifying TCLS1=- (note that the dash is not enclosed in apostrophes) denotes that no UTM TAC is to be allocated.
(NAT1, 128)	This is the default value.

## TCLS2, TCLS3, TCLS4 - UTM TACs for Async Transaction w. Priority Levels 2, 3, 4

These parameters allocate UTM TACs for asynchronous transactions with priority levels 2, 3 and 4 using the UTM TACCLASS concept. Their values are used analogous to TCLS1 (see above).

Possible values are:

Value:	Explanation:
TCLS $n = tac$ TCLS $n =(tac , nn)$ TCLS $n =-$	Analogous to TCLS1, but for priority levels $n=2, 3, 4$ .
TCLS2=(NAT2, 112) TCLS3=(NAT3, 96) TCLS4=(NAT4, 80)	These are the default values.

## TERMTAB - Terminal Control Table for Natural Roll File Management

This parameter defines the terminal control table needed to manage the Natural roll file. Possible values are:

Value:	Explanation:
( SWP , TERMNAME , CHECKPNT ) ( SWP , INTERNID , CHECKPNT ) ( SWP , TERMNAME ) ( SWP , INTERNID ) ( N , TERMNAME ) ( N , INTERNID )	20 bytes long 12 bytes 10 bytes 2 bytes 10 bytes 2 bytes  <i>See Explanation of Operands below.</i>
( SWP , TERMNAME )	This is the default value.

The terminal control table is allocated either in the Natural swap pool or in the Natural roll file. It contains a header (48 bytes) and an entry for each active user or active session. Its size depends on the size of the Natural roll file, on the value of the parameter ROLLTSZ and on the length of its own entries.

The Natural UTM Interface computes the length of the terminal control table as follows:

Roll file pages / (ROLLTSZ / 2) = N

$N * \text{terminal control table entry length} + 48 = \text{length of the terminal control table}$

### Explanation of Operands

Operand	Meaning
SWP	The terminal control table is allocated in the Natural swap pool.
TERMNAME	The logical terminal name will be used to identify an entry in the terminal control table.
INTERNID	The internal terminal ID (serial number) will be used to identify an entry in the terminal control table. INTERNID is two bytes long.
CHECKPNT	Is only allowed when the terminal control table is allocated in the Natural swap pool. It is necessary if terminals are defined with RESTART=NO or if a terminal pool is defined in KDCDEF. The terminal control table entry contains a checkpoint (timestamp) for the last Natural user thread that has been rolled out. A user thread in the Natural roll file should not be overwritten by a thread with a timestamp lower than the timestamp in the terminal control table entry.  CHECKPNT is 10 bytes long.
N	The number of PAM pages for the terminal control table in the Natural roll file. Possible values of this operand are 1 to 16 (PAM pages). For each terminal, 10 bytes are needed in the terminal control table. For each session, two bytes are needed.

### Examples:

TERMTAB= ( 2 , TERMNAME )


The maximal number of entries in the terminal control table:  $2 * 2048 - 48 / 10 = 404$

TERMTAB= ( 1 , INTERNID )

The maximum number of entries in the terminal control table:  $1 * 2048 - 48 / 2 = 1000$

## TID - Adabas User ID Construction Method

This parameter specifies the method to be used to construct the "unique" Adabas user ID. Possible values are:

Value:	Explanation:
$n$	<p>The Adabas user ID is constructed from the defined (<math>n</math>) character of the operand of the parameter APPLNAM (default value: N) and the last two characters of the user's first SWAPPAMKEY. <math>n</math> must be a number in the range of 1 to 8.</p> <p> <b>Warning:</b>  <b>If you specify TID= <math>n</math>, the "defined character" of the value specified with the parameter APPLNAM must be different from that of other Natural UTM applications if these use the same Adabas; otherwise, the uniqueness of the Adabas user IDs - and thus data consistency - cannot be guaranteed.</b></p>
(T, $n$ )	A unique 4-byte user ID is constructed by taking characters $n$ to (max. $n + 3$ ) of the logical UTM terminal name (KCLOGTER). $n$ must be a number in the range 1 - 8. The resulting character string must consist of valid characters (0 - 9 and A - F) and must be unique. See example below.
(U, $n$ )	The characters are taken from the UTM user ID (KCBENID), starting at the position specified by the second subparameter. The resulting character string must consist of valid characters and must be unique.
(TID=1)	<p>The Adabas user ID consists of the first digit from the operand of parameter APPLNAM and of the two-byte entry number in the terminal control table.</p> <p>This is the default value.</p>

Example:

TID=(T, 4)

	KCLOGTER	Adabas User ID
1st terminal	LTU9A110	X'00009A11'
2nd terminal	LTU9F110	X'00009F11'
3rd terminal	LTU9F120	X'00009F12'

If "mixed" Adabas calls occur within one Natural UTM application (that is, calls from both Natural and non-Natural programs), the Adabas user ID can be found using the ENTRY CMTRMID in macro NATUTM. The current Adabas user ID (4 bytes) can be found at address CMTRMID; see also the parameter ADACALL.



**Example:**

```
EXTRN CMTRMID
.
.
L R1,CMTRMID
MVC ADAID(4),0(R1)
```

**With Adabas Version 5.2:**

With Adabas Version 5.2, the Adabas user IDs are constructed either from the logical UTM terminal name (KCLOGTER) or from the UTM user ID (KCBENID); this depends on the setting of the parameter TID. For compatibility reasons, the second digit of the operand specified with TID must be a number from 1 to 8. For example:

```
KCLOGTER=LTU9A110   KCBENID=HARRIS

TID=(T,3) -> Adabas user ID: D9A110
TID=(U,1) -> Adabas user ID: HARRIS__
```

**Note:**

For "mixed" Adabas calls no particular considerations have to be observed.

In the case of asynchronous transaction processing, the Adabas user ID is constructed from the "packed" TSN of the asynchronous task.

**With Adabas Version 5.3 and above:**

With Adabas Version 5.3 and above, for the Adabas user ID, the full terminal name (KCLOGTER) will be used when  $TID=(T, n)$  or the full user ID (KCBENID) will be used when  $TID=(U, n)$ . Default is  $TID=((T, 1))$ .

**TRACE - Trace File Number and Trace Print Record Length**

With this parameter, you specify the number of a trace file and the maximal length of a trace print record. Possible values are:

Value:	Explanation:
$(nn, ll)$	$nn$ is the number for the SYSLST $nn$ trace file. Possible range: 01 -99.  $ll$ is the maximal length in characters of a trace print record. Possible range: 71 - 132.
$(99, 71)$	This is the default value.

If any external Natural trace function is active, the trace records will be written to SYSLST  $nn$ . In this case, the Natural UTM driver creates the following trace file:

**Example:**

```
applname.Natural.TRACE,SPACE=(90,60)
SYSFILE SYSLSTnn=applname.Natural.TRACE
/* applname is the application name
```

This file will be used by all tasks of the Natural UTM application. Before the Natural UTM application is terminated, the trace file will be closed as follows:

```
SYSFILE SYSLSTnn=(PRIMARY)
```

To activate the Natural trace functions, see the parameters ETRACE and ITRACE of the Natural parameter module.

## TTYLS - Physical Line Size for TTY Devices

With this parameter you can adjust Natural's physical line length to different paper formats used with a telex machine. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
<i>nn</i>	<i>nn</i> specifies the physical line size for TTY devices.
80	This is the default value.

## TTYPS - Physical Page Size for TTY Devices

With this parameter you can adjust Natural's physical page size to different paper formats used with a telex machine. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
<i>nn</i>	<i>nn</i> specifies the physical page size (number of lines) for TTY devices.
24	This is the default value.

## UMODE - Session Processing Mode

This parameter determines the Natural session processing mode. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
( <i>S</i> , <i>y</i> )	<i>S</i> specifies the mode of operation; <i>y</i> is the language indicator.  See <i>Explanation of Operands</i> below.
( <i>S</i> , <i>E</i> )	This is the default value.

## Explanation of Operands

Operand :	Meaning:
S	Mode of operation = single; that is, one Natural session can be started from one terminal/user.
Y	Language indicator for the restart message, the message for "free-running messages" and the logoff message.
	D Danish
	E English
	F French
	G German
	I Italian
	N Dutch
	S Spanish

## NATUTM Macro Entries

- CMKBADR - Current Address of UTM KB
- User Area in the Swap Pool Directory

### CMKBADR - Current Address of UTM KB

The entry CMKBADR holds the current address of the UTM communication area KB(*Kommunikationsbereich*).

The communication area can be accessed as shown in the following example, which illustrates an Assembler program that could be called from a Natural program.

#### Example:

```

EXAMPLE CSECT
      STM 14,12,12(13)
      USING EXAMPLE,15
      L 2,VCONST          LOAD ADDRESS OF KB-ADDRESS
      L 3,0(,2)          LOAD ADDRESS OF KB
      .
      .
      LM 14,12,12(13)
      BR 14
VCONST DC V(CMKBADR)    ENTRY ADDRESS
      END

```

In this case, the program name EXAMPLE must be defined with the parameter LINK or LINK2 of macro NATUTM, and the program itself must be linked with the front-end part of the Natural UTM Interface.

## User Area in the Swap Pool Directory

One fullword is available for user-defined purposes in the Natural swap pool directory - see label USERWRD in DSECT MEMPOOL of macro NAMSWDIR. This word can be used for synchronization, for example, for switching accounting on and off, whilst the Natural UTM application is running.

The following example shows how this area can be addressed.

### Example:

```

          WXTRN CMKBADR          ENTRY IN MACRO NATUTM
PROG      CSECT
          STM  14,12,12(13)  SAVE REGISTERS
          USING PROG,15      BASE OF PROGRAM
          USING KB,4         BASE OF UTM KB
          USING MAINDIR,5    BASE OF SWAP POOL DIRECTORY
          L    3,KBADR       LOAD ADDRESS OF KB ADDRESS
          L    4,0(,3)       LOAD ADDRESS UTM KB
          L    5,ASWPDIR     ADDRESS SWAP POOL DIRECTORY
          OI   USERWRD+3,1   SET THE LOW ORDER BIT OF FIELD
*
          LM  14,12,12(13)  RELOAD REGISTERS
          BR  14             RETURN
KBADR     DC   A(CMKBADR)   ENTRY IN MACRO NATUTM
          NAMSWDIR          MACRO CALL FOR SWAP POOL DSECT
MAINDIR   DSECT
          .
          .
USERWRD   DS    F           DIRECTORY USER AREA
          .
          CMKBNEX           MACRO CALL FOR UTM KB DSECT
KB        DSECT
          .
          .
ADRSWAP   DS    F           ADDRESS OF Natural SWAP POOL
          .
          END

```

When working in this area, the user must take care not to overwrite any other data in the swap pool directory. Mistakes could lead to abnormal termination of the UTM task.

## NURENT Macro Parameters

The following parameters are available:

ACCNT | ATTKEY | AUTOLINK | CALLM31 | CLR3270 | EXTAPPL | FPUT | ILCS | K2 | PARMOD |  
SCRNTRC | SPOOL | UINPEX | UOUTEX

### ACCNT - Call Logic for User Account Routine

This parameter is used to define the logic for call of the user account routine (user exit ACCEXIT). Possible values are:

<b>Value:</b>	<b>Explanation:</b>
APPL	ACCEXIT is called at change of application (new Natural logon ID). This is the default value.
DIAL	ACCEXIT is called after every dialog step.

## ATTKEY - Attention Interrupt Key

This parameter is used to define an attention interrupt key. Such a key definition only makes sense for output in non-conversational mode. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
<i>nnZ</i>	<i>nnZ</i> can be in the range of 26Z to 39Z.
ATTKEY=	Default value: no value

## AUTOLINK - Use of AUTOLINK Function

This parameter specifies whether the AUTOLINK function of the dynamic binder/loader for loading of 3GL programs is activated or not. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
YES	The AUTOLINK function is activated. This is the default value.
NO	The AUTOLINK function is deactivated.

## CALLM31 - Switching from 31 to 24-Bit Address Mode

This parameter is only relevant if Natural is generated for the 31-bit addressing mode and the front part is loaded below (PARMOD=31, see below ).

<b>Value:</b>	<b>Explanation:</b>
YES	A call from a Natural program to a 3GL program will be executed in 31-bit addressing mode.
NO	Call in 24-bit mode. The addressing mode is switched from 31-bit to 24-bit before a 3GL program will be called from a Natural program. This is the default value.

Exceptions:

- The 3GL program is loaded above the 16-MB line.
- The address of the parameter list is above the 16-MB line.

## CLR3270= xxx - Definition of CLEAR Key

This parameter defines the CLEAR key in the AID character table V (AID3270) for 3270-type devices (IBM).

Value:	Explanation:
xxx	xxx defines the CLEAR key.
PA1	By default, PA1 is the CLEAR key.

## EXTAPPL - UTM TERMN Name of External DCAM or PDN Applications

This parameter defines the UTM TERMN name (see the parameter PTERM of UTM KDCDEF) of external DCAM or PDN applications. For these TERMN names, the MGET return code 05Z (format changed) is ignored. Possible values are:

Value:	Explanation:
xx	xx and yy define the UTM TERMN name.
(xx, yy)	

No default value is provided.

## FPUT - Operation Supplement for Printing via KDCS-Call FPUT

This parameter defines the operation supplement for printing via KDCS-Call FPUT to a printer which is defined in the UTM KDCDEF.

Value:	Explanation:
NE	Total print message.
NT	Part of a print message.

## ILCS - Support of CRTE or ILCS

This parameter specifies whether the common runtime environment for calls (CRTE) or the ILCS interface for calls of 3GL programs will be supported. Possible values are:

<b>Value:</b>	<b>Explanation:</b>
CRTE	The common runtime environment for calls of 3GL programs will be supported.  Prerequisite: The program IT0SL# must be included in the front part.  INCLUDE IT0SL#,SYSLNK.CRTE.010 RESOLVE,SYSLNK.CRTE.010
YES	Only the ILCS interface for calls of 3GL programs will be supported.  Prerequisite: The program IT0INITS must be included in the front part.  INCLUDE IT0INITS,SYSLNK.ILCS RESOLVE,SYSLNK.ILCS
NO	CRTE or ILCS are not supported.  This is the default value.

## K2 - UTM Return Code for K2 Key

This parameter specifies the UTM return code for the K2 key (for Natural PA2)

<b>Value:</b>	<b>Explanation:</b>
<i>nnn</i>	<i>nnn</i> must be in the range of 26Z to 39Z.
27Z	This is the default value.

## PARMOD - Generation of Front-End and Reentrant Parts

This parameter applies to the generation of both the front-end and reentrant parts. Possible values are:

<b>Value:</b>	<b>Explanation:</b>		
<i>(nn, loc)</i>	<i>nn</i>	24/31	The first part of this parameter ( <i>nn</i> ) is used to define an addressing mode (24-bit or 31-bit mode) for the Natural UTM application.
	<i>loc</i>	BELOW/ABOVE	The second part of this parameter ( <i>loc</i> ) is used to define the front part location of the Natural UTM application.
<i>(31, ABOVE)</i>	This is the default value.		

PARMOD=*(nn, loc)* must also be defined for assembling macro NATUTM. Operands must be identical for the front-end and reentrant parts.

## SCRNTRC - Tracing of Screen I/Os

This parameter is used for debugging screen I/O to find out the reason for certain error situations. If this parameter is set to ON/ (*ON, nn*), a special debug buffer for each user will be allocated (default buffer size is 3 KB). Possible values are:

Value:	Explanation:
ON	A debug buffer for each user is allocated with a default buffer size of 3 KB.
(ON, nn)	A debug buffer for each user is allocated where nn is used to define a specific screen debug buffer size other than the default value of 3 KB.
OFF	This is the default value.

**Warning:**

**You should only set this parameter to ON/(ON, nn) after having consulted with Software AG Technical Support.**

## SPOOL - Hardcopy Destination

This parameter enables you to specify a spooling system. The value for this parameter must correspond to the value for the SPOOL parameter in macro NATUTM. Possible values are:

Value:	Explanation:
NATSPOOL	Hardcopy will be printed via NAF (Natural Advanced Facilities), see <i>Using NATSPOOL</i> .
REPRO-2000	For use with a remote spooling system, see <i>Using REPRO-2000 Remote Spooling System</i> .
RMSPOOL	For use with your own user exit program, see <i>Using RMSPOOL User Exit</i> .
No value	Hardcopy will be printed via UTM (FPUT). This is the default.

## UINPEX - Name of User Exit

With this parameter, you can specify the name of a user exit. This user exit is invoked by Natural under UTM after a terminal message has been sent; see also *User Exits*. Possible values are:

Value:	Explanation:
name	name specifies the name of a user exit
INPSCR	By default, user exit INPSCR is used.

## UOUTEX - Name of User Exit

With this parameter, you can specify the name of a user exit. This user exit is invoked by Natural under UTM before a terminal message is to be sent; see also *User Exits*.



<b>Value:</b>	<b>Explanation:</b>
<i>name</i>	<i>name</i> specifies the name of a user exit
OUTSCR	User exit OUTSCR is used. This is the default value.