# Natural under UTM - Part 3

This part of the Natural UTM Interface documentation covers the following topics:

Notation *vrs* or *vr*: If used in the following document, the notation *vrs* or *vr* stands for the relevant *version, release, system maintenance level* numbers.

---

# User Exits

Several user exits are provided in the Natural UTM Interface. These are described below.

To use any of these exits, the corresponding user program must be linked with the front-end part of the Natural UTM application. The user exit RP2PRNT is an exception.

User exit routines are called with the customary register conventions.

ACCEXIT | ACCINIT | INPTEX | RP2PRNT | RMSPOOL | SHUTALL | SHUTLST | STRTALL | STRTFST | TRMIOEX | UINPEX | UOUTEX | UVGEXIT | WHCEXT

## ACCEXIT - Macro NATUTM

The user exit ACCEXIT can be used to retrieve accounting information. Depending on the value of the parameter ACCNT in macro NURENT, this user exit is activated either at the end of each dialogue step or at each change of application (new Natural logon ID); see also *Accounting for Natural UTM Applications*.

## ACCINIT - Macro NATUTM

The user exit ACCINIT can be used to gather accounting information. It is activated at the beginning of each dialogue step; see also *Accounting for Natural UTM Applications*.

## INPTEX - Program FREXIT

The user exit INPTEX is activated whenever an input message is read. See also the description of the program INPTEX in the section *Utility Programs for Use with Natural under UTM*.

## RP2PRNT - Macro NURENT

The user exit RP2PRNT is intended as an interface to other manufacturers' spooling systems. The user exit routine (spooling program) must be reentrant and linked with the reentrant part of the Natural UTM application. See also *Other Spooling Systems* and the description of the parameter SPOOL in the section *Parameters of Macro NATUTM*.

## RMSPOOL - Macros NATUTM and NURENT

If you wish to write your own spooling interface program, call it RMSPOOL. The program RMSPOOL can be linked to the (non-reentrant) front-end part or to the reentrant part of the Natural UTM application. If it is to be linked to the reentrant part, the program itself must be written so as to be reentrant.

**Important:**
If program RMSPOOL is to be used, the SPOOL parameter in the macros NATUTM and NURENT must be set to SPOOL=RMSPOOL.

The Natural UTM Interface passes the following parameters to program RMSPOOL:

| Address (Format/Length) | Contents | |
|---|---|---|
| 1st Address (A2) | Function code. Possible function codes are: | |
| | OP | The print file has to be opened, and the first print record is passed. |
| | PR | Any subsequent print record is passed. |
| | CL | The print file has to be closed. |
| 2nd Address | Print record (data to be printed).<br><br>The first byte of the print record contains the line/form feed character. (If function code CL, this is a dummy address.) | |
| 3rd Address (B2) | Length of print record (including feed character).<br><br>(If function code CL, this is a dummy address.) | |
| 4th Address (A8) | Printer name. | |
| 5th Address | Print buffer.<br><br>This buffer can be used as work area by RMSPOOL (also if RMSPOOL is reentrant) for any purpose. The buffer is available for exclusive use by RMSPOOL between dialogue input and dialogue output. | |
| 6th Address (B2) | Length of print buffer. | |
| 7th Address (A8) | Current user ID (as in the system variable *USER). | |
| 8th Address (A8) | Current terminal ID (as in the system variable *INIT-ID). | |
| 9th Address (A8) | Current Natural library name (as in the system variable *LIBRARY-ID). | |
| 10th Address (A8) | Current Natural program name (as in the system variable *PROGRAM). | |
| 11th Address (A4/B4) | Return code.<br><br>When RMSPOOL is invoked, the Natural UTM Interface sets this field to binary 0. Upon return of control from RMSPOOL, any value other than binary 0 is interpreted as error code and (if displayable) is displayed to the user on the terminal screen and also output to SYSLST. | |

## SHUTALL - Macro NATUTM

The user exit specified with the SHUTALL parameter in macro NATUTM is activated whenever a UTM task is terminated (KDCSHUT*n*). By default, this user exit is SHUTEX1.

If the user exit specified with SHUTALL is to be used, the parameter USAGE=SHUT in KDCDEF for the Natural UTM Interface must have been set when generating KDCROOT.

## SHUTLST - Macro NATUTM

The user exit specified with the SHUTLST parameter in macro NATUTM is activated when the *last* UTM task is terminated (KDCSHUT*n*). By default, this user exit is SHUTEX2.

If the user exit specified with SHUTLST is to be used, the parameter USAGE=SHUT in KDCDEF for the Natural UTM Interface must have been set when generating KDCROOT.

## STRTALL - Macro NATUTM

The user exit specified with the STRTALL parameter in macro NATUTM is activated whenever a UTM task is started. By default, this user exit is STARTEX.

## STRTFST - Macro NATUTM

The user exit specified with the STRTFST parameter in macro NATUTM is activated when the *first* UTM task is started. By default, this user exit is STAPPLX.

## TRMIOEX - Program FREXIT

The user exit TRMIOEX is activated with each formatted input or output message.

## UINPEX - Macro NURENT

The user exit specified with the UINPEX parameter in macro NURENT is activated *after* a terminal message has been sent. By default, this user exit is INPSCR.

Natural under UTM passes the following parameters to the user exit:

| Address (Format/Length) | Contents |
|---|---|
| 1st Address | Address input buffer. |
| 2nd Address (B2) | Address message length. |

## UOUTEX - Macro NURENT

The user exit specified with the UOUTEX parameter in macro NURENT is activated *before* a terminal message is to be sent. By default, this user exit is OUTSCR.

Natural under UTM passes the following parameters to the user exit:

| Address (Format/Length) | Contents |
|---|---|
| 1st Address | Address output buffer. |
| 2nd Address (B2) | Address message length. |

## UVGEXIT - Macro NATUTM

The user exit UVGEXIT is activated at the start, restart and end (normal or abnormal) of a UTM DC transaction. The current task ID (*Vorgangskennzeichen*, KCKNZVG) is passed to the user exit routine.

## WHCEXT - Macro NURENT

The user exit WHCEXT can be used to modify an output which is to be printed before it is passed by FPUT to UTM. When WHCEXT is called, Register 9 contains the address of the output to be printed and Register 13 the address of the save area.

WHCEXT must be reentrant and it must be linked to the reentrant part of the Natural UTM application. For further information, please refer to the source listing of the assembled macro NURENT (Label 'NUWHC').

# Asynchronous Transaction Processing under UTM

To start an asynchronous transaction, the service routine NATASYN in the Natural UTM Interface has to be called.

The start of an asynchronous transaction in a Natural program is done by passing dynamic parameters according to the following pattern:

```
...
COMPRESS dynamic parameters INTO field
CALL 'NATASYN'[parameter area
SET CONTROL 'H'
WRITE NOTITLE NOHDR field
[WRITE ...]
INPUT 'text' ifield (A1)
END
```

If the length of the dynamic parameters exceeds 250 bytes (that is, if more than one WRITE statement is required), a *parameter area* has to be passed with the CALL 'NATASYN' statement.

The parameter area is also required if the asynchronous transaction is to be started with UTM DPUT; that is, at a specific time. The aggregate length of the dynamic parameters must not exceed 3750 bytes. The *parameter area* for the CALL 'NATASYN' has the following structure:

| Bytes | Contents | |
|---|---|---|
| 01-02 | Number of WRITE statements. | |
| 03 | DPUT time indicator: | |
| | R | a relative time, |
| | A | an absolute time |
| | *blank* | FPUT |
| 04-06 | Day of the year. | |
| 07-08 | Hours. | |
| 09-10 | Minutes. | |
| 11-12 | Seconds. | |

For the contents of Bytes 03 - 12, the same rules apply as described for DPUT calls in the respective Siemens UTM documentation. Natural programming examples can be found in the Natural library SYSEXTP (programs STARTAS1, ASYNMULT, STARTAS, READAUTO, AWINDOW1, AWINDOW2).

For asynchronous transaction processing, KDCROOT, KDCDEF and the UTM startup job must be modified as necessary (see the Siemens UTM documentation).

All UTM TACs for asynchronous transactions must begin with the character sequence which is defined as a unique identifier for asynchronous TACs in parameter ASYNTAC of macro NATUTM. Conversely, the first five characters of UTM TACs for synchronous transactions must *not* be this character string.

Mixed transaction processing (that is, both within a single UTM application and between two UTM applications) is not possible.

## Asynchronous Processing within a Natural UTM Application

If transactions are to be processed asynchronously within a Natural UTM application, the operands of the parameters SYAPPLI and ASAPPLI of macro NATUTM must be set to NO (this is the default value).

### Example:

This is an example of a Natural program that initializes an asynchronous transaction within a Natural UTM application.

```
* STARTAS - EXAMPLE OF THE INITIALIZATION FOR ASYNCHRONOUS
*           TRANSACTION WORKING WITHIN ONE UTM APPLICATION
*           PARMS ARE SEPARATED BY ’,’
*           SUBLIST IN STACK IS SEPARATED BY ’;’
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8)
MOVE ’PRINTER1’ TO PRDEST                      /* --> Note 1
MOVE *INIT-ID TO LTDEST                        /* --> Note 2
COMPRESS ’SENDER=’ PRDEST ’,OUTDEST=’ LTDEST ’,’
’MENU=F,STACK=(LOGON APPL1;READAUTO)’
INTO PARM1 LEAVING NO                          /* --> Note 3
CALL ’NATASYN’                                 /* --> Note 4
```

```
SET CONTROL 'H'                                  /* --> Note 5
WRITE NOTITLE NOHDR PARM1                        /* --> Note 6
INPUT 'ASYNTASK INVOKED - HOPEFULLY' IFELD(A1) /* --> Note 7
END
```

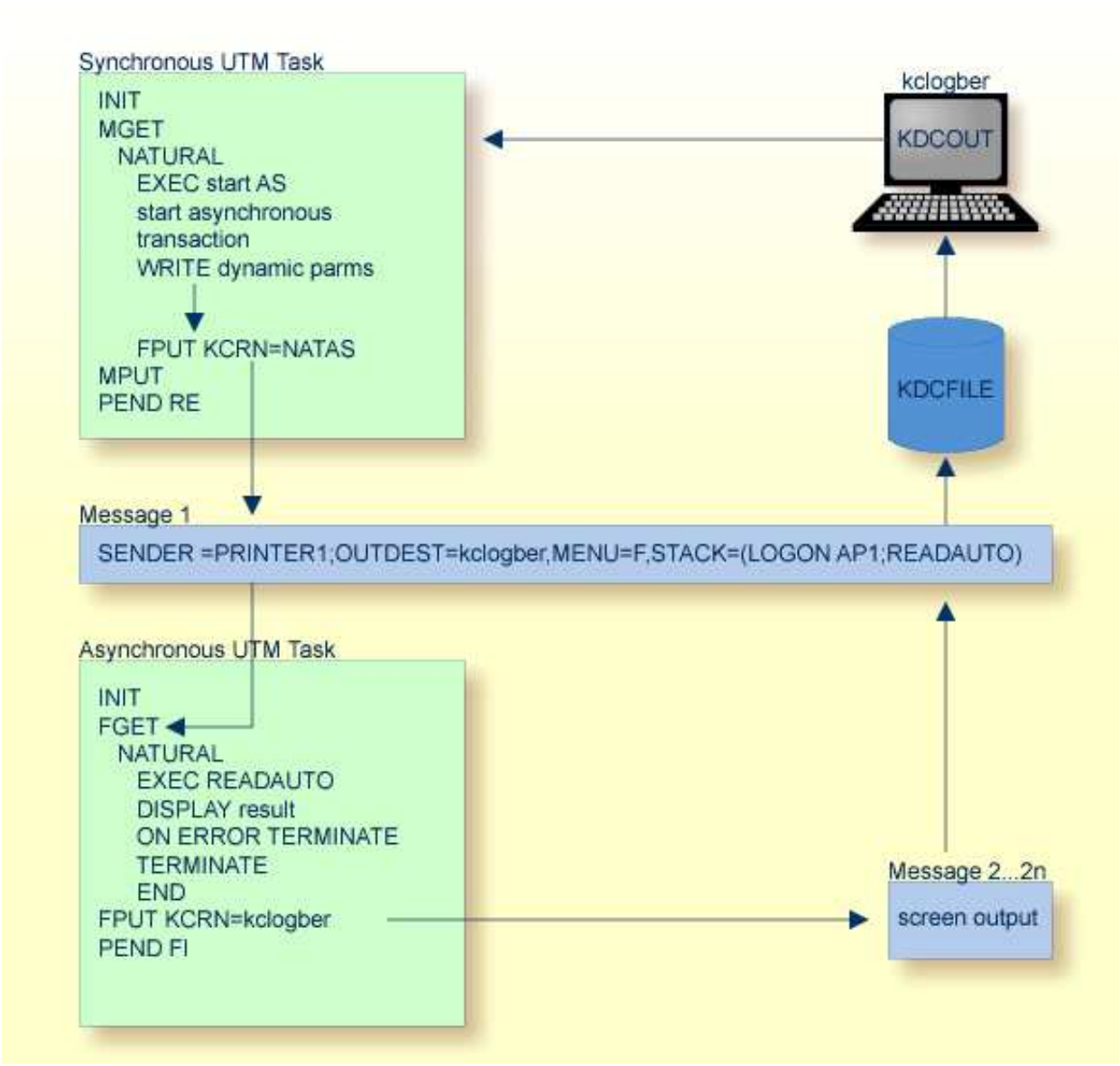
| Note | |
|---|---|
| 1 | The name (dummy) of a printer is moved into field PRDEST. |
| 2 | The logical name of the UTM terminal is moved into field LTDEST. |
| 3 | The message that is to be sent and processed by Natural is assembled, with the following information:<br>● the printer name (in this example, an arbitrary 8-character name),<br>● the logical name (KCLOGTER) of the terminal to which the message is to be sent,<br>● suppression of the main menu (this must be specified),<br>● the application name (Natural logon ID),<br>● the name of the program to be started and to be run in an asynchronous UTM transaction (READAUTO in the example). |
| 4 | When the subroutine NATASYN (in macro NATUTM) is called, a marker is set to indicate that an asynchronous transaction is to be initialized. The subroutine NATASYN conforms to the conventions for calling non-Natural programs. |
| 5 | The Natural offline report is activated. |
| 6 | The message (PARM1) is output by FPUT as an asynchronous transaction. |
| 7 | The Natural offline report is "switched off" by means of an INPUT statement that must have at least one input field. |

An example of the program that is to be executed asynchronously:

```
* READAUTO - EXAMPLE FOR ASYNCHRONOUS TRANSACTION WORKING
READ (75) AUTOMOBILES BY MAKE
WRITE MAKE MODEL HORSEPOWER YEAR
LOOP
ON ERROR DO                          /* --> Note 1
   ERRNO(A4) = *ERROR
   WRITE '*****************************************************'
        /'ERROR NO.: ' ERRNO ' IN ASYNCHRONOUS PROGRAM ' *PROGRAM
        /'*****************************************************'
   TERMINATE
   DOEND
TERMINATE                           /* --> Note 2
END
```

| Note | |
|---|---|
| 1 | An `ON ERROR` routine must be defined in each program that is to be executed asynchronously. The routine must end with a `TERMINATE` statement. |
| 2 | Each program that is to be executed asynchronously must end with a `TERMINATE` statement. |

**Logic of an Asynchronous Transaction within one Natural UTM Application:**




## Asynchronous Processing between two Natural UTM Applications

When processing transactions asynchronously between two Natural UTM applications, the logical UTM terminal name (`LTERM` name) of the synchronous application must be defined with the parameter `SYAPPLI` of macro `NATUTM`, and the logical UTM terminal name (`LTERM` name) of the asynchronous application must be defined with the parameter `ASAPPLI` of macro NATUTM.

**Warning:**
**KDCROOT and KDCDEF must be generated as appropriate for both applications.**

## Example:

```
NUSTART NATUTM SYAPPLI=LNATUTM,ASAPPLI=LNATASY,...
ASYNDRV NATUTM SYAPPLI=LNATUTM,ASAPPLI=LNATASY,...
```

## Example of Synchronous Application:

```
OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATS
ROOT.KDCNATS
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=520
MAX APPLINAME=NATUTM,APPLIMODE=S,KDCFILE=(NATUTM,S)
MAX TASKS=10,ASYNTASKS=5
.
.
EXIT PROGRAM=NUSTART,USAGE=START
EXIT PROGRAM=NUSTART,USAGE=SHUT
EXIT PROGRAM=FREXIT,USAGE=FORMAT
.
.
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM NUSTART
PROGRAM FREXIT
PROGRAM NUERROR
.
.
DEFAULT TAC TYPE=D,PROGRAM=NUSTART,EXIT=NUERROR,CALL=BOTH,...
TAC NAT,ADMIN=NO,TIME=0
TAC NAT1,ADMIN=NO,TIME=0
.
.
DEFAULT TAC TYPE=A,PROGRAM=NUSTART,EXIT=NUERROR,CALL=FIRST,...
TAC NATSY
TAC NATAS
.
.
PTERM NATASY,PRONAM=HOST,PTYPE=APPLI,TERMN=A1,LTERM=LNATASI
DEFAULT PTERM PRONAM=PCDF,PTYPE=T9750,TERMN=FE,CONNECT=N,STATUS=ON
PTERM DFDSS001,LTERM=DF97501
PTERM DFDSS002,LTERM=DF97502
.
.
LTERM LNATASY
DEFAULT LTERM USAGE=D,STATUS=ON,ANNOAMSG=Y,RESTART=YES
LTERM DF97501
LTERM DF97502
.
.
SFUNC F1,RET=21Z
.
.
END
```

## Example of Asynchronous Application:

```
OPTION GEN=ALL,ROOTSRC=INPUT.KDCROOT.KDCNATA
ROOT.KDCNATA
MAX KB=400,SPAB=8192,NB=5120,TRMSGLTH=520
MAX APPLINAME=NATASY,APPLIMODE=S,KDCFILE=(NATASY,S)
MAX TASKS=10,ASYNTASKS=5
.
.
EXIT PROGRAM=ASYNDRV,USAGE=START
EXIT PROGRAM=ASYNDRV,USAGE=SHUT
EXIT PROGRAM=FREXIT,USAGE=FORMAT
.
.
DEFAULT PROGRAM COMP=ASSEMB
PROGRAM ASYNDRV
PROGRAM FREXIT
PROGRAM NUERROR
.
.
DEFAULT TAC TYPE=D,PROGRAM=ASYNDRV,EXIT=NUERROR,CALL=BOTH,...
TAC NAT,ADMIN=NO,TIME=0
TAC NAT1,ADMIN=NO,TIME=0
.
.
DEFAULT TAC TYPE=A,PROGRAM=ASYNDRV,EXIT=NUERROR,CALL=FIRST,...
TAC NATSY
TAC NATAS
.
.
PTERM NATUTM,PRONAM=HOST,PTYPE=APPLI,TERMN=A1,LTERM=LNATUTM
DEFAULT PTERM PRONAM=PCDF,PTYPE=T9750,TRMN=FE,CONNECT=N,STATUS=ON
PTERM DFDSS001,LTERM=97501
PTERM DFDSS002,LTERM=97502
.
.
LTERM LNATUM
DEFAULT LTERM USAGE=D,STATUS=ON,ANNOAMSG=Y,RESTART=YES
LTERM DF97501
LTERM DF97502
.
.
SFUNC F1,RET=21Z
.
.
END
```

Please see also the Siemens UTM documentations. If the asynchronous application is primarily intended for processing asynchronous transactions, storage can be saved by generating this application with a small (local) Natural swap pool of about 64 KB.

**Important:**
The TAC that was defined with the parameter SYNTAC (the default value is NATSY) must always be defined for KDCDEF with TYPE=A; this is an exception to the rules for naming UTM TACs. If, in addition, the synchronous application uses the UTM TACCLASS concept, an asynchronous TAC class must also be allocated to this TAC

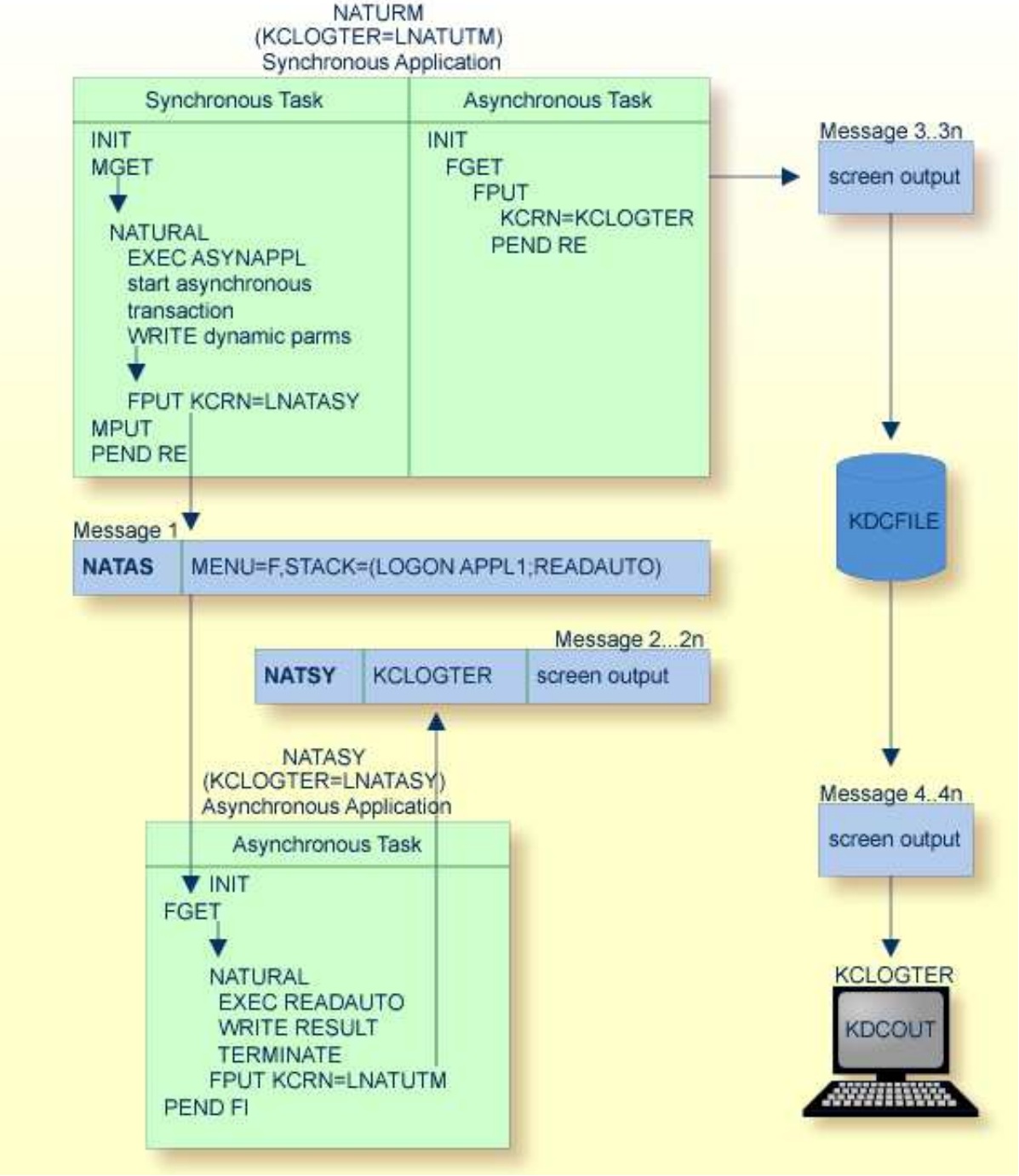
## Example of a Natural Program to Initialize an Asynchronous Transaction between two Natural UTM Applications:

```
* ASYNAPPL - EXAMPLE OF INITIALIZATION FOR ASYNCHRONOUS
*            TRANSACTION WORKING BETWEEN TWO UTM APPLICATIONS
FORMAT LS=145
RESET PARM1(A144) PRDEST(A8) LTDEST(A8) ASYNTAC(A8)
MOVE 'PRINTER1' TO PRDEST                          /* --> Note 1
MOVE *INIT-ID   TO LTDEST                          /* --> Note 2
MOVE 'NATSY'    TO ASYNTAC                         /* --> Note 3
COMPRESS 'NATAS' ' SENDER=' PRDEST ',OUTDEST=' LTDEST
         ',ASYNNAME=' ASYNTAC ','
         'MENU=F,STACK=(LOGON APPL1;READAUTO)'
         INTO PARM1 LEAVING NO                     /* --> Note 4
CALL 'NATASYN'                                     /* --> Note 5
SET CONTROL 'H'                                    /* --> Note 6
WRITE NOTITLE NOHDR PARM1                          /* --> Note 7
INPUT 'ASYNTASK INVOKED - HOPEFULLY' IFELD(A1)     /* --> Note 8
END
```

| Notes | |
|---|---|
| 1 | The name of a printer (simulation) is moved into the field PRDEST. |
| 2 | The logical name of the UTM terminal (KCLOGTER) is moved into the field LTDEST. |
| 3 | The standard TAC for sending "free-running" messages from the asynchronous application to the synchronous application is put in the field ASYNTAC. See also the description of the parameter SYNTAC of macro NATUTM. |
| 4 | The message that is to be sent and processed by Natural is assembled, with the following information:<br>• transaction code for the asynchronous transaction (NATAS),<br>• printer name (in this example, an arbitrary 8-character string),<br>• the logical name (KCLOGTER) of the terminal to which the message is to be sent,<br>• the name of the standard TAC for sending free-running messages from the asynchronous application to the synchronous application,<br>• main menu suppression (this must be specified),<br>• the name of the application (Natural logon ID),<br>• the name of the Natural program to be started and to be run in the asynchronous transaction/application (in this example, READAUTO). |
| 5 | When subroutine NATASYN (in macro NATUTM) is called, a marker for the initialization of an asynchronous transaction is set. The subroutine conforms to the conventions for calling non-Natural programs. |
| 6 | The Natural offline report is activated. |
| 7 | The message (PARM1) is output with FPUT as an asynchronous transaction. |
| 8 | The Natural offline report is "switched off" by means of an INPUT statement with at least one input field. |

The program to be executed asynchronously (READAUTO) must conform to the conventions that apply to asynchronous transaction processing within one Natural UTM application.

## Logic of Asynchronous Transaction between two Natural UTM Applications:




# Printing under UTM

The following topics are covered:

- Using Local Non-Spooled Printers
- Using NATSPOOL (Natural Advanced Facilities)

- Other Spooling Systems

## Using Local Non-Spooled Printers

A Natural program that wishes to use local printers without spooling (that is, with FPUT via UTM), should proceed as shown in the following example.

### Example:

```
* TESTPRNT - TEST FOR THE Natural OFFLINE REPORT
RESET PARAM(A9)
REDEFINE PARAM (PARAM1(A1) PARAM2(A8))
MOVE 'H' TO PARAM1                              /* --> Note 1
MOVE 'PRINTER1' TO PARAM2                       /* --> Note 2
SET CONTROL PARAM                               /* --> Note 3
READ (50) AUTOMOBILES BY MAKE
WRITE NOTITLE NOHDR MAKE MODEL HORSEPOWER YEAR  /* --> Note 4
LOOP
EJECT
INPUT 'PRINT ORDER WAS EXECUTED' IFELD(A1)      /* --> Note 5
END
```

| Notes | |
|---|---|
| 1 | The Natural offline report is activated by putting an H in the first position of the field PARAM. |
| 2 | The logical UTM printer name is defined starting at the second position of the field PARAM. |
| 3 | The SET CONTROL statement, together with the content of the field PARAM, activates the Natural offline report and specifies the name of the printer. To ensure compatibility for existing programs written using Natural Version 1, the programs CMLIST and NATPRNT continue to be available; see *Utility Programs for Use with Natural under UTM*. |
| 4 | The print records are passed to UTM from the Natural UTM Interface using FPUT. |
| 5 | The INPUT statement (which must have at least one input field) deactivates the Natural offline report. |

All the necessary steps for using local printers must have been taken when generating UTM; for further details, please refer to the Siemens UTM documentation. The appropriate UTM administration commands can be used to verify that a connection to the defined printers exists.

## Using NATSPOOL (Natural Advanced Facilities)

The parameter SPOOL of macro NATUTM is provided for using NATSPOOL under UTM. Further details are given in the section *Parameters of Macro NATUTM*. Please refer also to the BS2000/OSD-specific installation information in the *Natural Advanced Facilities* documentation.

If in an asynchronous UTM transaction printing is to be done with NATSPOOL, the TERMINATE statement must be preceded by an END OF TRANSACTION statement.

### Other Spooling Systems

The User Exit RP2PRNT is provided for interfacing to other spooling systems. This user exit is activated if REPRO-2000 is specified with the parameter SPOOL in macro NURENT. (This value should be used for all spooling systems.)

Since it must be linked with the reentrant part of the Natural UTM application, the user exit routine RP2PRNT must be reentrant.

The logic of the transfer of print records from Natural, buffer processing, etc., can be seen in macro NURENT (labels CMWTERM and CMWHC) and the appropriate routines in macro NATUTM.

As an alternative, it is possible to use the User Exit RMSPOOL; see section *User Exits* above.

Software AG does not provide support for this interface to other spooling systems except as described in the preceding paragraphs.

# Calling Non-Natural Programs

Non-Natural programs are called using the standard register conventions for inter-program communication. If the program to be called is reentrant (uses shared code), it can be defined with the profile parameter CSTATIC in the Natural parameter module (macro NTPRM ) and linked with the reentrant part of the UTM application. Otherwise, one of the following procedures can be used:

- The programs can be dynamically loaded at runtime. To do this, the programs must be in the library defined with the profile parameter LIBNAM in the Natural parameter module or in the BLSLIB libraries specified in the UTM start job;
- The programs can be linked with the front-end part of the Natural UTM application. To do this, the names of the programs must be defined in the operand of the parameter LINK, LINK2, LINK3 or LINK4 of macro NATUTM. This procedure is always necessary for programs that contain an EXTRN reference to an ENTRY that is already present in the front-end part of the Natural UTM Interface. The Natural UTM Interface executes a TABLE macro call for the programs that have been defined in this way. This makes an entry in the dynamic loader's LINK table, indicating that it is not necessary to dynamically load the programs when they are called by the Natural program.

In both cases, the maximum number of called non-Natural programs must be defined with the parameter CDYNAM of macro NATUTM; see *Parameters of Macro NATUTM*.

**Note:**
If parameter KB in macro NATUTM is set to YES, Natural always passes the address of the UTM communication area KB (*Kommunikationsbereich*) as the first parameter address. This does not apply to programs which are defined with profile parameter CSTATIC.

# Calling UTM Chained Partial Programs

Several methods are provided for ending a Natural session (FIN or TERMINATE) with a PEND PR(OGRAM) instead of a PEND FI(NISH), so that another UTM partial program is called:

- The UTM TAC for the UTM partial program that is to be called can be passed using the Natural dynamic parameter PROGRAM at the start of the Natural session, for example:

  ```
  STACK=(LOGON APPL1;MENU),PROGRAM=NAT10
  ```

- The UTM TAC for the UTM partial program that is to be called can be defined in the operand of the parameter PENDPR of macro NATUTM, for example:

  ```
  NATUTM PENDPR='NAT10'
  ```

- The utility program TACSWTCH can be used.

In all cases, the Natural UTM Interface would execute a PEND PR(OGRAM) with the UTM TAC NAT10 at the end of the Natural session, which means that the UTM partial program associated with this TAC would be started.

Another way to execute a PEND PR(OGRAM) is by activating the function key defined for this purpose, which suspends, but not terminates, the Natural session. On return from the UTM partial program with PEND PR(OGRAM), the Natural session can be continued from the point at which it has been suspended; see also the parameter PRKEY. If the function key for PEND PR(OGRAM) is activated without a UTM TAC for another UTM partial program being defined, an appropriate error message is displayed.

**Note:**
The programs NUEXAMPL, UTMNAV and UTMCOB show examples of the logic necessary in a UTM partial program that wishes to communicate with the Natural UTM Interface (and therefore with Natural itself) - see the descriptions of programs UTMCOB and UTMNAV in the section *Software Exchange*.

# Calling Adabas from Non-Natural Programs in a Natural UTM Application

If a Natural program calls a non-Natural program that also includes Adabas calls, the appropriate field in the Adabas control block must be supplied with the current Adabas user ID. In this case, generate the CSECT ADACALL in the Natural UTM Interface.

ADACALL contains an entry which is defined with the parameter ADACALL in macro NATUTM (the default value of this parameter is ADABAS).

This entry is activated for every CALL [ADABAS] USING .... The current Adabas user ID is passed to the field ADDITIONS2 of the Adabas command block, and subsequent processing of the Adabas call is passed to the Adabas interface module ADALNN; see also the parameter ADACALL.

# Terminating a UTM Task Abnormally

The Natural session (and thereby also the UTM task) can be abnormally terminated by entering the CANCEL parameter's value of the Natural parameter module (default is *CANCEL in upper-case letters).