

---

# 1 Preface

The *open*UTM Universal Transaction Monitor is a comprehensive middleware platform, offering a wealth of options for designing and implementing transaction-oriented OLTP applications, as well as the functionality of a complete message queuing system.

Thanks to its optimum performance, sophisticated security functions, and high availability, *open*UTM is also suitable for situations in which conventional OLTP systems have long been pushed to their limits.

*open*UTM forms a secure, efficient framework for modern, multi-tier client/server architectures. Among other things, it controls global transactions, optimizes the utilization of system resources (memory, CPU, etc.), manages parallel access, takes care of access control, and sets up network connections.

The name “*open*UTM” says it all:

- open*** ... because *open*UTM complies with the reference model for Distributed Transaction Processing (DTP) defined by X/Open and supports the open interfaces standardized by X/Open.
- U**niversal ... because *open*UTM links different environments and is designed for use in the most varied scenarios: it integrates heterogeneous networks, platforms, resource managers, and applications.
- T**ransaction ... because *open*UTM guarantees complete global transaction security in accordance with the classical ACID properties of atomicity, consistency, isolation and durability.
- M**onitor ... because *open*UTM not only offers “pure” transaction processing, but also allows for the management of distributed, enterprise-wide IT solutions.

## 1.1 Summary of contents and target group

This manual is intended to support programmers writing *openUTM* applications in Fortran in their work. It is a supplement to the *openUTM* manual "Programming Applications with KDCS for COBOL, C and C++".

A basic knowledge of the operating system and *openUTM*, as well as of the core manual "Programming Applications with KDCS for COBOL, C and C++" is required. For more detailed information, the *openUTM* manuals "Generating and Administering Applications", "Messages, Debugging and Diagnostics" and "Concepts and Functions" should be consulted.

This manual describes the language-specific points to be observed when writing Fortran program units.

It provides sample programs written in Fortran for individual KDCS calls and for the event service MSGTAC, as well as an example for a complete *openUTM* application.

The Fortran data structures are listed in chapter "Data structures for Fortran" on page 59ff).

### README file

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific README file.

On a BS2000 computer, you will find the README file under the file name *SYSRME.product.version.language*. Please ask your system supervisor for the user ID on which the README file is located. You can view the README file with the /SHOW-FILE command or in an editor or you can print it to a standard printer with the following command:

```
/PRINT-DOCUMENT filename , LINE-SPACING=*BY-EBCDIC-CONTROL
```

If you have a SPOOL version prior to 3.0A:

```
/PRINT-FILE FILE-NAME=filename , LAYOUT-CONTROL=  
PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

---

## 2 Structure of Fortran program units

This chapter tells you

- how to write a Fortran program unit as a subroutine
- what points to observe when formulating data declarations in a Fortran program unit
- what the code section should look like and how to program a KDCS call in Fortran
- what special features (shared code, addressing aids, etc.) and restrictions apply to Fortran program units.

The basis for the description is the Fortran77 compiler FOR1 (Version 2.2C) and the Fortran90 compiler (Version 1.0A). FOR1 supports the UTM link in BS2000 as of FOR1 Version 2.1A. Any references below to the Fortran compiler thus refer to the FOR1 and Fortran90 compilers, and if only one of these compilers is meant this is mentioned explicitly.

### 2.1 Fortran program units as subroutines

UTM program units are subroutines of the UTM main routine. This has the following consequences for Fortran:

- UTM Fortran program units are subroutines. The subroutine name defines the entry point (see page 4).
- At least one data structure must be defined in the data declarations (data section) (see page 5).
- The program unit is terminated dynamically with the PEND call, except in the case of the event exits, which are exited with the END statement (see the section entitled "Event functions" in the core manual "Programming Applications with KDCS for COBOL, C and C++".  
The STOP statement is prohibited.

There must be no RETURN statement before the PEND call. A RETURN statement after the PEND call is purely formal.

In FOR1, no data structures are supported as they are, for example, in Fortran90, COBOL, Pascal and C. In a UTM application, UTM modules use these structures to communicate with program units in the various languages. In order to enable such communication between UTM modules and Fortran program units under the Fortran77 standard, the necessary data structures are simulated on a relatively large scale by EQUIVALENCES and COMMONs in the UTM-FOR1 link (and, for reasons of compatibility, this applies also to Fortran90). It is in this sense that the term "data structures" is used in this description.

For compatibility purposes and in order to work with error-free entries for simulating the data structures, a series of INCLUDE elements is available. The description of how to use these INCLUDE elements begins on page 8; the elements themselves are listed in chapter "Data structures for Fortran" on page 59ff).

## 2.2 Subroutine name as the entry name

You must define the entry name (entry point) of the program unit in the SUBROUTINE statement.

This name is freely selectable within the framework of the Fortran90 or Fortran77 conventions. It must be unique within an application program, i.e. there must be no name conflicts between the program name, the runtime systems, the database systems, the formatting system and UTM.

In particular, all names beginning with KDC, KC or I are reserved.

The program unit name (entry point) must also be specified in the KDCDEF statement PROGRAM when the application is generated (see the *open*UTM-Handbuch „Anwendungen generieren und betreiben“).

## 2.3 Data section of a Fortran program unit (data declarations)

### 2.3.1 Communication area and KDCS parameter area

Every program unit, including event services and event exits, must contain in the declaration section (data declarations) a data structure simulated by EQUIVALENCE statements which describes the KDCS communication area. This data structure is created as a COMMON with the fixed name KDCKB and is defined in INCLUDE element KCCAF.

This can be followed by a further data structure which describes the standard primary working area (SPAB). This structure is used if the program unit contains at least one KDCS call. In addition to the KDCS parameter area it is possible to accommodate the message areas in the SPAB. As in the case of the communication area, this data structure is implemented with the name KDCSPAB by means of EQUIVALENCE statements and a COMMON. The associated INCLUDE element is called KCPAF.

The data structures of the communication area and the KDCS parameter area are contained in the library SYSLIB.UTM.040.FOR as INCLUDE elements (see page 8). You have to define the message areas yourself. If you are working with FHS (Format Handling System), you can create your own addressing aids (see page 16).

When starting a program unit that was generated with COMP=FOR1 and compiled with LINKAGE=FOR1-SPECIFIC (old FOR1 linkage), the FOR1 linkage module is called. This module deletes the first two parameters (KB and SPAB), which FOR1 does not need, from the parameter list.

#### *Example*

```

%INCLUDE      KCCAF                      (1)
CHARACTER    CAPA * 128
COMMON       /KDCKB/      CAPA           (2)
%INCLUDE      KCPAF                      (3)
CHARACTER    NB * 360
COMMON       /KDCSPAB/    NB             (4)
%INCLUDE      FORMA1
EQUIVALENCE ( NB, FORMA1)
%INCLUDE      FORMA2
EQUIVALENCE ( NB(200:), FORMA2)        (5)

```

- (1) KDCS communication area.
- (2) User-specific declaration of the KB program area (extension of COMMON area KDCKB).
- (3) SPAB with KDCS parameter area.
- (4) The COMMON area for the SPAB data structure is extended by the message area.
- (5) Addressing aids for the FORMA1 and FORMA2 formats are accommodated in the message area, i.e. in the COMMON KDCSPAB.

### 2.3.2 Additional shareable data areas

In addition to the communication area and the parameter area, you can set up a maximum of 99 further areas which can then be used within a UTM application as shared data areas.

You can change the contents of these areas from within the application programs but *openUTM* does not offer any support for serializing access or transferring the modified data to the next application run. You can prevent the contents of such an area being changed by means of the operand ACCESS=READ in the MPOOL statement at generation time.

The additional data areas are declared in the Fortran program units as CHARACTER strings of fixed length. The maximum length of an area is therefore restricted to 32K - 1 (= 32767 bytes).

These areas are defined with the AREA statement when the UTM application is generated. The sequence of the areas and their types (local or global) are defined at the same time (see the *openUTM-Handbuch* „Anwendungen generieren und betreiben“).

#### *Example*

The area AREA1 is to be used as a common data area with a length of 2000 bytes. It makes sense to generate the area as an Assembler program:

```
AREA1      CSECT
           DS      2000C
           END
```

### How to use areas of this type in your Fortran programs:

In the SUBROUTINE statement you define symbolic addresses of the additional shareable data areas as formal parameters. The sequence in which these areas are defined in the AREA statement is significant. If an area defined at the nth position is required, you have to specify all areas up to this point in the SUBROUTINE statement.

When starting a program unit that was generated with COMP=FOR1 and compiled with LINKAGE=FOR1-SPECIFIC (old FOR1 linkage), the FOR1 linkage module is called. This module deletes the first two parameters (KB and SPAB), which FOR1 does not need, from the parameter list.

#### *Example*

##### FOR1 linkage:

```
SUBROUTINE TPR1 (AREA1,AREA2,AREA3)
CHARACTER *2000  AREA1,AREA2,AREA3
.
.
.
END
```

##### ILCS linkage:

```
SUBROUTINE TPR1 (DUMMY1,DUMMY2,AREA1,AREA2,AREA3)
CHARACTER *2000  AREA2,AREA3
CHARACTER *1    DUMMY1,DUMMY2
.
.
.
END
```

This function is not part of DIN standard 66 265.

The FOR1 compiler option TESTOPT = (ARG) and the Fortran90 TEST-SUPPORT option with CHECK-CODE=PARAMETERS(PROCEDURE-ARGUMENTS=YES) must not be used. By the same token, the FOR1 compiler option TESTOPT = (ALL) and the Fortran90 TEST-SUPPORT option with CHECK-CODE=ALL, which include these options, must not be set.

### 2.3.3 Data structures for UTM Fortran programs

In order to structure data areas, INCLUDE elements containing these data structures simulated in Fortran are supplied with UTM. The data structures available for Fortran programs and their functions are shown in the table below.

Name	Contents and meaning
KCCAF	Data structure for the KDCS communication area, containing: <ul style="list-style-type: none"> <li>- current data of the service and program,</li> <li>- return information following a call to UTM and</li> <li>- the KB program area for data transfer between programs within a service.</li> </ul> You must define the KB program area and its fields yourself (see the example later in this section).
KCPAF	Data structure for the KDCS parameter area: This area is used for the parameters of a KDCS call.
KCDFF	KDCS screen functions: You can use this symbolic name to influence the screen display by entering the name of the desired function in the KCDF field of the KDCS parameter area (see page 80).
KCINFF	Data structure for the INFO call: You should place this data structure over the message area if you want to use the KDCS call INFO DT/SI/PC.
KCMMSGF	Data structure for the UTM messages: You need this data structure when handling messages in an MSGTAC routine (see also the <i>openUTM</i> manual "Programming Applications with KDCS for COBOL, C and C++") or when you want to interpret the SYSLOG file with your own program.
KCDADF	Data structure for the DADM call: You should place this data structure over the message area if you want to use the KDCS call DADM RQ.
KCPADF	Data structure for the PADM call: You should place this data structure over the message area if you want to use the KDCS call PADM AI/PI.
KCAPROF	Optional second parameter area for the APRO call: This area is used for the selection of specific OSI TP function combinations and the security type.
KCINPF	Data structure for the INPUT exit: This data structure contains the input and output parameters of the INPUT exit.
KCATF	KDCS attribute functions: When using + formats, you can modify the attribute fields of the formats by means of the symbolic name for attribute functions.



Name	Contents and meaning
KCCFF	This defines the second parameter transferred by UTM with the event exit INPUT. In this parameter UTM transfers the contents of the control fields of screen formats to the program unit. For this reason this parameter is also known as the control fields area.
KCINIF	This defines a second parameter area for the INIT call (necessary only with INIT PU). In this parameter area UTM returns the information requested with INIT PU.

In Fortran the KB communication area and the standard primary working area SPAB are implemented as COMMONs and are addressed jointly by the linkage editor with the UTM areas of the same name. It is not permissible to use a BLOCK DATA module to initialize these COMMON areas in Fortran program units.

If you intend to extend these COMMONs with your own areas (such as a program area in the KB or a message area in the SPAB), you must keep to the COMMON names KDCKB and KDCSPAB assigned by UTM. The other Fortran INCLUDEs are simply EQUIVALENCE groups. The data structures KCINFF, KCDADF and KCPADF should be put above the message area.

The following example demonstrates how to use UTM Fortran INCLUDEs and extend the COMMON.

*Example*

```

SUBROUTINE TPR1
%EXPAND OFF
%INCLUDE KCCAF
CHARACTER CAPA * 128
COMMON /KDCKB/ CAPA
%INCLUDE KCPAF
CHARACTER NB * 360
COMMON /KDCSPAB/ NB
%EXPAND ON
.
.
.
```

These INCLUDE elements are listed in chapter “Data structures for Fortran” on page 59ff.

## 2.4 Command section of a Fortran program unit

You are free to design the command section of a Fortran program unit as you wish. The only restriction is that you have to observe the transaction processing rules, as described in detail in the chapter dealing with the structure and use of UTM programs in the *openUTM* manual “Programming Applications with KDCS for COBOL, C and C++”. This refers to:

- program unit as a subroutine of KDCROOT
- reentrant programming
- strict dialog (in dialog programs)

The name KDCROOT identifies the UTM main routine. The source program for KDCROOT is generated with the KDCDEF generation tool (see the *openUTM-Handbuch* „Anwendungen generieren und betreiben“).

Event exits are governed by special rules which are described in section “Program skeleton” on page 14.

### 2.4.1 Program skeleton

The command section of a Fortran dialog or asynchronous program unit must start with a KDCS INIT call and is exited with a KDCS PEND call. Control is returned to KDCROOT by means of the PEND call. The END statement merely defines the physical end of the subroutine, whereas the PEND call functions as the logical end of the subroutine (similar to RETURN). RETURN and STOP statements are prohibited in the Fortran program units.

The following example illustrates the structure of a Fortran dialog (or asynchronous) program unit (for KDCS calls see section “Program skeleton” on page 12).

```

SUBROUTINE TPR2
%EXPAND OFF
%INCLUDE  KCCAF                               (1)
CHARACTER  CAPA * 128                         (2)
COMMON  /KDCKB/ CAPA                         (3)
%INCLUDE  KCPAF                               (4)
.
.
KCOP      = 'INIT'
KCLCAPA  = 128
KCLSPA   = 512
CALL KDCS ( KCSPA )
.
.                                           (5)
.
KCOP      = 'PEND'
KCOM      = 'FI'
CALL KDCS ( KCSPA )                         (6)
END

```

- (1) This defines COMMON area KDCKB (communication area KB).
- (2) A program area of 128 bytes is defined in the communication area.
- (3) The program area is appended to the communication area (COMMON KDCKB).
- (4) This defines COMMON area KDCSPAB (standard primary working area, SPAB).
- (5) Processing section.
- (6) Control is transferred to UTM.

If a Fortran program unit calls other Fortran subroutines (subroutines or functions) the RETURN statement may be used for returning to the calling program unit. The STOP statement, however, is prohibited even in the subroutines of the program unit. See the following example:

```
SUBROUTINE TPR3
  .
  .
  .
  KCOP      = 'INIT'
  CALL KDCS ( KCSPA )
  .
  .
  CALL UTPR
  .
  .
  .
  KCOP      = 'PEND'
  CALL KDCS ( KCSPA )
  END

SUBROUTINE UTPR
  .
  .
  .
  RETURN
  .
  .
  .
  END
```

## 2.4.2 Calling UTM functions (KDCS calls)

Before you can call a UTM function in the program all the necessary parameters must be set in the KDCS parameter area.

These include

- the operation code of the call (KCOP)
- additional parameters determined by the operation code (see the chapter entitled "The KDCS calls" in the *openUTM* manual "Programming Applications with KDCS for COBOL, C and C++").

The data structures for the KDCS parameter area are defined in INCLUDE KCPAF; the names of the variables and COMMONs are binding for the user.

The KDCS call cannot be issued until all the necessary data areas have been supplied with values. The entry point for all operations is "KDCS".

The format of the KDCS call is as follows:

```
CALL KDCS (param1 [, param2 ]) .
```

**param1** is the data name of the KDCS parameter area. When the corresponding INCLUDE element is used (see page 8), this name is "KCSPA". It is a mandatory specification.

**param2** is the data name of the storage area from which the program unit transfers data to UTM or to which UTM supplies data (e.g. message area).

### *Example*

**KDCS call with one parameter:**

```
KCOP    = 'INIT'
KCLCAPA = 128
KCLSPA  = 512
CALL KDCS ( KCSPA )
```

**KDCS call with two parameters:**

```
KCOP    = 'MGET'
KCLA    = 360
KCFN    = '#FORMAT'
CALL KDCS ( KCSPA , NB )
```

Please note that all the program units in UTM have to be written in reentrant code.

### 2.4.3 STXIT handling

#### Program units with COMP=ILCS

For FOR1 and Fortran90, COMP=ILCS should be used for the UTM link. The default value for FOR1 is COMP=ILCS.

In the case of COMP=ILCS, FOR1 generates an output for the STXIT event classes PROCHCK and ERROR. Following this, the runtime system terminates the program with TERM; this is intercepted by UTM (70Z/XT90).

#### Program units with COMP=FOR1

In a UTM application there must be no STXIT events defined in the language-specific environment. STXIT logon in Fortran is not executed by the runtime system. The STXIT events are intercepted centrally by UTM and (only) event classes PROCHK and ERROR are forwarded by UTM via the UTM FOR1 linkage module KDCCFOR1 to the FOR1 runtime system, where Fortran error handling then takes place (see the "FOR1" User Guide). The other STXIT events are either suppressed or handled directly in UTM, e.g. as in the case of TERM.

STXIT handling is always executed in the UTM Fortran environment; there is no way in which STXIT handling can be disabled. The runtime option RUNOPT STXIT = NO (entry in UTM only via a file with the link name FOR1RUN) has no effect. The signing on of STXIT events in the I\$STXIT (or IF@STXIT) routine has no effect in the UTM environment.

Program termination is not executed in Fortran. Control is returned to UTM. This means that a STOP statement is ignored. Besides, a STOP statement would mean an inconsistent return to UTM, i.e. not via the KDCS call PEND as prescribed. STOP and PAUSE statements should therefore not be used in a UTM environment.

For Fortran error handling, the runtime modules IF@DBG2 and IF@ERR2 are loaded dynamically if they are not explicitly linked in the linkage run or if you are working with SYSLNK.FOR1-LZS.022.DYNAMIC; otherwise they have to be assigned via TASKLIB (see the "FOR1" User Guide).

## 2.5 Event exits

The event exits START, SHUT and VORGANG (= "service") must not contain any KDCS calls. They must be written as subroutines which are terminated with the END statement.

With START, SHUT and VORGANG it is possible to access fields (variables) of the communication area (KB). Accordingly, you must declare the KB area in the data section in the same way as program units with KDCS calls.

The following points apply only in conjunction with COMP=FOR1:

If an error occurs in the program in connection with START or SHUT, e.g. if a file cannot be opened, the task is terminated after interrogation of the Fortran runtime system. An example of a combined START/SHUT exit is given on page 26.

For technical reasons the event exit INPUT is not supported in Fortran.

## 2.6 Special points relating to Fortran

This section tells you

- what points to observe with FOR1 program units when compiling and linking the KDCROOT main routine
- how to create and use Fortran addressing aids
- how to make Fortran programs shareable for shareable modules
- what other compiler-dependent points need to be observed.

### 2.6.1 Compiling the KDCROOT main routine (FOR1)

When compiling the KDCDEF-generated source of the ROOT table module KDCROOT, you should remember that, when LINKAGE=FOR1-SPECIFIC für PROGRAM ..., COMP=FOR1 is specified, in addition to the defined macro libraries (UTM, FMS, etc.), the Fortran macro library SYSLIB.FOR1.022 also has to be assigned. The LWKAF macro, which is used when generating the main routine, is called from this library. Instead of assigning the Fortran macro library as well, you can also copy the LWKAF macro to the UTM macro library.

In the case of FOR1 programs created with LINKAGE=STD and of Fortran90 programs created with LINKAGE=ILCS, the LWKAF macro is not required.

An example of the compilation procedure is given in the example of a complete UTM Fortran application (page 26ff).

### 2.6.2 Linking a UTM Fortran application (FOR1)

When linking the ROOT table module KDCROOT, the following must be observed:

UTM Version 4.0 is XS-compatible. The linked phase is executable in conjunction with FOR1 in the upper address space. The shareable FOR1 runtime system SYSLNK.FOR1-LZS.022.DYNAMIC is needed for this and load address LOADPT = \*XS must be specified in the PROG linkage statement.

An example of the linkage procedure is given in the example which starts on page 26.

PROG-MOD=ANY must be specified at the start of the UTM Fortran application when using the KDCLOAD utility routine.

For more detailed information, see the "FOR1" User Guide.

### 2.6.3 Linking a UTM Fortran application (Fortran90)

Unless otherwise specified, Fortran90 generates LLMs (link and load modules) and requires the linkage editor. No special runtime library is required in the case of Fortran90 (see the “Fortran90” User Guide).

### 2.6.4 Creating formats with IFG

The “IFG” manual explains in detail how to create formats with the Interactive Format Generator IFG. When creating these formats for use in Fortran, pay attention to the following points:

- The format name must not be more than 7 characters long.
- Select "structure of the data transfer area" in the user profile
  - separate attribute blocks and field contents; the format then generated is the #format.

IFG does not support the +formats and \*formats in Fortran; any attempt to generate such formats is rejected by IFG.

The following example shows how to use the addressing aids created by IFG:

*Example*

```
%INCLUDE      KCPAF                (1)
CHARACTER    NB * 360
COMMON      /KDCSPAB/           NB    (2)
%INCLUDE     FORMA
EQUIVALENCE ( NB , FORMA )      (3)
```

- (1) SPAB area defined.
- (2) COMMON for SPAB extended by the message area.
- (3) Addressing aid stored in the message area.

If you intend to use the message area simply for accommodating the addressing aids, you can simplify the declaration even further:

```
%INCLUDE      KCPAF
%INCLUDE     FORMA
COMMON      /KDCSPAB/           FORMA
```

If you wish to use more than just one format in the UTM application, you must extend the common block KDCSPAB with each format.



Please bear in mind that the formats created by IFG for Fortran (`#formats`) are defined as packages of EQUIVALENCE statements which consist of three basic elements:

- global attribute block
- field attribute block
- field data part

An example of a `#format` (FORMA) is given on page 26. In the last few declarations you can see how the three basic elements (FORMAGLOBALS, FORMAATTR, FORMADATA) redefine the area FORMA.

When extending the COMMON for the SPAB area you must list the entire area (FORMA in the example), not just the data section (FORMADATA), for instance. This would constitute an invalid use of the EQUIVALENCE and COMMON statements.

## 2.6.5 Shareable modules

There are three ways of loading shareable modules:

- shareable modules can be loaded by UTM into the common memory pool in the user area (class 6 memory)
- up to and including BS2000 OSD V2, shareable modules can be loaded into class 4 memory using the ADD-SHARED-PROGRAM command (FOR1 only)
- shareable modules are loaded as a subsystem.

Details can be found in the *open*UTM-Handbuch „Anwendungen generieren und betreiben“.

The following example applies only to FOR1. In the case of Fortran90, shareable modules can only be generated if link and load modules are created in LLM format without additional options. These objects must then be processed with the linkage editor.

The following points must be observed when programming and compiling shareable program units. The symbolic names have been taken from the example on page 26. At the end of the section on page 54 a complete procedure is given for the execution of the UTM application with shareable program units (in a common memory pool).

1. By specifying the compiler option `COMOPT OBJECT=(SHARE)`, a shareable code is created for the program units during compilation. With the aid of the `SHARE-LIBRARY` option the shareable modules can be stored directly in a separate PLAM library.

```
/EXEC $FOR1
COMOPT SOURCE=UTM.BSP.SRC,OBJECT=(SHARE)
COMOPT MODULE-LIBRARY=UTM.TP.PLAMLIB.NOSHARE
COMOPT SHARE-LIBRARY=UTM.TP.PLAMLIB.SHARE
END
```

2. The names of the shareable program units are written in a file (with the file name SHRNAM in this instance).

```

/EXEC $EDT
TPREAD
TPUPDAT
OPNCLOS
BADTAC
@W' SHRNAM'
@H

```

3. Call the procedure FOR1.P.SHARE or SYSPRC.FOR1.022.SHARE. This procedure creates adapter modules for the nonshareable data section and shareable code section and links these modules in front of the nonshareable or shareable section. In the following example a prelinked module with the name SHRMOD is created. A description of the procedure FOR1.P.SHARE can be found in the "FOR1" User Guide.

```

DO FOR1.P.SHARE
... SHRNAM
... UTM.TP.PLAMLIB.SHARE
... UTM.TP.PLAMLIB.NOSHARE
... LMS
... ADAPTS
... ADAPTN
... SHRMOD

```

4. When generating the UTM application the shareable program units should be defined as follows:

- a) Loading into the common memory pool (here MPOOL1)

```

.
.
.
OPTION GEM=ALL,ROOTSRC=UTM.SRC.ROOT (1)
OPTION SHARETAB=UTM.SRC.SHARETAB
.
.
MPOOL MPOOL1,SIZE=2,SHARETAB=SHARETAB,LIB=UTM.TP.PLAMLIB.SHARE, (2)
ACCESS=READ,SCOPE=GLOBAL

DEFAULT PROGRAM LOAD=STATIC,LIB=UTM.TP.PLAMLIB.NOSHARE (3)
PROGRAM TPREAD,COMP=ILCS
PROGRAM TPUPDAT,COMP=ILCS
PROGRAM OPNCLOS,COMP=ILCS
PROGRAM BADTAC,COMP=ILCS

```

```
MODULE SHRMOD,LOAD=(POOL,MPOOL1),LIB=UTM.TP.PLAMLIB.SHARE (4)
```

```
MODULE FORMA,LOAD=(POOL,MPOOL1),LIB=UTM.TP.PLAMLIB.SHARE (5)
```

- (1) Create a KDCSHARE file with the name UTM.SRC.SHARETAB.
- (2) Define the characteristics of the common memory pool MPOOL1.
- (3) Define the nonshareable data sections of the program units.
- (4) Prelinked module SHRMOD contains the adapter module ADAPTS and the shareable code sections of the program units.
- (5) The format FORMA is also loaded into the common memory pool.

Creating the SHREATAB module:

```
/EXEC $ASSEMB
*COMOPT SOURCE=UTM.SRC.SHARETAB
*COMOPT MODULE=UTM.TP.PLAMLIB.SHARE(SHARETAB)
*END HALT
```

#### b) Loading into class 4 memory

```

.
.
DEFAULT PROGRAM LOAD= { STATIC } , LIB=UTM.TP.PLAMLIB.NOSHARE
                       { STARTUP }

```

```
PROGRAM TPREAD,COMP=ILCS
PROGRAM TPUPDAT,COMP=ILCS
PROGRAM OPNCLOS,COMP=ILCS
PROGRAM BADTAC,COMP=ILCS
.
```

The shareable code sections of the program units are loaded into class 4 memory by the system administrator with the command

```
/ADD-SHARED_PROGRAM SHRMOD
```

This method is no longer possible as of BS2000/OSD V2. Instead the shareable section can be loaded as a subsystem (see the "Subsystem Management" manual).

For further information see the *openUTM-Handbuch* „Anwendungen generieren und betreiben“.

## 2.6.6 Restrictions affecting the Fortran language elements under UTM

1. The STOP statement must not be used in Fortran program units and their subroutines. In a program unit the RETURN statement may only appear after the PEND call but does not have any function. In the subroutine of the program unit the RETURN statement retains its normal functions. The PAUSE statement should not be used at all.
2. The name prefixes KDC, KC and I are reserved for UTM or for runtime systems in the appropriate programming languages.
3. No additional entry points (ENTRY statement) are permitted in program units.
4. The compiler options TESTOPT = (ARG) and TESTOPT = (ALL), as well as the Fortran90 TEST-SUPPORT option with CHECK-CODE=PARAM-ETERS(PROCEDURE-ARGUMENTS=YES) and with CHECK-CODE=ALL, are prohibited.
5. The use of BLOCK-DATA programs for initializing UTM areas is prohibited.
6. In the UTM environment the FOR1 runtime option RUNOPT STXIT = NO and the I\$STXIT routine have no effect.
7. Dynamically created fields can be used only within **one** program unit; it is therefore not advisable to use dynamic fields at all under UTM.

## 2.6.7 Fortran-specific version dependencies

If a FOR1 program unit with LINKAGE=STD (ILCS linkage) is compiled, it must be generated with COMP=ILCS. In this case the ILCS linkage module, which does not convert the parameter list, is called when starting the program unit. If a FOR1 program unit has a parameter list, this means that the parameter list has to be extended to include these two first parameters KB and SPAB for conversion for ILCS linkage. These parameters must be declared as dummy parameters, i.e. they must not be used. KB and SPAB are transferred via COMMON, as before.

---

## 3 Examples in Fortran

This chapter gives simple examples for coding a KDCS call and an example of a complete UTM application, including the KDCDEF generation.

### 3.1 Examples of individual KDCS calls

This section contains coding examples for the following KDCS calls:

- MGET
- MPUT
- DPUT

As the remaining KDCS calls are coded in the same way, no explicit description of them is given here.

In the KDCS call, KCSPA is the address of the KDCS parameter area and NB the address of the message area.

## MGET call

1. An 80-byte unformatted dialog message is to be received. If a shorter message is sent by mistake, a prompt for the input to be repeated is to be issued.

```

      .
      .
      .
      KCOP = 'MGET'
      KCLA = 80
      KCFN = ' '
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL MGETRC
      IF ( KCRLM .NE. KCLA ) CALL REPEAT          1)
      .
      .
      .

```

- 1) In the 'REPEAT' routine, a prompt for the input to be repeated is returned.
2. The "FORM15" format was requested from a display terminal. The length of the unprotected data is 500 characters in various format fields. This format is to be received in the program.

```

      .
      .
      .
      KCOP = 'MGET'
      KCLA = 500
      KCFN = '#FORM15'
      CALL KDCS ( KCSPA, FORM15 )
      IF ( KCRCCC .EQ. '05Z' ) CALL ERRFMT          1)
      IF ( KCRCCC .NE. '000' ) CALL MGETRC
      .
      .
      .

```

- 1) In the 'ERRFMT' routine, the format must be displayed again to enable you to continue work with the correct format.

3. In an ongoing service, an input may occur consisting of a short message generated with the F2 function key and an additional 10 characters of data. This input is to activate a special function. The F2 key was assigned the return code 21Z during generation.

```
      .  
      .  
      .  
      KCOP = 'MGET'  
      CALL KDCS ( KCSPA, NB )  
      IF ( KCRCCC .EQ. '21Z' ) GOTO 2000          1)  
      .  
      .  
2000  .  
      KCOP = 'MGET'          2)  
      KCLA = 10  
      KCRN = ' '  
      CALL KDCS ( KCSPA, NB )  
      IF ( KCRCCC .NE. '000' ) CALL MGETRC  
      .  
      .  
      .
```

- 1) A special function is queried.
- 2) Another MGET is required for the 10 characters.

## MPUT call

1. An 80-byte unformatted message is to be sent to the terminal.

```

      .
      KCOP = 'MPUT'
      KCOM = 'NE'
      KCLM = 80
      KCRN = ' '
      KCFN = ' '
      KCDF = 0
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL MPUTRC
      .

```

2. The final message in a service is to be sent to a format terminal. The name of the #format is "FORM15". The screen should be cleared beforehand.

```

      .
      EQUIVALENCE ( NB, FORM15 )
      .
      .
      KCOP = 'MPUT'
      KCOM = 'NE'
      KCLM = 500
      KCRN = ' '
      KCFN = '#FORM15'
      KCDF = KCREPL
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL MPUTRC
      .

```

- 1) REPLACE is performed by default when you change from one format to another. The output is made in order to prevent errors due to undefined field contents.
3. In a #format "FORM10", which according to the last input at the terminal still exists, all variable fields are to be deleted as the response.

```

      .
      KCOP = 'MPUT'
      KCOM = 'NE'
      KCLM = 0
      KCRN = ' '
      KCFN = '#FORM10'
      KCDF = KCERAS
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL MPUTRC
      .

```



## DPUT call

1. An asynchronous message of 11 characters is to be sent to a follow-up program on the 6th day of the 6th month (= 157th day of the year) at 12.00 p.m. (absolute time entry). The TAC is "DEEDAY".

```

      .
      .
      .
      KCOP  = 'DPUT'
      KCOM  = 'NE'
      KCLM  = 11
      KCRN  = 'DEEDAY '
      KCFN  = ' _ '
      KCDF  = 0
      KCMOD = 'A'
      KCDAY = '157'
      KCHOUR = '12'
      KCMIN = '00'
      KCSEC = '00'
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL DPUTRC
      .
      .
      .

```

2. An asynchronous message of 80 characters is to be sent after one hour (relative time entry) to the data display terminal 'DDT1'. The screen function "audible alarm" (BEL) is also to be triggered.

```

      .
      .
      .
      KCOP  = 'DPUT'
      KCOM  = 'NE'
      KCLM  = 80
      KCRN  = 'DDT1 '
      KCFN  = ' _ '
      KCDF  = KCALARM
      KCMOD = 'R'
      KCDAY = '000'
      KCHOUR = '01'
      KCMIN = '00'
      KCSEC = '00'
      CALL KDCS ( KCSPA, NB )
      IF ( KCRCCC .NE. '000' ) CALL DPUTRC
      .
      .
      .

```

## 3.2 Example of a complete UTM application

This sample application can be used to manage address data located in an ISAM file. This file (UTM.FOR.ADDRESSES) must be declared before the start of the UTM application with the following CREATE-FILE command:

```
FILE UTM.FOR.ADDRESSES, OPEN=INOUT , FCBTYPE=ISAM,      -
      RECFORM=F , RECSIZE=136 , KEYPOS=1, KEYLEN=16,      -
      LINK=DSET10
```

The application supplies the following functions which can be called by entering the appropriate TACs in the field provided. A format is used for input and output.

### TAC Function

- |   |         |  |
|---|---------|--|
| 1 | Display | displays an address from the file. The search criterion is the last name and the first two letters of the first name, which have to be specified in the associated fields. |
| 2 | Add     | enters a new address in the file.  |
| 3 | Modify  | modifies an address entry.   |
| 4 | Delete  | deletes an address from the file.  |

An input error produces an error message in the bottom line of the format.

The above-named digits are the transaction codes (TACs) used to control the application. Transaction code 1 calls the program unit TPREAD, transaction codes 2, 3 and 4 the program unit TPUPDAT.

The program unit OPNCLOS serves as a START and SHUT event exit and opens or closes the UTM.FOR.ADDRESSES file.

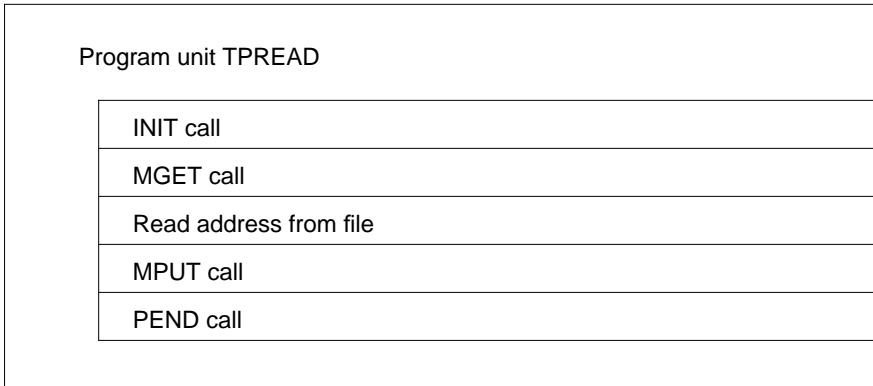
The program unit BADTAC is called automatically by UTM whenever an invalid TAC is entered.

Interaction with the user then proceeds in a strict dialog, i.e. when a TAC and the key are entered, the application responds by displaying the format containing the desired address or by outputting a success or error message in the bottom line.

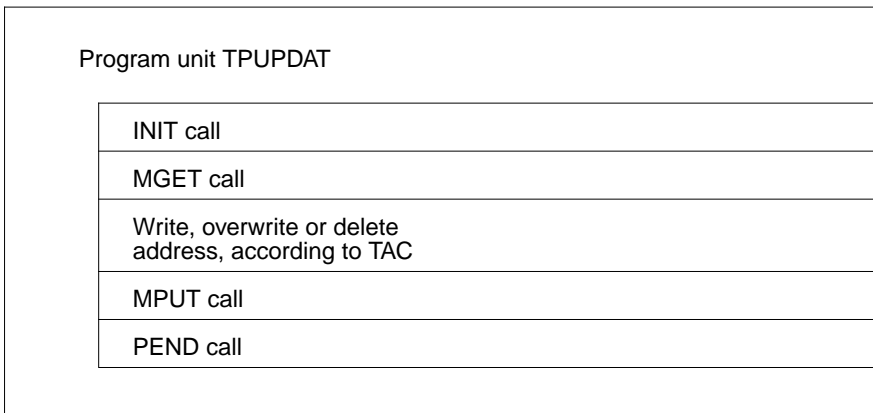
### Note

This program is intended only to show how to program with UTM. The file access operations are not backed up by the UTM transaction concept.

The following structure diagrams show the structure of the program units:



Structure diagram of the TPREAD program unit



Structure diagram of the TPUPDAT program unit

For completeness, the Fortran programs are followed here by the procedure for defining the configuration, generating and compiling the UTM main routine, and linking and starting the UTM application (test and productive). The precise meaning of the individual operands and statements can be found in the *openUTM-Handbuch „Anwendungen generieren und betreiben“*.

The figure below shows the format used for this application:

```

1  1   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80
2  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
3  |*****|
4  |               A d r e s s   M a n a g e m e n t               |
5  |*****|
6  | Please select a function: _____ |
7  |-----|
8  | Current function: nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn |
9  |-----|
10 | Last name: _____ First name: _____ |
11 | Street: _____ No: _____ |
12 | ZIP code: ##### City: _____ |
13 | Phone: _____ |
14 |-----|
15 |                               Function menu                               |
16 | 1 = Display addresses          4 = Delete addresses                    |
17 | 2 = Add new addresses          Quit with kdcoff                        |
18 | 3 = Modify addresses          |                                         |
19 |-----|
20 | nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn |
21 |-----|
22 |
23 |
24 |-----|
    1   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80

```

The #format "FORMA" used by this application

## IFG attribute list for this format

```

POSITION
LI CO FIELD NAME LENGTH ATTRIBUTES
( (*) OR (**) INDICATES DEVIATION FROM USER PROFILE VALUES )
01 001 080 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
02 023 035 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
03 001 080 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
04 007 026 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
04 033 TAC 008 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
05 001 080 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
06 007 018 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
06 025 FUNCTION 036 OUTPUT FIELD, PROTECTED, NORMAL, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
09 007 011 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
09 018 LASTNAME 014 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
09 043 012 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
09 055 FST 002 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
09 057 001 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
09 058 FSTREST 018 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
11 007 008 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
11 015 STREET 025 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
11 040 007 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
11 047 HOUSENO 010 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
13 007 010 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
13 017 ZIP 005 INPUT FIELD, NUMERIC, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: RIGHT / '0'
ALIGNMENT / FILL CHARACTER OUTPUT : RIGHT / NIL
13 043 006 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '

```

```

13 049 CITY          027 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
15 007              007 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
15 014 PHONE        018 INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : LEFT / ' '
17 001              080 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
18 034              013 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
19 007              056 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
20 007              021 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
21 007              056 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
22 001              080 TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '
23 001 MESSAGETEXT 080 OUTPUT FIELD, PROTECTED, NORMAL, ACCESSIBLE TO PROGRAM
                      ALIGNMENT / FILL CHARACTER FOR INPUT: NONE / ' '
                      ALIGNMENT / FILL CHARACTER OUTPUT : NONE / ' '

```

**Addressing aid for this format**

```

* FORMAT NAME: FORMA      USER AREA LENGTH: 00334
*
*****
*                          GLOBAL ATTRIBUTE BLOCK                          *
*****
*
* CHARACTER * 52 FORMAGLOBALS
*
*
* FORM RETURNCODE
*
* INTEGER * 4 FORMARCMAN
* RC MAIN
*
* INTEGER * 2 FORMARCCATEGO
* RC CATEGORY
*
* INTEGER * 2 FORMARCREASON
* RC REASON
*
*
* FORM INDICATORS
*
* CHARACTER * 1 FORMAFMOD
* FIELDS MOD
*
* CHARACTER * 1 FORMAFDDET
* FIELDS DET
*
* CHARACTER * 1 FORMAFDVALID
* FIELDS VALID
*
* CHARACTER * 1 FORMAUEREXRC
* USER EXIT RC
*
* CHARACTER * 1 FORMARESERV1
* RESERVED1
*
*
* INPUT IDENTIFICATION
*
* CHARACTER * 1 FORMAINPCLASS
* INPUT KEY CLASS
*
* INTEGER * 2 FORMAINPNUMB
* INPUT KEY NUMBER
*
* CHARACTER * 4 FORMARESERV2
* RESERVED2
*
*
* DEVICE CONTROLS
*
* CHARACTER * 1 FORMAINITCTL
* INIT CTL
*
* CHARACTER * 1 FORMAINITOPT
* INIT OPT
*
* CHARACTER * 1 FORMATABCTL
* TAB CTL
*
* CHARACTER * 1 FORMAFCTLOCK
* FCT LOCK
*

```

```

CHARACTER * 1 FORMAVMICTL
*
* VMI CTL
CHARACTER * 1 FORMAHMICTL
*
* HMI CTL
CHARACTER * 2 FORMARESERV3
*
* RESERVED3
*
* OUTPUT CONTROLS
CHARACTER * 1 FORMACYCLCTL
*
* CYCLE CTL
CHARACTER * 1 FORMACOPYCTL
*
* COPY CTL
CHARACTER * 1 FORMAALARMCTL
*
* ALARM CTL
CHARACTER * 1 FORMARESERV4
*
* RESERVED4
*
* FORM CONTROLS
CHARACTER * 1 FORMADISPLSEL
*
* DISPLAY SEL
CHARACTER * 1 FORMALEVELSEL
*
* LEVEL SEL
CHARACTER * 1 FORMAOUTMODE
*
* OUTPUT MODE
CHARACTER * 1 FORMACURSCTL
*
* CURSOR CTL
INTEGER * 4 FORMACURSPOS
*
* CURSOR POS
CHARACTER * 1 FORMAUSEXCTL
*
* USER EXIT CTL
CHARACTER * 1 FORMARESERV5
*
* RESERVED5
INTEGER * 2 FORMARESERV6
*
* RESERVED6
*
*
CHARACTER * 8 FORMAPKEYSET
*
* P KEY SET
*
EQUIVALENCE (FORMAGLOBALS ( 1: 4), FORMARCMAIN)
EQUIVALENCE (FORMAGLOBALS ( 5: 6), FORMARCCATEGO)
EQUIVALENCE (FORMAGLOBALS ( 7: 8), FORMARCREASON)
EQUIVALENCE (FORMAGLOBALS ( 9: 9), FORMAFLDMOD)
EQUIVALENCE (FORMAGLOBALS (10:10), FORMAFLDDET)
EQUIVALENCE (FORMAGLOBALS (11:11), FORMAFLDVALID)
EQUIVALENCE (FORMAGLOBALS (12:12), FORMAUSEREXRC)
EQUIVALENCE (FORMAGLOBALS (13:13), FORMARESERV1)
EQUIVALENCE (FORMAGLOBALS (14:14), FORMAINPCCLASS)

```



```

EQUIVALENCE (FORMAGLOBALS (15:16), FORMAINPNUMB)
EQUIVALENCE (FORMAGLOBALS (17:20), FORMARESERV2)
EQUIVALENCE (FORMAGLOBALS (21:21), FORMAINITCTL)
EQUIVALENCE (FORMAGLOBALS (22:22), FORMAINITOPT)
EQUIVALENCE (FORMAGLOBALS (23:23), FORMATABCTL)
EQUIVALENCE (FORMAGLOBALS (24:24), FORMAFCTLOCK)
EQUIVALENCE (FORMAGLOBALS (25:25), FORMAVMICTL)
EQUIVALENCE (FORMAGLOBALS (26:26), FORMAHMICTL)
EQUIVALENCE (FORMAGLOBALS (27:28), FORMARESERV3)
EQUIVALENCE (FORMAGLOBALS (29:29), FORMACYCLCTL)
EQUIVALENCE (FORMAGLOBALS (30:30), FORMACOPYCTL)
EQUIVALENCE (FORMAGLOBALS (31:31), FORMAALARMCTL)
EQUIVALENCE (FORMAGLOBALS (32:32), FORMARESERV4)
EQUIVALENCE (FORMAGLOBALS (33:33), FORMADISPLSEL)
EQUIVALENCE (FORMAGLOBALS (34:34), FORMALEVELSEL)
EQUIVALENCE (FORMAGLOBALS (35:35), FORMAOUTMODE)
EQUIVALENCE (FORMAGLOBALS (36:36), FORMACURSCNTL)

      EQUIVALENCE (FORMAGLOBALS (37:40), FORMACURSPOS)
EQUIVALENCE (FORMAGLOBALS (41:41), FORMAUSEXCTL)
EQUIVALENCE (FORMAGLOBALS (42:42), FORMARESERV5)
EQUIVALENCE (FORMAGLOBALS (43:44), FORMARESERV6)
EQUIVALENCE (FORMAGLOBALS (45:52), FORMAPKEYSET)

```

\*

\*\*\*\*\*

\*

FIELD ATTRIBUTE BLOCKS

\*

\*\*\*\*\*

\*

```

CHARACTER * 44 FORMAATTR

```

\*

\*

```

CHARACTER * 4 TACFAB

```

\*

FLD ATTRIBUTE BLOCK

```

CHARACTER * 4 TACBAT

```

\*

BASIC ATTR

```

CHARACTER * 1 TACIST

```

\*

INPUT STATE

```

CHARACTER * 1 TACISA

```

\*

INPUT STATE ACT

```

CHARACTER * 1 TACEST

```

\*

EDIT STATE

```

CHARACTER * 1 TACOCCT

```

\*

OUTPUT CTL

\*

```

EQUIVALENCE (FORMAATTR ( 1: 4), TACFAB)

```

```

EQUIVALENCE (TACFAB ( 1: 4), TACBAT)

```

```

EQUIVALENCE (TACBAT (1:1), TACIST)

```

```

EQUIVALENCE (TACBAT (2:2), TACISA)

```

```

EQUIVALENCE (TACBAT          (3:3), TACEST)
EQUIVALENCE (TACBAT          (4:4), TACOCT)
*
*
CHARACTER * 4 FUNCTIONFAB
CHARACTER * 4 FUNCTIONBAT
CHARACTER * 1 FUNCTIONIST
CHARACTER * 1 FUNCTIONISA
CHARACTER * 1 FUNCTIONEST
CHARACTER * 1 FUNCTIONOCT
*
EQUIVALENCE (FORMAATTR ( 5: 8), FUNCTIONFAB)
EQUIVALENCE (FUNCTIONFAB ( 1: 4), FUNCTIONBAT)
EQUIVALENCE (FUNCTIONBAT (1:1), FUNCTIONIST)
EQUIVALENCE (FUNCTIONBAT (2:2), FUNCTIONISA)
EQUIVALENCE (FUNCTIONBAT (3:3), FUNCTIONEST)
EQUIVALENCE (FUNCTIONBAT (4:4), FUNCTIONOCT)
*
*
CHARACTER * 4 LASTNAMEFAB
CHARACTER * 4 LASTNAMEBAT
CHARACTER * 1 LASTNAMEIST
CHARACTER * 1 LASTNAMEISA
CHARACTER * 1 LASTNAMEEST
CHARACTER * 1 LASTNAMEOCT
*
EQUIVALENCE (FORMAATTR ( 9: 12), LASTNAMEFAB)
EQUIVALENCE (LASTNAMEFAB ( 1: 4), LASTNAMEBAT)
EQUIVALENCE (LASTNAMEBAT (1:1), LASTNAMEIST)
EQUIVALENCE (LASTNAMEBAT (2:2), LASTNAMEISA)
EQUIVALENCE (LASTNAMEBAT (3:3), LASTNAMEEST)
EQUIVALENCE (LASTNAMEBAT (4:4), LASTNAMEOCT)
*
*
CHARACTER * 4 FSTFAB
CHARACTER * 4 FSTBAT
CHARACTER * 1 FSTIST
CHARACTER * 1 FSTISA
CHARACTER * 1 FSTEST
CHARACTER * 1 FSTOCT
*
EQUIVALENCE (FORMAATTR ( 13: 16), FSTFAB)
EQUIVALENCE (FSTFAB ( 1: 4), FSTBAT)
EQUIVALENCE (FSTBAT (1:1), FSTIST)
EQUIVALENCE (FSTBAT (2:2), FSTISA)
EQUIVALENCE (FSTBAT (3:3), FSTEST)
EQUIVALENCE (FSTBAT (4:4), FSTOCT)
*

```

```

*
CHARACTER * 4 FSTRESTFAB
CHARACTER * 4 FSTRESTBAT
CHARACTER * 1 FSTRESTIST
CHARACTER * 1 FSTRESTISA
CHARACTER * 1 FSTRESTEST
CHARACTER * 1 FSTRESTOCT

*
EQUIVALENCE (FORMAATTR ( 17: 20), FSTRESTFAB)
EQUIVALENCE (FSTRESTFAB ( 1: 4), FSTRESTBAT)
EQUIVALENCE (FSTRESTBAT (1:1), FSTRESTIST)
EQUIVALENCE (FSTRESTBAT (2:2), FSTRESTISA)
EQUIVALENCE (FSTRESTBAT (3:3), FSTRESTEST)
EQUIVALENCE (FSTRESTBAT (4:4), FSTRESTOCT)

*
*
CHARACTER * 4 STREETFAB
CHARACTER * 4 STREETBAT
CHARACTER * 1 STREETIST
CHARACTER * 1 STREETISA
CHARACTER * 1 STREETEST
CHARACTER * 1 STREETOCT

*
EQUIVALENCE (FORMAATTR ( 21: 24), STREETFAB)
EQUIVALENCE (STREETFAB ( 1: 4), STREETBAT)
EQUIVALENCE (STREETBAT (1:1), STREETIST)
EQUIVALENCE (STREETBAT (2:2), STREETISA)
EQUIVALENCE (STREETBAT (3:3), STREETEST)
EQUIVALENCE (STREETBAT (4:4), STREETOCT)

*
*
CHARACTER * 4 NOFAB
CHARACTER * 4 NOBAT
CHARACTER * 1 NOIST
CHARACTER * 1 NOISA
CHARACTER * 1 NOEST
CHARACTER * 1 NOOCT

*
EQUIVALENCE (FORMAATTR ( 25: 28), NOFAB)
EQUIVALENCE (NOFAB ( 1: 4), NOBAT)
EQUIVALENCE (NOBAT (1:1), NOIST)
EQUIVALENCE (NOBAT (2:2), NOISA)
EQUIVALENCE (NOBAT (3:3), NOEST)
EQUIVALENCE (NOBAT (4:4), NOOCT)

*
CHARACTER * 4 ZIPFAB
CHARACTER * 4 ZIPBAT
CHARACTER * 1 ZIPIST

```

```

CHARACTER * 1 ZIPISA
CHARACTER * 1 ZIPEST
CHARACTER * 1 ZIPOCT
*
EQUIVALENCE (FORMAATTR ( 29: 32), ZIPFAB)
EQUIVALENCE (ZIPFAB ( 1: 4), ZIPBAT)
EQUIVALENCE (ZIPBAT (1:1), ZIPIST)
EQUIVALENCE (ZIPBAT (2:2), ZIPISA)
EQUIVALENCE (ZIPBAT (3:3), ZIPEST)
EQUIVALENCE (ZIPBAT (4:4), ZIPOCT)
*
*
CHARACTER * 4 CITYFAB
CHARACTER * 4 CITYBAT
CHARACTER * 1 CITYIST
CHARACTER * 1 CITYISA
CHARACTER * 1 CITYEST
CHARACTER * 1 CITYOCT
*
EQUIVALENCE (FORMAATTR ( 33: 36), CITYFAB)
EQUIVALENCE (CITYFAB ( 1: 4), CITYBAT)
EQUIVALENCE (CITYBAT (1:1), CITYIST)
EQUIVALENCE (CITYBAT (2:2), CITYISA)
EQUIVALENCE (CITYBAT (3:3), CITYEST)
EQUIVALENCE (CITYBAT (4:4), CITYOCT)
*
*
CHARACTER * 4 PHONEFAB
CHARACTER * 4 PHONEBAT
CHARACTER * 1 PHONEIST
CHARACTER * 1 PHONEISA
CHARACTER * 1 PHONEEST
CHARACTER * 1 PHONEOCT
*
EQUIVALENCE (FORMAATTR ( 37: 40), PHONEFAB)
EQUIVALENCE (PHONEFAB ( 1: 4), PHONEBAT)
EQUIVALENCE (PHONEBAT (1:1), PHONEIST)
EQUIVALENCE (PHONEBAT (2:2), PHONEISA)
EQUIVALENCE (PHONEBAT (3:3), PHONEEST)
EQUIVALENCE (PHONEBAT (4:4), PHONEOCT)
*
*
CHARACTER * 4 MESSEGETEXTFAB
CHARACTER * 4 MESSEGETEXTBAT
CHARACTER * 1 MESSEGETEXTIST
CHARACTER * 1 MESSEGETEXTISA
CHARACTER * 1 MESSEGETEXTTEST
CHARACTER * 1 MESSEGETEXTTOCT

```

```

*
EQUIVALENCE (FORMAATTR ( 41: 44), MESSAGETEXTFAB)
EQUIVALENCE (MESSAGETEXTFAB ( 1: 4), MESSAGETEXTBAT)
EQUIVALENCE (MESSAGETEXTBAT (1:1), MESSAGETEXTIST)
EQUIVALENCE (MESSAGETEXTBAT (2:2), MESSAGETEXTISA)
EQUIVALENCE (MESSAGETEXTBAT (3:3), MESSAGETEXTTEST)
EQUIVALENCE (MESSAGETEXTBAT (4:4), MESSAGETEXTTOCT)
*
*
*****
*
FIELD DATA PART
*****
*
CHARACTER * 243 FORMADATA
*
CHARACTER * 8 TAC
CHARACTER * 36 FUNCTION
CHARACTER * 14 LASTNAME
CHARACTER * 2 FST
CHARACTER * 18 FSTREST
CHARACTER * 25 STREET
CHARACTER * 10 NO
CHARACTER * 5 ZIP
CHARACTER * 27 CITY
CHARACTER * 18 PHONE
CHARACTER * 80 MESSAGETEXT
*
*
EQUIVALENCE (FORMADATA ( 1: 8), TAC)
EQUIVALENCE (FORMADATA ( 9: 44), FUNCTION)
EQUIVALENCE (FORMADATA ( 45: 58), LASTNAME)
EQUIVALENCE (FORMADATA ( 59: 60), FST)
EQUIVALENCE (FORMADATA ( 61: 78), FSTREST)
EQUIVALENCE (FORMADATA ( 79: 103), STREET)
EQUIVALENCE (FORMADATA ( 104: 113), NO)
EQUIVALENCE (FORMADATA ( 114: 118), ZIP)
EQUIVALENCE (FORMADATA ( 119: 145), CITY)
EQUIVALENCE (FORMADATA ( 146: 163), PHONE)
EQUIVALENCE (FORMADATA ( 164: 243), MESSAGETEXT)
*
*
CHARACTER * 339 FORMA
EQUIVALENCE (FORMA ( 1: 52), FORMAGLOBALS)
EQUIVALENCE (FORMA ( 53: 96), FORMAATTR)
EQUIVALENCE (FORMA ( 97: 339), FORMADATA)

```

## Program unit TPREAD

```

      SUBROUTINE TPREAD
*
*****
*
* The program reads a record from the file
*
* /FILE UTM.FOR.ADDRESSES,FCBTYPE=ISAM,RECFORM=F,RECSIZE=136,
*           LINK=DSET10,KEYLEN=16,KEYPOS=1,OPEN=INOUT
*
*****
*----- common KDCKB -----
*
      %EXPAND OFF
      %INCLUDE KCCAF
*
*                               communication area
      CHARACTER CAPA * 128
      COMMON /KDCKB/ CAPA
*
*----- common KDCSPAB -----
*                               ca - program area
*
      %INCLUDE KCPAF
*
*                               parameter area
      CHARACTER NB * 360
*
*                               message area:
      COMMON /KDCSPAB/ NB
*
*
*
      %INCLUDE FORMA
*
      CHARACTER  ADDRESSRECORD * 120
      CHARACTER  KEY           * 16
      CHARACTER  FSTNAME       * 20
*
      EQUIVALENCE ( NB , FORMA )
      EQUIVALENCE ( LASTNAME , KEY , ADDRESSRECORD )
      EQUIVALENCE ( FSTNAME , FST )
*
*----- end of commons -----*
      INTEGER  RETCODE * 4
*
      CHARACTER  ERRORTXT * 80
*
      CHARACTER  TEXT1 * 34 /'**** E R R O R **** Program unit: '/
      CHARACTER  FTPR * 8
      CHARACTER  TEXT2 * 17 /' Operation code: '/

```

```

CHARACTER  FOP  * 4
CHARACTER  TEXT3 * 13 // 'Return code: '/
CHARACTER  FRC  * 3
EQUIVALENCE ( ERRORTXT(1:) , TEXT1 )
EQUIVALENCE ( ERRORTXT(35:) , FPCR )
EQUIVALENCE ( ERRORTXT(43:) , TEXT2 )
EQUIVALENCE ( ERRORTXT(60:) , FOP )
EQUIVALENCE ( ERRORTXT(64:) , TEXT3 )
EQUIVALENCE ( ERRORTXT(77:) , FRC )

*
*----- end of declaration -----
*
      KCOP      = 'INIT'
      KCLCAPA = 128
      KCLSPA   = 512
      CALL KDCS ( KCSPA )

*
      IF ( KCRCCC .NE. '000' ) THEN
                FOP = 'INIT'
                GOTO 8000

      END IF

*
*----- end of INIT -----
*
      KCOP      = 'MGET'
      KCLA      = 360
      KCFN      = '#FORMA'
      CALL KDCS ( KCSPA , NB )

*
      IF ( KCRCCC .EQ. '05Z' ) THEN
                FORMADATA = ' '
                GOTO 3000

      END IF

*
      IF ( KCRCCC .NE. '000' ) THEN
                FOP = 'MGET'
                GOTO 8000

      END IF

*
*----- end of MGET -----*
      FUNCTION = ' '
      FUNCTION = '*** Display addresses ***'
      READ ( UNIT=10, FMT=200, REC = KEY ) ADDRESSRECORD
200  FORMAT ( A )
*
*
*
*
*-----
      read a record from
      UTM.FOR.ADDRESSES
*-----

```

```
*
3000 KCOP    = 'MPUT'
      KCOM    = 'NE'
      KCLM    = 360
      KCRN    = ' '
      KCFN    = '#FORMA'
      MESSAGETEXT = ' '
      CALL KDCC ( KCSPA , NB )
*
      IF ( KCRCCC .NE. '000' ) THEN
                                FOP = 'MPUT'
                                GOTO 8000
      END IF
*
*----- end of MPUT -----
*
      KCOP    = 'PEND'
      KCOM    = 'FI'
      CALL KDCC ( KCSPA )
*
*----- end of PEND -----
*
*                               error handling
*
8000 FTPT    = 'TPREAD'
      FRC     = KCRCCC
      NB      = ERRORTXT
      KCOP    = 'MPUT'
      KCOM    = 'NE'
      KCLM    = 80
      KCRN    = ' '
      KCFN    = ' '
      KCDF    = 0
      CALL KDCC ( KCSPA , NB )
*
      KCOP    = 'PEND'
      KCOM    = 'ER'
      CALL KDCC ( KCSPA )
*
      END
```



## Program unit TPUPDAT

```

      SUBROUTINE TPUPDAT
*
*****
*
* The program writes, overwrites or deletes a record in
* the file :
*
* /FILE UTM.FOR.ADDRESSES,FCBTYPE=ISAM,RECFORM=F,RECSIZE=136,
* LINK=DSET10,KEYLEN=16,KEYPOS=1,OPEN=INOUT
*
*****
*----- common KDCKB -----
*
      %EXPAND OFF
      %INCLUDE KCCAF
*
*                               communication area
      CHARACTER CAPA * 128
      COMMON /KDCKB/ CAPA
*
*                               ca - program area
*----- common KDCSPAB -----
*
      %INCLUDE KCPAF
*
*                               parameter area
      CHARACTER NB * 360
*
*                               message area:
*
      COMMON /KDCSPAB/ NB
*
*
*
      %INCLUDE FORMA
*
*
*
      CHARACTER KEY * 16
      CHARACTER FSTNAME * 20
      CHARACTER ADDRESSRECORD * 120
*
      EQUIVALENCE ( NB , FORMA )
      EQUIVALENCE ( LASTNAME , KEY , ADDRESSRECORD )
      EQUIVALENCE ( FSTNAME , FST )
*
*----- end of commons -----
*
      INTEGER RETCODE * 4
*

```

```

*
CHARACTER ERRORETEXT * 80
*
CHARACTER TEXT1 * 34 /'**** E R R O R **** Program unit: '/
CHARACTER TEXT2 * 17 /' Operation code: '/
CHARACTER FOP * 4
CHARACTER TEXT3 * 13 /'Return code: '/
CHARACTER FRC * 3
*
EQUIVALENCE ( ERRORETEXT(1:) , TEXT1 )
EQUIVALENCE ( ERRORETEXT(35:) , FOP )
EQUIVALENCE ( ERRORETEXT(43:) , TEXT2 )
EQUIVALENCE ( ERRORETEXT(60:) , FOP )
EQUIVALENCE ( ERRORETEXT(64:) , TEXT3 )
EQUIVALENCE ( ERRORETEXT(77:) , FRC )
*
*----- end of declaration -----
*
KCOP = 'INIT'
KCLCAPA = 128
KCLSPA = 512
CALL KDCS ( KCSPA )
*
IF ( KCRCCC .NE. '000' ) THEN
    FOP = 'INIT'
    GOTO 8000
END IF
*----- end of INIT -----
*
KCOP = 'MGET'
KCLA = 360
KCFN = '#FORMA '
CALL KDCS ( KCSPA , NB )
*
IF ( KCRCCC .EQ. '05Z' ) THEN
    FORMADATA = ' '
    GOTO 3000
END IF
*
IF ( KCRCCC .NE. '000' ) THEN
    FOP = 'MGET'
    GOTO 8000
END IF
*----- end of MGET -----
*
FUNCTION = ' '
IF ( KCCVTAC .EQ. '2' ) THEN
*

```

```

        FUNCTION = '*** Add new addresses ***'
        WRITE ( UNIT=10, FMT=200, REC=KEY ) ADDRESSRECORD
*
    ELSE IF ( KCCVTAC .EQ. '3' ) THEN
*
        FUNCTION = '*** Overwrite addresses ***'
        WRITE ( UNIT=10, FMT=200, REC=KEY ) ADDRESSRECORD
*
    ELSE IF ( KCCVTAC .EQ. '4' ) THEN
*
        FUNCTION = '*** Delete addresses ***'
        CALL ELIMCHR ( 10, KEY, RETCODE )
*
    ELSE
*
    END IF
*
200  FORMAT ( A )
*
*-----
*
3000  KCOP      = 'MPUT'
      KCOM      = 'NE'
      KCLM      = 360
      KCRN      = ' '
      KCFN      = '#FORMA '
      MESSTEXT  = ' '
      CALL KDCS ( KCSPA , NB )
*
      IF ( KCRCCC .NE. '000' ) THEN
                                FOP = 'MPUT'
                                GOTO 8000
      END IF
*-----
*----- end of MPUT -----
*
      KCOP      = 'PEND'
      KCOM      = 'FI'
      CALL KDCS ( KCSPA )
*-----
*----- end of PEND -----
*
*-----
*----- error handling -----
*
8000  FTPR      = 'TPUPDAT'
      FRC       = KCRCCC
      NB        = ERRORTXT
      KCOP      = 'MPUT'
      KCOM      = 'NE'
      KCLM      = 80

```

```
KCRN   = ' '
KCFN   = ' '
KCDF   = 0
CALL KDCS ( KCSPA , NB )
*
KCOPI  = 'PEND'
KCOM   = 'ER'
CALL KDCS ( KCSPA )
*
*
END
```

**Program unit BADTAC**

```

      SUBROUTINE BADTAC
*
*****
*
*   The program handles the case of an invalid TAC.
*
*****
*----- common KDCKB -----
*
      %EXPAND OFF
      %INCLUDE KCCAF
*
*                               communication area
      CHARACTER CAPA * 128
      COMMON /KDCKB/ CAPA
*
*                               ca - program area
*----- common KDCSPAB -----
*
      %INCLUDE KCPAF
*
*                               parameter area
      CHARACTER NB * 360
*
*                               message area
      COMMON /KDCSPAB/ NB
*
      %INCLUDE FORMA
*
      EQUIVALENCE ( NB , FORMA )
*
*----- end of commons -----
*
      INTEGER   RETCODE * 4
*
      CHARACTER ERRORTXT * 80
*
      CHARACTER TEXT1 * 35 /'**** E R R O R **** Program unit : '/
      CHARACTER FTPR * 8
      CHARACTER TEXT2 * 17 /' Operation code: '/
      CHARACTER FOP * 4
      CHARACTER TEXT3 * 13 /'Return code: '/
      CHARACTER FRC * 3
*
      EQUIVALENCE ( ERRORTXT(1:) , TEXT1 )
      EQUIVALENCE ( ERRORTXT(36:) , FTPR )
      EQUIVALENCE ( ERRORTXT(44:) , TEXT2 )
      EQUIVALENCE ( ERRORTXT(61:) , FOP )
      EQUIVALENCE ( ERRORTXT(65:) , TEXT3 )

```

```

      EQUIVALENCE ( ERRORTXT(78:) , FRC      )
*
      CHARACTER ERRORTXT2 * 80 /'*** Invalid function - please repeat
      input ***'/
*
*----- end of declaration -----
*
      NB      = ' _ '
      KCOP    = 'INIT'
      KCLCAPA = 128
      KCLSPA  = 360
      CALL KDACS ( KCSPA )
*
      IF ( KCRCCC .NE. '000' ) THEN
                                FOP = 'INIT'
                                GOTO 8000
      END IF
*
*----- end of INIT -----
*
      KCOP    = 'MGET'
      KCLA    = 360
      KCFN    = '#FORMA '
      CALL KDACS ( KCSPA , NB )
*
      IF ( KCRCCC .EQ. '05Z' ) THEN
                                FORMADATA = ' '
                                GOTO 3000
      END IF
*
      IF ( KCRCCC .NE. '000' ) THEN
                                FOP = 'MGET'
                                GOTO 8000
      END IF
*
*----- end of MGET -----
*
*
*
3000  MESSAGETEXT = ERRORTXT2
      TAC = ' '
      KCOP    = 'MPUT'
      KCOM    = 'NE'
      KCLM    = 360
      KCRN    = ' '
      KCFN    = '#FORMA '
      CALL KDACS ( KCSPA , NB )
*
      IF ( KCRCCC .NE. '000' ) THEN

```

```

                                                    FOP = 'MPUT'
                                                    GOTO 8000

      END IF

*
*----- end of MPUT -----
*
      MESSAGETEXT = ' '
      KCOP      = 'PEND'
      KCOM      = 'FI'
      CALL KDCS ( KCSPA )

*
*----- end of PEND -----
*
*                                     error handling
*
8000  FTPR      = 'BADTAC'
      FRC       = KCRCC
      NB        = ERRORTXT
      KCOP      = 'MPUT'
      KCOM      = 'NE'
      KCLM      = 80
      KCRN      = ' '
      KCFN      = ' '
      KCDF      = 0
      CALL KDCS ( KCSPA , NB )

*
      KCOP      = 'PEND'
      KCOM      = 'ER'
      CALL KDCS ( KCSPA )

*
      END
```

**Program unit OPNCLOS**

```

      SUBROUTINE OPNCLOS
*
*****
*
* This program it is used for event exits START and SHUT.
* It opens and closes the file:
* /FILE UTM.FOR.ADDRESSES, FCBTYPE=ISAM, RECFORM=F, RECSIZE=120,
* LINK=DSET10, KEYLEN=16, KEYPOS=1, OPEN=INPUT
*
*****
*
*
      %EXPAND OFF
      %INCLUDE KCCAF
*
*----- end of declaration -----
*
*
*** Start / Shut Routine ***
*
      IF ( KCCVTAC .EQ. 'STARTUP' ) THEN
+         OPEN ( UNIT=10,
+             ACCESS='DIRECT,C',RECL=120 )
      END IF
*
      IF ( KCCVTAC .EQ. 'SHUTDOWN' ) THEN
          CLOSE ( UNIT=10 )
      END IF
*
      END

```



## Procedure UTM.BSP.P for defining the configuration, generating the UTM main routine and linking and starting the UTM application

```

/PROC N,(&FKT,&TEST=YES,&LD)
/,SUBDTA=&
/  SYSFILE SYSDTA=(SYSCMD)
/  TCHNG OFLOW=NO
/  SETSW ON=(4)
/WR-TEXT '
/
/
/                               Define the configuration .....-
/..... (D)                               Generate the linkage pro-
/gram KDCROOT..... (A)                               Link the UTM -
/FOR1 application ..... (B)                               Start-
/the UTM FOR1 APPLICATION (TEST) ..... (S)
/  Header for KDCLOAD ..... (K)
/                               Start the UTM FOR1 application (productive) .. (P)
/                               Terminate procedure .....-
/ (E)
/                               Please enter desired func-
/tion
/
/  SKIP .&FKT
/.D EXEC $TSOS.SYSPRG.UTM.040.KDCDEF
REM *****
REM *****
REM ***      D E F      STATEMENTS      ***
REM *****
REM *****
OPTION GEN=ALL,ROOTSRC=UTM.BSP.KDCROOT,TEST=&TEST
ROOT UTMFBSP
REM *****
REM ***      FORMATTING SYSTEM      ***
REM *****
FORMSYS ENTRY=KDCFHS,LIB=$MFHSROUT,TYPE=FHS
REM *****
REM ***      GLOBAL AND MAXIMUM VALUES      ***
REM *****
MAX APPLINAME=UTMFBSP
MAX KDCFILE=(FBSP.KDCFILE,S)
MAX CONRTIME=5,LOGACKWAIT=60
REM *****
REM ***      PROGRAM UNIT STATEMENTS      ***
REM *****
PROGRAM KCADM,COMP=ILCS
PROGRAM TPREAD,COMP=FOR1
PROGRAM TPUPDAT,COMP=FOR1
PROGRAM OPNCLOS,COMP=FOR1

```

```

PROGRAM BADTAC,COMP=FOR1
REM *****
REM ***          EXIT STATEMENTS          ***
REM *****
EXIT PROGRAM=OPNCLOS,USAGE=START
EXIT PROGRAM=OPNCLOS,USAGE=SHUT
REM *****
REM ***    TAC STATEMENTS                    ***
REM *****
DEFAULT TAC ADMIN=Y,PROGRAM=KDCADM
TAC KDCTAC
TAC KDCLTERM
TAC KDCPTERM
TAC KDCSWTCH
TAC KDCUSER
TAC KDCSEND
TAC KDCAPPL
TAC KDCDIAG
TAC KDCLOG
TAC KDCINF
TAC KDCHELP
TAC KDCSHUT
DEFAULT TAC TYPE=A,ADMIN=Y,PROGRAM=KDCADM
TAC KDCTACA
TAC KDCLTRMA
TAC KDCPTRMA
TAC KDCSWCHA
TAC KDCUSERA
TAC KDCSENA
TAC KDCAPPLA
TAC KDCDIAGA
TAC KDCLOGA
TAC KDCINF A
TAC KDCHELPA
TAC KDCSHUTA
DEFAULT TAC TYPE=D,ADMIN=N,PROGRAM=(STD)
TAC KDCBADTC,PROGRAM=BADTAC
TAC 1,PROGRAM=TPREAD
TAC 2,PROGRAM=TPUPDAT
TAC 3,PROGRAM=TPUPDAT
TAC 4,PROGRAM=TPUPDAT
REM *****
REM *****          USER STATEMENTS          *****
REM *****
USER  SUSIE ,KSET=BUNCH1 ,STATUS=ADMIN ,PASS=( EMPLOY , DARK )
USER  GERTRUDE ,KSET=BUNCH2 ,STATUS=ON ,PASS=( EMPLOY , DARK )
USER  BARBARA ,KSET=BUNCH3 ,STATUS=ON ,PASS=( EMPLOY , DARK )
REM *****

```

```

REM *****
                PTERM/LTERM STATEMENTS                *****
REM *****
DEFAULT PTERM PRONAM=PT01,PTYPE=T9750
PTERM DDT01,LTERM=UTMDT1
PTERM DDT02,LTERM=UTMDT2
PTERM DDT03,LTERM=UTMDT3
DEFAULT PTERM PRONAM=TC01,PTYPE=T9022,USAGE=0
PTERM G01,LTERM=PRINTER,CONNECT=A
LTERM UTMDT1,KSET=BUNCH1
LTERM UTMDT2,LOCK=4,KSET=BUNCH1
LTERM UTMDT3,LOCK=5,KSET=BUNCH1
LTERM PRINTER,USAGE=0
REM *****
                KSET STATEMENTS                        *****
REM *****
KSET  BUNCH1,KEYS=(1,2,3,4,5)
KSET  BUNCH2,KEYS=(1,2,4)
KSET  BUNCH3,KEYS=(1)
REM *****
                TLS STATEMENTS                        *****
REM *****
TLS   TLSA
TLS   TLSB
END
/SKIP .CALL
/STEP
/WR-TEXT ' >> Definition of configuration incorrect << '
/SKIP .CALL
/.A FILE $TSOS.SYSMAC.FHS.060,LINK=ALTLIB
/   FILE $TSOS.SYSLIB.UTM.040.ASS,LINK=ALTLIB2
/   ER*
/   EXEC $ASSEMB
*COMOPT OLDLIB,OLDLIB2,OLDLIB3,SOURCE=UTM.BSP.KDCROOT
*COMOPT MODULE=UTM.KDCROOT.PLAMLIB
*END HALT
/   ER UTM.BSP.KDCROOT
/   SKIP .CALL
/   STEP
/   WR-TEXT ' >> Generation of linkage program incorrect << '
/   SKIP .CALL
/.B EXEC $TSOSLNK
PROG BSP,FILENAM=UTM.BSP.L,LOADPT=&LD,SYM=ALL
INCLUDE UTMFBSP,UTM.KDCROOT.PLAMLIB
INCLUDE (TPREAD,TPUPDAT),UTM.BSP.PLAMLIB
INCLUDE (OPNCLOS,BADTAC),UTM.BSP.PLAMLIB
RESOLVE , $TSOS.SYSLNK.FOR1-LZS.022.FPOOL
RESOLVE , $TSOS.KDC.DLL.OML
RESOLVE , $TSOS.SYSLNK.UTM.040.SPLRTS

```

```

RESOLVE , $TSOS.SYSLNK.FOR1-LZS.022.DYNAMIC
RESOLVE , $TSOS.SYSLNK.CRTE
RESOLVE , $TSOS.SYSLNK.UTM.040
NCAL
END
/  SKIP .CALL
/  STEP
/  WR-TEXT ' >> Linking of FOR1-UTM application incorrect << '
/  SKIP .CALL
/.K WR-TEXT '          Header for KDCLOAD'
/  WR-TEXT '          Create FGG'
/  ER FBSP.KDCFILE.PROG
/  STEP
/  CAT FBSP.KDCFILE.PROG.GEN=3
/  FILE FBSP.KDCFILE.PROG(*1)
/  EXEC UTM.BSP.L
PREPARE FILEBASE=FBSP.KDCFILE
END
/  SKIP .CALL
/.P WR-TEXT '          Start the UTM FOR1 application'
/  CAT FBSP.KDCFILE.PROG,BASE=0,STATE=U
/  ENTER UTM.BSP.E,PRIORITY=(,EXPRESS),TIME=NTL
/  SKIP .CALL
/.S WR-TEXT '          Please enter PTERM,PRONAME'
/  SYSFILE TASKLIB=$TSOS.SYSLNK.FOR1-LZS.022.DYNAMIC
/  FILE FBSP.KDCFILE.SYSLOG,LINK=SYSLOG,SHARUPD=YES
/  FILE FBSP.KDCFILE.KDCA,LINK=KDCFILE,SHARUPD=NO
/  FILE UTM.FOR.ADDRESSES,LINK=DSET10,FCBTYPE=ISAM,KEYLEN=16,
/  KEYPOS=1,RECFORM=F,RECSIZE=136,OPEN=INOUT
/  EXEC UTM.BSP.L,SYM=ALL
.UTM START FILEBASE=FBSP.KDCFILE
      START TASKS=1
.FHS MAPLIB=UTM.BSP.IFGLIB
      END
/  STEP
/  ENDP
/.CALL DO UTM.BSP.P
/  STEP
/.E OPT MSG=F
/  TCHNG OFLOW=ACK
/  SETSW OFF=(4)
/  ENDP

```

**Starting up the UTM Fortran application (productive) with the KDCLOAD****ENTER job UTM.BSP.E**

```
/.UTMFBSB LOGON
/      SYSFILE SYSOUT=UTM.BSP.SYSOUT
/      SYSFILE SYSDDTA=(SYSCMD)
/      SYSFILE TASKLIB=$TSOS.SYSLNK.FOR1-LZS.022.DYNAMIC
/      FILE FBSP.KDCFILE.SYSLOG, LINK=SYSLOG, SHARUPD=YES
/.REPEAT FILE FBSP.KDCFILE.PROG(+0), LINK=KDCLOAD
/      FILE UTM.FOR.ADDRESSES, LINK=DSET10, FCBTYP=ISAM, KEYLEN=16, -
/      KEYPOS=1, RECFORM=F, RECSIZE=136, OPEN=INOUT
/      EXEC (KDCLOAD, $TSOS.SYSLNK.UTM.040), PROG=MOD=ANY
.UTM START FILEBASE=:A:$FORDIAG.FBSP.KDCFILE
.UTM START STARTNAME=:A:$FORDIAG.UTM.BSP.E
.UTM START TASKS=1
.UTM START ASYNTASKS=0
.UTM START TESTMODE=ON
.FHS MAPLIB=UTM.BSP.IFGLIB
.UTM END
/      SKIP .REPEAT
/      STEP
/      SYSFILE SYSOUT=(PRIMARY)
/      LOGOFF NOSPOOL
```

### Modified procedure UTM.BSP.P for loading the shareable code sections of the program units into a common memory pool

```

/PROC N,(&FKT,&TEST=YES,&LD),SUBDTA=&
/  SYSFILE SYSDTA=(SYSCMD)
/  TCHNG OFLOW=NO
/WR-TEXT '
/
/                               Compile FOR1 program unit.....-
/ (F)                               Define the configuration.....-
/..... (D)                               Generate the linkage pro-
/gram KDCROOT..... (A)                               Link the UTM -
/FOR1 application..... (B)                               Start-
/the UTM FOR1 application (Test) ..... (S)            -
/  Header for KDCLOAD ..... (K)                        -
/                               Start the UTM FOR1 application (productive) .. (P) -
/                               Terminate procedure.....-
/ (E)                               -
/                               Please enter desired func-
/tion                               -
/
/  SKIP .&FKT
/.F ER *
/  SETSW ON=(4)
/  EXEC $FOR1
COMOPT SRC=UTM.BSP.SRC,XS=YES,OBJ=(SHARE)
COMOPT MODULE-LIBRARY=UTM.TP.PLAMLIB.NOSHARE
COMOPT SHARE-LIBRARY=UTM.TP.PLAMLIB.SHARE
COMOPT LF=UTM.SHARE.LF(SRC,XR,OP)
COMOPT INCLUDE--($TSOS.SYSLIB.UTM.040.FOR,UTM.BSP.IFGLIB)
END
/  SKIP .CALL
/  STEP
/  WR-TEXT ' >> FOR1 compilation incorrect << '
/  SKIP .CALL
/.D EXEC $TSOS.SYSPRG.UTM.040.KDCDEF
REM *****
REM ***      D E F      STATEMENTS      ***
REM *****
REM *****
OPTION GEN=ALL,ROOTSRC=UTM.SRC.ROOT,TEST=&TEST
OPTION SHARETAB=UTM.SRC.SHARETAB
ROOT UTMFSHR
REM *****
REM ***  FORMATTING SYSTEM  ***
REM *****
FORMSYS ENTRY=KDCFHS,LIB=$MFHSROUT,TYPE=FHS
REM *****
REM ***  GLOBAL AND MAXIMUM VALUES  ***

```

```

REM *****
MAX APPLNAME=UTMFSHR
MAX KDCFILE=(FSHR.KDCFILE,S)
MAX CONRTIME=5,LOGACKWAIT=60
REM *****
REM ***   COMMON MEMORY POOL   ***
REM *****
MPOOL MPOOL1,SIZE=2,SHARETAB=SHARETAB,LIB=UTM.TP.PLAMLIB.SHARE
REM *****
REM ***   PROGRAM UNIT STATEMENTS   ***
REM *****
PROGRAM KDCADM,COMP=ILCS,LOAD=STATIC
DEFAULT PROGRAM LOAD=STATIC,LIB=UTM.TP.PLAMLIB.NOSHARE
PROGRAM TPREAD,COMP=FOR1
PROGRAM TPUPDAT,COMP=FOR1
PROGRAM OPNCLOS,COMP=FOR1
PROGRAM BADTAC,COMP=FOR1
REM *
REM *   Shareable module "SHRMOD" contains the adapter module
REM *   "ADAPTN" and the code section of the program units.
REM *
MODULE SHRMOD,LOAD=(POOL,MPOOL1),LIB=UTM.TP.PLAMLIB.SHARE
REM *
REM *   Format "FORMA" is also loaded into the memory pool.
REM *
MODULE FORMA,LOAD=(POOL,MPOOL1),LIB=UTM.BSP.IFGLIB
REM *
REM *   The MPOOL and MODULE statements are not required when loading
REM *   the shareable modules into the class 4 memory.
REM *
REM *****
REM ***           EXIT DEFINITIONS           ***
REM *****
EXIT PROGRAM=OPNCLOS,USAGE=START
EXIT PROGRAM=OPNCLOS,USAGE=SHUT
REM *****
REM ***           TAC DEFINITIONS           ***
REM *****
DEFAULT TAC ADMIN=Y,PROGRAM=KDCADM
TAC KDCTAC
TAC KDCLTERM
TAC KDCPTERM
TAC KDCSWTCH
TAC KDCUSER
TAC KDCSEND
TAC KDCAPPL
TAC KDCDIAG
TAC KDCLOG

```

```

TAC KDCINF
TAC KDCHELP
TAC KDCSHUT
DEFAULT TAC TYPE=A,ADMIN=Y,PROGRAM=KDCADM
TAC KDCTACA
TAC KDCLTRMA
TAC KDCPTRMA
TAC KDCSWCHA
TAC KDCUSERA
TAC KDCSENA
TAC KDCAPPLA
TAC KDCDIAGA
TAC KDCLOGA
TAC KDCINFA
TAC KDCHELPA
TAC KDCSHUTA
DEFAULT TAC TYPE=D,ADMIN=N,PROGRAM=(STD)
TAC KDCBADTC,PROGRAM=BADTAC
TAC 1,PROGRAM=TPREAD
TAC 2,PROGRAM=TPUPDAT
TAC 3,PROGRAM=TPUPDAT
TAC 4,PROGRAM=TPUPDAT
REM *****
REM *****          USER STATEMENTS          *****
REM *****
USER  SUSIE,KSET=BUNCH1,PERMIT=ADMIN,PASS=(EMPLOY,DARK)
USER  GERTRUDE,KSET=BUNCH2,STATUS=ON,PASS=(EMPLOY,DARK)
USER  BARBARA,KSET=BUNCH3,STATUS=ON,PASS=(EMPLOY,DARK)
REM *****
REM *****          PTERM/LTERM STATEMENTS          *****
REM *****
DEFAULT PTERM PRONAM=PT01,PTYPE=T9750
PTERM DDT01,LTERM=UTMDT1
PTERM DDT02,LTERM=UTMDT2
PTERM DDT03,LTERM=UTMDT3
DEFAULT PTERM PRONAM=TC01,PTYPE=T9022,USAGE=0
PTERM G01,LTERM=PRINTER,CONNECT=A
LTERM UTMDT1,KSET=BUNCH1
LTERM UTMDT2,LOCK=4,KSET=BUNCH1
LTERM UTMDT3,LOCK=5,KSET=BUNCH1
LTERM PRINTER,USAGE=0
REM *****
REM *****          KSET STATEMENTS          *****
REM *****
KSET  BUNCH1,KEYS=(1,2,3,4,5)
KSET  BUNCH2,KEYS=(1,2,4)
KSET  BUNCH3,KEYS=(1)
REM *****

```



```

REM *****          TLS STATEMENTS          *****
REM *****
TLS   TLSA
TLS   TLSB
END
/SKIP .CALL
/STEP
/WR-TEXT ' >> Definition of configuration incorrect << '
/SKIP .CALL
/.A FILE $TSOS.SYSMAC.FHS.060, LINK=OLDLIB
/   FILE $TSOS.SYSLIB.UTM.040.ASS, LINK=OLDLIB2
/   ER *
/   EXEC $ASSEMB
*COMOPT OLDLIB, OLDLIB2, OLDLIB3, SOURCE=UTM.SRC.ROOT, SAVLST
*COMOPT MODULE=UTM.TP.PLAMLIB.NOSHARE(UTMFSHR)
*END HALT
/   EXEC $ASSEMB
*COMOPT OLDLIB, OLDLIB2, OLDLIB3, SOURCE=UTM.SRC.SHARETAB, SAVLST
*COMOPT MODULE=UTM.TP.PLAMLIB.SHARE(SHARETAB)
*END HALT
/   SKIP .CALL
/   STEP
/   WR-TEXT ' >> Generation of linkage program incorrect << '
/   SKIP .CALL
/.B SYSFILE SYSLST=UTM.SHARE.LNK
/SETSW ON=(4,5)
/EXEC $EDT
TPREAD
TPUPDAT
BADTAC
OPNCLOS
@W'SHRNAM' OVERWRITE
@H
/SETSW OFF=(4,5)
/CALL FOR1.P.SHARE, SHRNames=SHRnam, LIBN=UTM.TP.PLAMLIB.NOSHARE, -
/   LIBT=LMS, LIBS=UTM.TP.PLAMLIB.SHARE, XS=NO, HELP=NO, SHRMOD=SHRMOD, -
/   ADAPTn=ADAPTn, ADAPT=ADAPT
/   EXEC $TSOSLNK
PROG MODFSHR, FILENAM=UTM.SHARE.L, LOADPT=&LD
INCLUDE UTMFSHR, UTM.TP.PLAMLIB.NOSHARE
RESOLVE ,UTM.TP.PLAMLIB.NOSHARE
RESOLVE , $TSOS.SYSLNK.FOR1-LZS.022.FPOOL
RESOLVE , $TSOS.SYSLNK.UTM.040.SPLRTS
RESOLVE , $TSOS.SYSLNK.FOR1-LZS.022.DYNAMIC
RESOLVE , $TSOS.SYSLNK.CRTE
RESOLVE , $TSOS.SYSLNK.UTM.040
NCAL
END

```

```

/  SKIP .CALL
/  STEP
/  WR-TEXT ' >> Linking of FOR1-UTM application incorrect << '
/  SKIP .CALL
/.K WR-TEXT '          Header for KDCLOAD'
/  WR-TEXT '          Create FGG'
/  ER FSHR.KDCFILE.PROG
/  STEP
/  CAT FSHR.KDCFILE.PROG,GEN=3
/  FILE FSHR.KDCFILE.PROG(*1)
/  EXEC UTM.SHARE.L
PREPARE FILEBASE=FSHR.KDCFILE
END
/  SKIP .CALL
/.P WR-TEXT '          Start the UTM FOR1 application'
/  CAT FSHR.KDCFILE.PROG,BASE=0,STATE=U
/  ENTER UTM.BSP.E,PRIORITY=(,EXPRESS),TIME=NTL
/  SKIP .CALL
/.S WR-TEXT '          Please enter PTERM,PRONAME'
/  SYSFILE TASKLIB=$TSOS.SYSLNK.FOR1-LZS.022.DYNAMIC
/  FILE FSHR.KDCFILE.SYSLOG,LINK=SYSLOG,SHARUPD=YES
/  FILE FSHR.KDCFILE.KDCA,LINK=KDCFILE,SHARUPD=NO
/  FILE UTM.FOR.ADDRESSES,LINK=DSET10,FCBTYPE=ISAM,KEYLEN=16,KEYPOS=1, -
/  RECFORM=F,RECSIZE=120,OPEN=INOUT
/  EXEC UTM.SHARE.L
.UTM START FILEBASE=FSHR.KDCFILE
      START TASKS=1
.FHS MAPLIB=UTM.BSP.IFGLIB
      END
/  STEP
/  ENDP
/.CALL SETSW OFF=(4,5)
/  CALL UTM.BSP.P
/  STEP
/.E OPT MSG=F
/  TCHNG OFLOW=ACK
/  SETSW OFF=(4)
/  ENDP

```

---

# 4 Data structures for Fortran

## 4.1 Data structure KCAPROF

```
*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
*
*****+***
*                                     *
* Parameter Information of APRO Call                                  *
*                                                                     *
* Include for FOR1 : KCAPROF                                        *
*                                                                     *
*****+***
*
* CHARACTER    KCAAPRO * (43)
*
* INTEGER     KCAVERS * 2
*                                     version
* CHARACTER   KCAFUPOL * 1
*                                     polarized / shared
* CHARACTER   KCAFUHSH * 1
*                                     handshake
* CHARACTER   KCAFUCOM * 1
*                                     commit
* CHARACTER   KCAFUCHN * 1
*                                     chained / unchained
* CHARACTER   KCSECTYP * 1
*                                     security type none/same/program
* CHARACTER   KCUIDTYP * 1
*                                     string type printable/T61/octet
* INTEGER     KCUIDLTH * 2
*                                     lth of userid
* CHARACTER   KCUSER * 16
*                                     userid
* CHARACTER   KCPWDTYP * 1
*                                     string type printable/T61/octet
* INTEGER     KCPWDLTH * 2
*                                     lth of password
* CHARACTER   KCPSWORD * (16)
*                                     password
```

```
EQUIVALENCE ( KCAAPRO(1:) , KCAVERS)
EQUIVALENCE ( KCAAPRO(3:) , KCFUPOL)
EQUIVALENCE ( KCAAPRO(4:) , KCFUHSH)
EQUIVALENCE ( KCAAPRO(5:) , KCFUCOM)
EQUIVALENCE ( KCAAPRO(6:) , KCFUCHN)
EQUIVALENCE ( KCAAPRO(7:) , KCSECTYP)
EQUIVALENCE ( KCAAPRO(8:) , KCUIDTYP)
EQUIVALENCE ( KCAAPRO(9:) , KCUIDLTH)
EQUIVALENCE ( KCAAPRO(11:) , KCUSER)
EQUIVALENCE ( KCAAPRO(28:) , KCPWDTYP)
EQUIVALENCE ( KCAAPRO(29:) , KCPWDLTH)
EQUIVALENCE ( KCAAPRO(31:) , KCPSWORD)
```

```
*****
```

## 4.2 Data structure KCATF

```

*
*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM 4.0          +***
*
*****
*                                     *
*      Attribut Functions                                           *
*                                     *
*      Include for FOR1 : KCATF                                       *
*                                     *
*****
*
      CHARACTER KCATF * (56)
*
      INTEGER KCPROT * 2 /4360/
*                                     PROT,NORM
      INTEGER KCAPN * 2
*                                     PROT,NORM
      INTEGER KCNPN * 2
*                                     PROT,NORM
      INTEGER KCALPH * 2 /20512/
*                                     UNPROT,BRT,PRINT
      INTEGER KCUNPR * 2
*                                     UNPROT,BRT
      INTEGER KCHINT * 2
*                                     UNPROT,BRT
      INTEGER KCAUH * 2
*                                     UNPROT,BRT
      INTEGER KCAPD * 2 /4356/
*                                     PROT,DRK
      INTEGER KCNPD * 2
*                                     PROT,DRK
      INTEGER KCNUME * 2 /21024/
*                                     UNPROT,BRT,NUM
      INTEGER KCNUH * 2
*                                     UNPROT,BRT,NUM
      INTEGER KCDETE * 2 /6432/
*                                     PROT,BRT,DET
      INTEGER KCAPHD * 2
*                                     PROT,BRT,DET
      INTEGER KCNPHD * 2
*                                     PROT,BRT,DET
      INTEGER KCNINT * 2 /20488/
*                                     UNPROT,NORM
      INTEGER KCAUN * 2
*                                     UNPROT,NORM
      INTEGER KCAPH * 2 /4384/
*                                     PROT,BRT
      INTEGER KCNPH * 2
*                                     PROT,BRT
      INTEGER KCDINT * 2 /20484/

```

*	INTEGER	KCAUD * 2		UNPROT, DRK
*	INTEGER	KCAPI * 2	/4362/	UNPROT, DRK
*	INTEGER	KCNPI * 2		PROT, NORM, ITAL
*	INTEGER	KCITAL * 2	/20514/	PROT, NORM, ITAL
*	INTEGER	KCAUI * 2		UNPROT, BRT, ITAL
*	INTEGER	KCAPS * 2	/4361/	UNPROT, BRT, ITAL
*	INTEGER	KCNPS * 2		PROT, NORM, SIGN
*	INTEGER	KCSIGN * 2	/20513/	PROT, NORM, SIGN
*	INTEGER	KCAUS * 2		UNPROT, BRT, SIGN
*	INTEGER	KCPREM * 2	/5152/	UNPROT, BRT, SIGN
*	INTEGER	KCAUHP * 2		FSET, BRT
*	INTEGER	KCNUN * 2	/21000/	FSET, BRT
*	INTEGER	KCAUNP * 2	/5128/	UNPROT, NORM, NUM
*	INTEGER	KCNUD * 2	/20996/	FSET, NORM
*	INTEGER	KCNUNP * 2	/5640/	UNPROT, DRK, NUM
*	INTEGER	KCNUI * 2	/21026/	FSET, NORM, NUM
*	INTEGER	KCNUHP * 2	/5664/	UNPROT, BRT, ITAL, NUM
*	INTEGER	KCNUS * 2	/21025/	FSET, BRT, NUM
*	INTEGER	KCAPHP * 2	/12320/	UNPROT, BRT, SIGN, NUM
*	INTEGER	KCNPHP * 2		PROTRET, BRT
*	INTEGER	KCAPNP * 2	/12296/	PROTRET, BRT
*	INTEGER	KCNPNP * 2		PROTRET, NORM
*	INTEGER	KCAPND * 2	/6408/	PROTRET, NORM
*	INTEGER	KCNPND * 2		PROT, NORM, DET
*	INTEGER	KCAPSD * 2		PROT, NORM, DET
*	INTEGER	KCNPSD * 2		PROT, NORM, DET
*	INTEGER	KCAUND * 2	/22536/	PROT, NORM, DET
*	INTEGER	KCNUND * 2		UNPROT, NORM, DET
*	INTEGER	KCAPID * 2	/6410/	UNPROT, NORM, DET
*	INTEGER	KCNPID * 2		PROT, NORM, DET, ITAL

```

*          INTEGER      KCAUHD * 2    /22560/      PROT,NORM,DET,ITAL
*
*          INTEGER      KCNUHD * 2
*
*          INTEGER      KCAUSD * 2
*
*          INTEGER      KCNUSD * 2
*
*          INTEGER      KCAUID * 2    /22562/
*
*          INTEGER      KCNUID * 2
*
*
*          EQUIVALENCE (KCATF( 1:), KCPROT , KCAPN , KCNPN )
*          EQUIVALENCE (KCATF( 3:), KCALPH , KCUNPR , KCHINT , KCAUH )
*          EQUIVALENCE (KCATF( 5:), KCAPD , KCNPD )
*          EQUIVALENCE (KCATF( 7:), KCNUME , KCNUH )
*          EQUIVALENCE (KCATF( 9:), KCDETE , KCAPHD , KCNPHD )
*          EQUIVALENCE (KCATF(11:), KCNINT , KCAUN )
*          EQUIVALENCE (KCATF(13:), KCAPH , KCNPH )
*          EQUIVALENCE (KCATF(15:), KCDINT , KCAUD )
*          EQUIVALENCE (KCATF(17:), KCAPI , KCNPI )
*          EQUIVALENCE (KCATF(19:), KCITAL , KCAUI )
*          EQUIVALENCE (KCATF(21:), KCAPS , KCNPS )
*          EQUIVALENCE (KCATF(23:), KCSIGN , KCAUS )
*          EQUIVALENCE (KCATF(25:), KCPREM , KCAUHP )
*          EQUIVALENCE (KCATF(27:), KCNUN )
*          EQUIVALENCE (KCATF(29:), KCAUNP )
*          EQUIVALENCE (KCATF(31:), KCNUD )
*          EQUIVALENCE (KCATF(33:), KCNUNP )
*          EQUIVALENCE (KCATF(35:), KCNUI )
*          EQUIVALENCE (KCATF(37:), KCNUHP )
*          EQUIVALENCE (KCATF(39:), KCNUS )
*          EQUIVALENCE (KCATF(41:), KCAPHP , KCNPHP )
*          EQUIVALENCE (KCATF(43:), KCAPNP , KCNPNP )
*          EQUIVALENCE (KCATF(45:), KCAPND , KCNPNND , KCAPSD , KCNPSD )
*          EQUIVALENCE (KCATF(47:), KCAUND , KCNUND )
*          EQUIVALENCE (KCATF(49:), KCAPID , KCNPID )
*          EQUIVALENCE (KCATF(51:), KCAUHD , KCNUHD , KCAUSD , KCNUSD )
*          EQUIVALENCE (KCATF(55:), KCAUID , KCNUID )
*
*
*
*****

```

## 4.3 Data structure KCCAF

```

*
*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                     +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0      +***
*
*****
*                                     *
*      Communication Area for UTM ( KDCKB )                       *
*                                     *
*      Include for FOR1 : KCCAF                                   *
*                                     *
*****
*-----*
*      Communication Area Header          ( 84 bytes )           *
*-----*
*
*      CHARACTER KCCAHDR * (84)
*
*      CHARACTER KCUSERID * 8
*
*      CHARACTER KCCV * (24)
*
*      CHARACTER KCCVTAC * 8
*
*      CHARACTER KCCVDATE * (9)
*
*      CHARACTER KCCVDAY * 2
*
*      CHARACTER KCCVMONTH * 2
*
*      CHARACTER KCCVYEAR * 2
*
*      CHARACTER KCCVDOY * 3
*
*      CHARACTER KCCVTIME * (6)
*
*      CHARACTER KCCVHOUR * 2
*
*      CHARACTER KCCVMINUTE * 2
*
*      CHARACTER KCCVSECOND * 2
*
*      CHARACTER KCCVSTATUS * 1
*
*      CHARACTER KCPR * (16)
*
*      CHARACTER KCPRTAC * 8
*
*      CHARACTER KCPRTIME * (6)

```

communication area header :

user identification

conversation specific data fields :

transaction code

date :

day

month

year

day of year

time :

hour

minute

second

conversation identification

data specific to current program run :

transaction code



```

*          CHARACTER          KCPRHOUR * 2          time :
*          CHARACTER          KCPRMINUTE * 2        hour
*          CHARACTER          KCPRSECOND * 2       minute
*          CHARACTER          KCCARD * 1           second
*          CHARACTER          KCTAIND * 1          A = card in reader
*          CHARACTER          KCLOGTER * 8         transaction indicator
*          CHARACTER          KCTERMN * 2         logical terminal name (LTERM)
*          CHARACTER          KCLPA * 2           device type of physical
*          INTEGER           KCHSTA * 2           terminal
*          CHARACTER          KCSTA * (3)         maximum length of KB program
*          INTEGER           KCDSTA * 2           area
*          CHARACTER          KCPRIND * 1         stack information
*          CHARACTER          KCOF1 * 1           current stack level
*          CHARACTER          KCOF2 * 1           change in stack level
*          CHARACTER          KCTARB * 1         program indicator
*          CHARACTER          KCTARB * 1         osi-tp function1
*          CHARACTER          KCTARB * 1         osi-tp function2
*          CHARACTER          KCTARB * 1         transaction is marked rollback
*
EQUIVALENCE ( KCCAHDR ( 1: 8 ) , KUSERID )
EQUIVALENCE ( KCCAHDR ( 9:32 ) , KCCV )
EQUIVALENCE ( KCCAHDR ( 9:16 ) , KCCVTAC )
EQUIVALENCE ( KCCAHDR ( 17:25 ) , KCCVDATE )
EQUIVALENCE ( KCCAHDR ( 17:18 ) , KCCVDAY )
EQUIVALENCE ( KCCAHDR ( 19:20 ) , KCCVMONTH )
EQUIVALENCE ( KCCAHDR ( 21:22 ) , KCCVYEAR )
EQUIVALENCE ( KCCAHDR ( 23:25 ) , KCCVDOY )
EQUIVALENCE ( KCCAHDR ( 26:31 ) , KCCVTIME )
EQUIVALENCE ( KCCAHDR ( 26:27 ) , KCCVHOUR )
EQUIVALENCE ( KCCAHDR ( 28:29 ) , KCCVMINUTE )
EQUIVALENCE ( KCCAHDR ( 30:31 ) , KCCVSECOND )
EQUIVALENCE ( KCCAHDR ( 32:32 ) , KCCVSTATUS )
EQUIVALENCE ( KCCAHDR ( 33:48 ) , KCPR )
EQUIVALENCE ( KCCAHDR ( 33:40 ) , KCPRTAC )
EQUIVALENCE ( KCCAHDR ( 41:46 ) , KCPRTIME )
EQUIVALENCE ( KCCAHDR ( 41:42 ) , KCPRHOUR )
EQUIVALENCE ( KCCAHDR ( 43:44 ) , KCPRMINUTE )
EQUIVALENCE ( KCCAHDR ( 45:46 ) , KCPRSECOND )
EQUIVALENCE ( KCCAHDR ( 47:47 ) , KCCARD )
EQUIVALENCE ( KCCAHDR ( 48:48 ) , KCTAIND )
EQUIVALENCE ( KCCAHDR ( 49:56 ) , KCLOGTER )
EQUIVALENCE ( KCCAHDR ( 57:58 ) , KCTERMN )
EQUIVALENCE ( KCCAHDR ( 59:60 ) , KCLPA )
EQUIVALENCE ( KCCAHDR ( 61:63 ) , KCSTA )

```

```

EQUIVALENCE ( KCCAHDR ( 61:62 ) , KCHSTA      )
EQUIVALENCE ( KCCAHDR ( 63:63 ) , KCDSTA      )
EQUIVALENCE ( KCCAHDR ( 65:65 ) , KCPRIND     )
EQUIVALENCE ( KCCAHDR ( 66:66 ) , KCOF1      )
EQUIVALENCE ( KCCAHDR ( 67:67 ) , KCOF2      )
EQUIVALENCE ( KCCAHDR ( 68:68 ) , KCTARB      )
*
*-----*
* CA Return Area ( 32 bytes ) *
*-----*
*
CHARACTER KCCARTI * (32)
*
CHARACTER KCRI * 2 return information from UTM :
*
INTEGER KCRDF * 2 return identification
*
INTEGER KCRML * 2 return device feature
*
CHARACTER KCRINFCC * (3) return length
*
CHARACTER KCRSTATE * 2 INFO call return code :
*
CHARACTER KCRST * (2) conversation and transaction
*
CHARACTER KCRCVST * 1 status
*
CHARACTER KCRFAST * 1 conversation status
*
CHARACTER KCRSIGN * (4) transaction status
*
CHARACTER KCRSIGN1 * 1 status of sign-on :
*
CHARACTER KCRSIGN2 * 2 primary code
*
CHARACTER KCRMGT * 1 secondary code
*
CHARACTER KCRC * (8) return info mget
*
CHARACTER KCRCCC * 3 return codes :
*
CHARACTER KCRCID * 1 KDCS error code
*
CHARACTER KCRCDC * 4 product indicator
*
CHARACTER KCRFN * 8 P=produktion, T=UTM-T (test)
*
CHARACTER KCRPI * 8 additional error code from UTM
*
CHARACTER KCRUS * 8 (not compatible)
*
CHARACTER KCRUS * 8 return format name
*
CHARACTER KCRUS * 8 return conversation id
*
CHARACTER KCRUS * 8 return user id (sign st)
*
EQUIVALENCE ( KCCARTI ( 1: 2 ) , KCRI , KCRDF )
EQUIVALENCE ( KCCARTI ( 3: 4 ) , KCRML )
EQUIVALENCE ( KCCARTI ( 5: 7 ) , KCRINFCC )
EQUIVALENCE ( KCCARTI ( 5: 6 ) , KCRSTATE )
EQUIVALENCE ( KCCARTI ( 5: 8 ) , KCRST , KCRSIGN )

```

```

EQUIVALENCE ( KCCARTI ( 5: 5 ) , KCRCVST , KCRSIGN1 )
EQUIVALENCE ( KCCARTI ( 6: 6 ) , KCRTAST )
EQUIVALENCE ( KCCARTI ( 6: 7 ) , KCRSIGN2 )
EQUIVALENCE ( KCCARTI ( 8: 8 ) , KCRMGT )
EQUIVALENCE ( KCCARTI ( 9:12 ) , KCRC )
EQUIVALENCE ( KCCARTI ( 9:11 ) , KCRCCC )
EQUIVALENCE ( KCCARTI ( 12:12 ) , KCRCID )
EQUIVALENCE ( KCCARTI ( 13:16 ) , KCRCDC )
EQUIVALENCE ( KCCARTI ( 17:24 ) , KCRFN )
EQUIVALENCE ( KCCARTI ( 25:32 ) , KCRPI , KCRUS )
*
*
*      COMMON /KDCKB/  KCCAHDR , KCCARTI
*
*-----*
*      CA Program Area
*-----*
```

## 4.4 Data structure KCCFF

```

*
*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
*
*****
* control fields for input exit                                     *
* Include for FOR1 : KCCFF                                         *
*                                                                     *
*****
*      CHARACTER  KCCFF * (7744)
*
*      CHARACTER  KCCFCREM * 8
*      CHARACTER  KCCFCFLD * 132      remark as defined by IFG
*      INTEGER    KCCFNOCF           control field
*      CHARACTER  KCCFS * 7600       number of control fields
*      CHARACTER  KCCFFNAM01 * 8     array of control
*      CHARACTER  KCCFREMO1 * 8     field information
*      CHARACTER  KCCFREMO1 * 8     format name
*      INTEGER    KCCFLOFLO1        remark from IFG
*      CHARACTER  KCCFFLD01 * 132   length of control field
*      CHARACTER  KCCFFLD01 * 132   control field
*
*      CHARACTER  KCCFFNAM02 * 8
*      CHARACTER  KCCFREMO2 * 8
*      INTEGER    KCCFLOFLO2
*      CHARACTER  KCCFFLD02 * 132
*
*      CHARACTER  KCCFFNAM03 * 8
*      CHARACTER  KCCFREMO3 * 8
*      INTEGER    KCCFLOFLO3
*      CHARACTER  KCCFFLD03 * 132
*
*      CHARACTER  KCCFFNAM04 * 8
*      CHARACTER  KCCFREMO4 * 8
*      INTEGER    KCCFLOFLO4
*      CHARACTER  KCCFFLD04 * 132
*
*      CHARACTER  KCCFFNAM05 * 8
*      CHARACTER  KCCFREMO5 * 8
*      INTEGER    KCCFLOFLO5
*      CHARACTER  KCCFFLD05 * 132

```

```
*      CHARACTER      KCCFFNAM06 * 8
      CHARACTER      KCCFREMO6 * 8
      INTEGER        KCCFLOFLO6
      CHARACTER      KCCFFLD06 * 132
*
      CHARACTER      KCCFFNAM07 * 8
      CHARACTER      KCCFREMO7 * 8
      INTEGER        KCCFLOFLO7
      CHARACTER      KCCFFLD07 * 132
*
      CHARACTER      KCCFFNAM08 * 8
      CHARACTER      KCCFREMO8 * 8
      INTEGER        KCCFLOFLO8
      CHARACTER      KCCFFLD08 * 132
*
      CHARACTER      KCCFFNAM09 * 8
      CHARACTER      KCCFREMO9 * 8
      INTEGER        KCCFLOFLO9
      CHARACTER      KCCFFLD09 * 132
*
      CHARACTER      KCCFFNAM10 * 8
      CHARACTER      KCCFREMO10 * 8
      INTEGER        KCCFLOFLO10
      CHARACTER      KCCFFLD10 * 132
*
*
      CHARACTER      KCCFFNAM11 * 8
      CHARACTER      KCCFREMO11 * 8
      INTEGER        KCCFLOFLO11
      CHARACTER      KCCFFLD11 * 132
*
      CHARACTER      KCCFFNAM12 * 8
      CHARACTER      KCCFREMO12 * 8
      INTEGER        KCCFLOFLO12
      CHARACTER      KCCFFLD12 * 132
*
      CHARACTER      KCCFFNAM13 * 8
      CHARACTER      KCCFREMO13 * 8
      INTEGER        KCCFLOFLO13
      CHARACTER      KCCFFLD13 * 132
*
      CHARACTER      KCCFFNAM14 * 8
      CHARACTER      KCCFREMO14 * 8
      INTEGER        KCCFLOFLO14
      CHARACTER      KCCFFLD14 * 132
*
      CHARACTER      KCCFFNAM15 * 8
      CHARACTER      KCCFREMO15 * 8
      INTEGER        KCCFLOFLO15
      CHARACTER      KCCFFLD15 * 132
*
      CHARACTER      KCCFFNAM16 * 8
      CHARACTER      KCCFREMO16 * 8
      INTEGER        KCCFLOFLO16
      CHARACTER      KCCFFLD16 * 132
*
      CHARACTER      KCCFFNAM17 * 8
      CHARACTER      KCCFREMO17 * 8
      INTEGER        KCCFLOFLO17
```

*	CHARACTER	KCCFFLD17 * 132
	CHARACTER	KCCFFNAM18 * 8
	CHARACTER	KCCFREML18 * 8
	INTEGER	KCCFLOFL18
	CHARACTER	KCCFFLD18 * 132
*		
	CHARACTER	KCCFFNAM19 * 8
	CHARACTER	KCCFREML19 * 8
	INTEGER	KCCFLOFL19
	CHARACTER	KCCFFLD19 * 132
*		
	CHARACTER	KCCFFNAM20 * 8
	CHARACTER	KCCFREML20 * 8
	INTEGER	KCCFLOFL20
	CHARACTER	KCCFFLD20 * 132
*		
*		
	CHARACTER	KCCFFNAM21 * 8
	CHARACTER	KCCFREML21 * 8
	INTEGER	KCCFLOFL21
	CHARACTER	KCCFFLD21 * 132
*		
	CHARACTER	KCCFFNAM22 * 8
	CHARACTER	KCCFREML22 * 8
	INTEGER	KCCFLOFL22
	CHARACTER	KCCFFLD22 * 132
*		
	CHARACTER	KCCFFNAM23 * 8
	CHARACTER	KCCFREML23 * 8
	INTEGER	KCCFLOFL23
	CHARACTER	KCCFFLD23 * 132
*		
	CHARACTER	KCCFFNAM24 * 8
	CHARACTER	KCCFREML24 * 8
	INTEGER	KCCFLOFL24
	CHARACTER	KCCFFLD24 * 132
*		
	CHARACTER	KCCFFNAM25 * 8
	CHARACTER	KCCFREML25 * 8
	INTEGER	KCCFLOFL25
	CHARACTER	KCCFFLD25 * 132
*		
	CHARACTER	KCCFFNAM26 * 8
	CHARACTER	KCCFREML26 * 8
	INTEGER	KCCFLOFL26
	CHARACTER	KCCFFLD26 * 132
*		
	CHARACTER	KCCFFNAM27 * 8
	CHARACTER	KCCFREML27 * 8
	INTEGER	KCCFLOFL27
	CHARACTER	KCCFFLD27 * 132
*		
	CHARACTER	KCCFFNAM28 * 8
	CHARACTER	KCCFREML28 * 8
	INTEGER	KCCFLOFL28
	CHARACTER	KCCFFLD28 * 132
*		
	CHARACTER	KCCFFNAM29 * 8
	CHARACTER	KCCFREML29 * 8

	INTEGER	KCCFLOFL29	
*	CHARACTER	KCCFFLD29	* 132
	CHARACTER	KCCFFNAM30	* 8
	CHARACTER	KCCFRE30	* 8
	INTEGER	KCCFLOFL30	
*	CHARACTER	KCCFFLD30	* 132
*			
	CHARACTER	KCCFFNAM31	* 8
	CHARACTER	KCCFRE31	* 8
	INTEGER	KCCFLOFL31	
*	CHARACTER	KCCFFLD31	* 132
	CHARACTER	KCCFFNAM32	* 8
	CHARACTER	KCCFRE32	* 8
	INTEGER	KCCFLOFL32	
*	CHARACTER	KCCFFLD32	* 132
	CHARACTER	KCCFFNAM33	* 8
	CHARACTER	KCCFRE33	* 8
	INTEGER	KCCFLOFL33	
*	CHARACTER	KCCFFLD33	* 132
	CHARACTER	KCCFFNAM34	* 8
	CHARACTER	KCCFRE34	* 8
	INTEGER	KCCFLOFL34	
*	CHARACTER	KCCFFLD34	* 132
	CHARACTER	KCCFFNAM35	* 8
	CHARACTER	KCCFRE35	* 8
	INTEGER	KCCFLOFL35	
*	CHARACTER	KCCFFLD35	* 132
	CHARACTER	KCCFFNAM36	* 8
	CHARACTER	KCCFRE36	* 8
	INTEGER	KCCFLOFL36	
*	CHARACTER	KCCFFLD36	* 132
	CHARACTER	KCCFFNAM37	* 8
	CHARACTER	KCCFRE37	* 8
	INTEGER	KCCFLOFL37	
*	CHARACTER	KCCFFLD37	* 132
	CHARACTER	KCCFFNAM38	* 8
	CHARACTER	KCCFRE38	* 8
	INTEGER	KCCFLOFL38	
*	CHARACTER	KCCFFLD38	* 132
	CHARACTER	KCCFFNAM39	* 8
	CHARACTER	KCCFRE39	* 8
	INTEGER	KCCFLOFL39	
*	CHARACTER	KCCFFLD39	* 132
	CHARACTER	KCCFFNAM40	* 8
	CHARACTER	KCCFRE40	* 8
	INTEGER	KCCFLOFL40	
*	CHARACTER	KCCFFLD40	* 132
*			
*			

```

CHARACTER      KCCFFNAM41 * 8
CHARACTER      KCCFRE41 * 8
INTEGER        KCCFLOFL41
CHARACTER      KCCFFLD41 * 132
*
CHARACTER      KCCFFNAM42 * 8
CHARACTER      KCCFRE42 * 8
INTEGER        KCCFLOFL42
CHARACTER      KCCFFLD42 * 132
*
CHARACTER      KCCFFNAM43 * 8
CHARACTER      KCCFRE43 * 8
INTEGER        KCCFLOFL43
CHARACTER      KCCFFLD43 * 132
*
CHARACTER      KCCFFNAM44 * 8
CHARACTER      KCCFRE44 * 8
INTEGER        KCCFLOFL44
CHARACTER      KCCFFLD44 * 132
*
CHARACTER      KCCFFNAM45 * 8
CHARACTER      KCCFRE45 * 8
INTEGER        KCCFLOFL45
CHARACTER      KCCFFLD45 * 132
*
CHARACTER      KCCFFNAM46 * 8
CHARACTER      KCCFRE46 * 8
INTEGER        KCCFLOFL46
CHARACTER      KCCFFLD46 * 132
*
CHARACTER      KCCFFNAM47 * 8
CHARACTER      KCCFRE47 * 8
INTEGER        KCCFLOFL47
CHARACTER      KCCFFLD47 * 132
*
CHARACTER      KCCFFNAM48 * 8
CHARACTER      KCCFRE48 * 8
INTEGER        KCCFLOFL48
CHARACTER      KCCFFLD48 * 132
*
CHARACTER      KCCFFNAM49 * 8
CHARACTER      KCCFRE49 * 8
INTEGER        KCCFLOFL49
CHARACTER      KCCFFLD49 * 132
*
CHARACTER      KCCFFNAM50 * 8
CHARACTER      KCCFRE50 * 8
INTEGER        KCCFLOFL50
CHARACTER      KCCFFLD50 * 132
*
*
*
*
EQUIVALENCE ( KCCFF(1:)      , KCCFCREM )
EQUIVALENCE ( KCCFF(9:)      , KCCFCFLD )
EQUIVALENCE ( KCCFF(141:)    , KCCFNOCF )
EQUIVALENCE ( KCCFF(145:)    , KCCFS      )
*
*
*

```



```

EQUIVALENCE ( KCCFS(1:)      , KCCFFNAM01 )
EQUIVALENCE ( KCCFS(9:)     , KCCFREM01 )
EQUIVALENCE ( KCCFS(17:)    , KCCFLOFL01 )
EQUIVALENCE ( KCCFS(21:)    , KCCFFLD01 )
*
EQUIVALENCE ( KCCFS(153:)   , KCCFFNAM02 )
EQUIVALENCE ( KCCFS(161:)   , KCCFREM02 )
EQUIVALENCE ( KCCFS(169:)   , KCCFLOFL02 )
EQUIVALENCE ( KCCFS(173:)   , KCCFFLD02 )
*
EQUIVALENCE ( KCCFS(305:)   , KCCFFNAM03 )
EQUIVALENCE ( KCCFS(313:)   , KCCFREM03 )
EQUIVALENCE ( KCCFS(321:)   , KCCFLOFL03 )
EQUIVALENCE ( KCCFS(325:)   , KCCFFLD03 )
*
EQUIVALENCE ( KCCFS(457:)   , KCCFFNAM04 )
EQUIVALENCE ( KCCFS(465:)   , KCCFREM04 )
EQUIVALENCE ( KCCFS(473:)   , KCCFLOFL04 )
EQUIVALENCE ( KCCFS(477:)   , KCCFFLD04 )
*
EQUIVALENCE ( KCCFS(609:)   , KCCFFNAM05 )
EQUIVALENCE ( KCCFS(617:)   , KCCFREM05 )
EQUIVALENCE ( KCCFS(625:)   , KCCFLOFL05 )
EQUIVALENCE ( KCCFS(629:)   , KCCFFLD05 )
*
EQUIVALENCE ( KCCFS(761:)   , KCCFFNAM06 )
EQUIVALENCE ( KCCFS(769:)   , KCCFREM06 )
EQUIVALENCE ( KCCFS(777:)   , KCCFLOFL06 )
EQUIVALENCE ( KCCFS(781:)   , KCCFFLD06 )
*
EQUIVALENCE ( KCCFS(913:)   , KCCFFNAM07 )
EQUIVALENCE ( KCCFS(921:)   , KCCFREM07 )
EQUIVALENCE ( KCCFS(929:)   , KCCFLOFL07 )
EQUIVALENCE ( KCCFS(933:)   , KCCFFLD07 )
*
EQUIVALENCE ( KCCFS(1065:)  , KCCFFNAM08 )
EQUIVALENCE ( KCCFS(1073:)  , KCCFREM08 )
EQUIVALENCE ( KCCFS(1081:)  , KCCFLOFL08 )
EQUIVALENCE ( KCCFS(1085:)  , KCCFFLD08 )
*
EQUIVALENCE ( KCCFS(1217:)  , KCCFFNAM09 )
EQUIVALENCE ( KCCFS(1225:)  , KCCFREM09 )
EQUIVALENCE ( KCCFS(1233:)  , KCCFLOFL09 )
EQUIVALENCE ( KCCFS(1237:)  , KCCFFLD09 )
*
EQUIVALENCE ( KCCFS(1369:)  , KCCFFNAM10 )
EQUIVALENCE ( KCCFS(1377:)  , KCCFREM10 )
EQUIVALENCE ( KCCFS(1385:)  , KCCFLOFL10 )
EQUIVALENCE ( KCCFS(1389:)  , KCCFFLD10 )
*
*
EQUIVALENCE ( KCCFS(1521:)  , KCCFFNAM11 )
EQUIVALENCE ( KCCFS(1529:)  , KCCFREM11 )
EQUIVALENCE ( KCCFS(1537:)  , KCCFLOFL11 )
EQUIVALENCE ( KCCFS(1541:)  , KCCFFLD11 )
*
EQUIVALENCE ( KCCFS(1673:)  , KCCFFNAM12 )
EQUIVALENCE ( KCCFS(1681:)  , KCCFREM12 )
EQUIVALENCE ( KCCFS(1689:)  , KCCFLOFL12 )
EQUIVALENCE ( KCCFS(1693:)  , KCCFFLD12 )

```

```
*
EQUIVALENCE ( KCCFS(1825:), KCCFFNAM13 )
EQUIVALENCE ( KCCFS(1833:), KCCFREM13 )
EQUIVALENCE ( KCCFS(1841:), KCCFLOFL13 )
EQUIVALENCE ( KCCFS(1845:), KCCFFLD13 )
*
EQUIVALENCE ( KCCFS(1977:), KCCFFNAM14 )
EQUIVALENCE ( KCCFS(1985:), KCCFREM14 )
EQUIVALENCE ( KCCFS(1993:), KCCFLOFL14 )
EQUIVALENCE ( KCCFS(1997:), KCCFFLD14 )
*
EQUIVALENCE ( KCCFS(2129:), KCCFFNAM15 )
EQUIVALENCE ( KCCFS(2137:), KCCFREM15 )
EQUIVALENCE ( KCCFS(2145:), KCCFLOFL15 )
EQUIVALENCE ( KCCFS(2149:), KCCFFLD15 )
*
EQUIVALENCE ( KCCFS(2281:), KCCFFNAM16 )
EQUIVALENCE ( KCCFS(2289:), KCCFREM16 )
EQUIVALENCE ( KCCFS(2297:), KCCFLOFL16 )
EQUIVALENCE ( KCCFS(2301:), KCCFFLD16 )
*
EQUIVALENCE ( KCCFS(2433:), KCCFFNAM17 )
EQUIVALENCE ( KCCFS(2441:), KCCFREM17 )
EQUIVALENCE ( KCCFS(2449:), KCCFLOFL17 )
EQUIVALENCE ( KCCFS(2453:), KCCFFLD17 )
*
EQUIVALENCE ( KCCFS(2585:), KCCFFNAM18 )
EQUIVALENCE ( KCCFS(2593:), KCCFREM18 )
EQUIVALENCE ( KCCFS(2601:), KCCFLOFL18 )
EQUIVALENCE ( KCCFS(2605:), KCCFFLD18 )
*
EQUIVALENCE ( KCCFS(2737:), KCCFFNAM19 )
EQUIVALENCE ( KCCFS(2745:), KCCFREM19 )
EQUIVALENCE ( KCCFS(2753:), KCCFLOFL19 )
EQUIVALENCE ( KCCFS(2757:), KCCFFLD19 )
*
EQUIVALENCE ( KCCFS(2889:), KCCFFNAM20 )
EQUIVALENCE ( KCCFS(2897:), KCCFREM20 )
EQUIVALENCE ( KCCFS(2905:), KCCFLOFL20 )
EQUIVALENCE ( KCCFS(2909:), KCCFFLD20 )
*
EQUIVALENCE ( KCCFS(3041:), KCCFFNAM21 )
EQUIVALENCE ( KCCFS(3049:), KCCFREM21 )
EQUIVALENCE ( KCCFS(3057:), KCCFLOFL21 )
EQUIVALENCE ( KCCFS(3061:), KCCFFLD21 )
*
EQUIVALENCE ( KCCFS(3193:), KCCFFNAM22 )
EQUIVALENCE ( KCCFS(3201:), KCCFREM22 )
EQUIVALENCE ( KCCFS(3209:), KCCFLOFL22 )
EQUIVALENCE ( KCCFS(3213:), KCCFFLD22 )
*
EQUIVALENCE ( KCCFS(3345:), KCCFFNAM23 )
EQUIVALENCE ( KCCFS(3353:), KCCFREM23 )
EQUIVALENCE ( KCCFS(3361:), KCCFLOFL23 )
EQUIVALENCE ( KCCFS(3365:), KCCFFLD23 )
*
EQUIVALENCE ( KCCFS(3497:), KCCFFNAM24 )
EQUIVALENCE ( KCCFS(3505:), KCCFREM24 )
EQUIVALENCE ( KCCFS(3513:), KCCFLOFL24 )
```

```
*      EQUIVALENCE ( KCCFS(3517:), KCCFFLD24 )
      EQUIVALENCE ( KCCFS(3649:), KCCFFNAM25 )
      EQUIVALENCE ( KCCFS(3657:), KCCFREM25 )
      EQUIVALENCE ( KCCFS(3665:), KCCFLOFL25 )
      EQUIVALENCE ( KCCFS(3669:), KCCFFLD25 )
*
      EQUIVALENCE ( KCCFS(3801:), KCCFFNAM26 )
      EQUIVALENCE ( KCCFS(3809:), KCCFREM26 )
      EQUIVALENCE ( KCCFS(3817:), KCCFLOFL26 )
      EQUIVALENCE ( KCCFS(3821:), KCCFFLD26 )
*
      EQUIVALENCE ( KCCFS(3953:), KCCFFNAM27 )
      EQUIVALENCE ( KCCFS(3961:), KCCFREM27 )
      EQUIVALENCE ( KCCFS(3969:), KCCFLOFL27 )
      EQUIVALENCE ( KCCFS(3973:), KCCFFLD27 )
*
      EQUIVALENCE ( KCCFS(4105:), KCCFFNAM28 )
      EQUIVALENCE ( KCCFS(4113:), KCCFREM28 )
      EQUIVALENCE ( KCCFS(4121:), KCCFLOFL28 )
      EQUIVALENCE ( KCCFS(4125:), KCCFFLD28 )
*
      EQUIVALENCE ( KCCFS(4257:), KCCFFNAM29 )
      EQUIVALENCE ( KCCFS(4265:), KCCFREM29 )
      EQUIVALENCE ( KCCFS(4273:), KCCFLOFL29 )
      EQUIVALENCE ( KCCFS(4277:), KCCFFLD29 )
*
      EQUIVALENCE ( KCCFS(4409:), KCCFFNAM30 )
      EQUIVALENCE ( KCCFS(4417:), KCCFREM30 )
      EQUIVALENCE ( KCCFS(4425:), KCCFLOFL30 )
      EQUIVALENCE ( KCCFS(4429:), KCCFFLD30 )
*
*
      EQUIVALENCE ( KCCFS(4561:), KCCFFNAM31 )
      EQUIVALENCE ( KCCFS(4569:), KCCFREM31 )
      EQUIVALENCE ( KCCFS(4577:), KCCFLOFL31 )
      EQUIVALENCE ( KCCFS(4581:), KCCFFLD31 )
*
      EQUIVALENCE ( KCCFS(4713:), KCCFFNAM32 )
      EQUIVALENCE ( KCCFS(4721:), KCCFREM32 )
      EQUIVALENCE ( KCCFS(4729:), KCCFLOFL32 )
      EQUIVALENCE ( KCCFS(4733:), KCCFFLD32 )
*
      EQUIVALENCE ( KCCFS(4865:), KCCFFNAM33 )
      EQUIVALENCE ( KCCFS(4873:), KCCFREM33 )
      EQUIVALENCE ( KCCFS(4881:), KCCFLOFL33 )
      EQUIVALENCE ( KCCFS(4885:), KCCFFLD33 )
*
      EQUIVALENCE ( KCCFS(5017:), KCCFFNAM34 )
      EQUIVALENCE ( KCCFS(5025:), KCCFREM34 )
      EQUIVALENCE ( KCCFS(5033:), KCCFLOFL34 )
      EQUIVALENCE ( KCCFS(5037:), KCCFFLD34 )
*
      EQUIVALENCE ( KCCFS(5169:), KCCFFNAM35 )
      EQUIVALENCE ( KCCFS(5177:), KCCFREM35 )
      EQUIVALENCE ( KCCFS(5185:), KCCFLOFL35 )
      EQUIVALENCE ( KCCFS(5189:), KCCFFLD35 )
*
      EQUIVALENCE ( KCCFS(5321:), KCCFFNAM36 )
      EQUIVALENCE ( KCCFS(5329:), KCCFREM36 )
```

```

      EQUIVALENCE ( KCCFS(5337:), KCCFLOFL36 )
      EQUIVALENCE ( KCCFS(5341:), KCCFFLD36 )
*
      EQUIVALENCE ( KCCFS(5473:), KCCFFNAM37 )
      EQUIVALENCE ( KCCFS(5481:), KCCFREM37 )
      EQUIVALENCE ( KCCFS(5489:), KCCFLOFL37 )
      EQUIVALENCE ( KCCFS(5493:), KCCFFLD37 )
*
      EQUIVALENCE ( KCCFS(5625:), KCCFFNAM38 )
      EQUIVALENCE ( KCCFS(5633:), KCCFREM38 )
      EQUIVALENCE ( KCCFS(5641:), KCCFLOFL38 )
      EQUIVALENCE ( KCCFS(5645:), KCCFFLD38 )
*
      EQUIVALENCE ( KCCFS(5777:), KCCFFNAM39 )
      EQUIVALENCE ( KCCFS(5785:), KCCFREM39 )
      EQUIVALENCE ( KCCFS(5793:), KCCFLOFL39 )
      EQUIVALENCE ( KCCFS(5797:), KCCFFLD39 )
*
      EQUIVALENCE ( KCCFS(5929:), KCCFFNAM40 )
      EQUIVALENCE ( KCCFS(5937:), KCCFREM40 )
      EQUIVALENCE ( KCCFS(5945:), KCCFLOFL40 )
      EQUIVALENCE ( KCCFS(5949:), KCCFFLD40 )
*
*
      EQUIVALENCE ( KCCFS(6081:), KCCFFNAM41 )
      EQUIVALENCE ( KCCFS(6089:), KCCFREM41 )
      EQUIVALENCE ( KCCFS(6097:), KCCFLOFL41 )
      EQUIVALENCE ( KCCFS(6101:), KCCFFLD41 )
*
      EQUIVALENCE ( KCCFS(6233:), KCCFFNAM42 )
      EQUIVALENCE ( KCCFS(6241:), KCCFREM42 )
      EQUIVALENCE ( KCCFS(6249:), KCCFLOFL42 )
      EQUIVALENCE ( KCCFS(6253:), KCCFFLD42 )
*
      EQUIVALENCE ( KCCFS(6385:), KCCFFNAM43 )
      EQUIVALENCE ( KCCFS(6393:), KCCFREM43 )
      EQUIVALENCE ( KCCFS(6401:), KCCFLOFL43 )
      EQUIVALENCE ( KCCFS(6405:), KCCFFLD43 )
*
      EQUIVALENCE ( KCCFS(6537:), KCCFFNAM44 )
      EQUIVALENCE ( KCCFS(6545:), KCCFREM44 )
      EQUIVALENCE ( KCCFS(6553:), KCCFLOFL44 )
      EQUIVALENCE ( KCCFS(6557:), KCCFFLD44 )
*
      EQUIVALENCE ( KCCFS(6689:), KCCFFNAM45 )
      EQUIVALENCE ( KCCFS(6697:), KCCFREM45 )
      EQUIVALENCE ( KCCFS(6705:), KCCFLOFL45 )
      EQUIVALENCE ( KCCFS(6709:), KCCFFLD45 )
*
      EQUIVALENCE ( KCCFS(6841:), KCCFFNAM46 )
      EQUIVALENCE ( KCCFS(6849:), KCCFREM46 )
      EQUIVALENCE ( KCCFS(6857:), KCCFLOFL46 )
      EQUIVALENCE ( KCCFS(6861:), KCCFFLD46 )
*
      EQUIVALENCE ( KCCFS(6993:), KCCFFNAM47 )
      EQUIVALENCE ( KCCFS(7001:), KCCFREM47 )
      EQUIVALENCE ( KCCFS(7009:), KCCFLOFL47 )
      EQUIVALENCE ( KCCFS(7013:), KCCFFLD47 )
*
      EQUIVALENCE ( KCCFS(7145:), KCCFFNAM48 )

```

```
      EQUIVALENCE ( KCCFS(7153:), KCCFREM48 )
      EQUIVALENCE ( KCCFS(7161:), KCCFLOFL48 )
      EQUIVALENCE ( KCCFS(7165:), KCCFFLD48 )
*
      EQUIVALENCE ( KCCFS(7297:), KCCFFNAM49 )
      EQUIVALENCE ( KCCFS(7305:), KCCFREM49 )
      EQUIVALENCE ( KCCFS(7313:), KCCFLOFL49 )
      EQUIVALENCE ( KCCFS(7317:), KCCFFLD49 )
*
      EQUIVALENCE ( KCCFS(7449:), KCCFFNAM50 )
      EQUIVALENCE ( KCCFS(7457:), KCCFREM50 )
      EQUIVALENCE ( KCCFS(7465:), KCCFLOFL50 )
      EQUIVALENCE ( KCCFS(7469:), KCCFFLD50 )
*
*
*****
```

## 4.5 Data structure KCDADF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
**
**
**      Return Information of DADM Call                                *
**
**      Include for FOR1 : KCDADF                                    *
**
*****
*
*      CHARACTER  KCDADF * (36)
*
*      CHARACTER  KCDAGUS * 8
*
*      CHARACTER  KCDADPID * 8
*
*      CHARACTER  KCDAGTIM * (9)
*
*      CHARACTER  KCDAGDOY * 3
*
*      CHARACTER  KCDAGHR * 2
*
*      CHARACTER  KCDAGMIN * 2
*
*      CHARACTER  KCDAGSEC * 2
*
*      CHARACTER  KCDASTIM * (9)
*
*      CHARACTER  KCDASDOY * 3
*
*      CHARACTER  KCDASHR * 2
*
*      CHARACTER  KCDASMIN * 2
*
*      CHARACTER  KCDASSEC * 2
*
*      CHARACTER  KCDAPMSG * 1
*
*      CHARACTER  KCDANMSG * 1
*
*
*      CHARACTER  KCDADPID * 8
*
*      CHARACTER  DPUT id
*
*      generation time of message :
*
*      day of year
*
*      hour
*
*      minute
*
*      second
*
*      desired start time (DPUT) :
*
*      day of year
*
*      hour
*
*      minute
*
*      second
*
*      positive acknowl. job
*
*      negative acknowl. job
*
*
*      EQUIVALENCE ( KCDADF(1:) , KCDAGUS )
*      EQUIVALENCE ( KCDADF(9:) , KCDADPID )
*      EQUIVALENCE ( KCDADF(17:) , KCDAGTIM , KCDAGDOY )
*      EQUIVALENCE ( KCDADF(20:) , KCDAGHR )
*      EQUIVALENCE ( KCDADF(22:) , KCDAGMIN )
*      EQUIVALENCE ( KCDADF(24:) , KCDAGSEC )
*      EQUIVALENCE ( KCDADF(26:) , KCDASTIM , KCDASDOY )
*      EQUIVALENCE ( KCDADF(29:) , KCDASHR )

```

```
EQUIVALENCE ( KCDADF(31:), KCDASMIN )  
EQUIVALENCE ( KCDADF(33:), KCDASSEC )  
EQUIVALENCE ( KCDADF(35:), KCDAPMSG )  
EQUIVALENCE ( KCDADF(36:), KCDANMSG )
```

```
*
```

```
*****
```

## 4.6 Data structure KCDFF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
**
**
**      Device Features                                             *
**
**      Include for FOR1 : KCDFF                                     *
**
*****
*
*      CHARACTER  KCDFF * (14)
*
*      INTEGER   KCREPL * 2   /1/      screen output functions :
*
*      INTEGER   KCERAS * 2   /2/      clear screen + display format
*
*      INTEGER   KCALARM * 2  /4/      erase unprotected fields
*
*      INTEGER   KCREPR * 2   /8/      BEL function
*
*      INTEGER   KCRESTRT * 2  /1/     output on local printer
*
*      INTEGER   KCEXTEND * 2  /8192/  screen restart with PEND RS
*
*      INTEGER   KCCARDRD * 2  /16384/ extended line mode
*
*
*      next input from card reader
*
*
*      EQUIVALENCE (KCDFF(1:), KCREPL)
*      EQUIVALENCE (KCDFF(3:), KCERAS)
*      EQUIVALENCE (KCDFF(5:), KCALARM)
*      EQUIVALENCE (KCDFF(7:), KCREPR)
*      EQUIVALENCE (KCDFF(9:), KCRESTRT)
*      EQUIVALENCE (KCDFF(11:), KCEXTEND)
*      EQUIVALENCE (KCDFF(13:), KCCARDRD)
*
*****

```



## 4.7 Data structure KCINFF

```

*
*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0         +***
*
*****
*
* Return Information of INFO Call                                     *
*
* Include for FOR1 : KCINFF                                         *
*
*****
*
*      CHARACTER  KCINFF * (65)
*
*                                     maximum size of return info
*
*      CHARACTER  KCRETINF * (65)
*
*-----*
* return information for  KCOM = DT                                  *
*-----*
*
*      CHARACTER  KCDTTM * (30)
*
*      CHARACTER  KCAS * (15)
*                                     date/time of application start
*      CHARACTER  KCASDATE * (9)
*                                     date :
*      CHARACTER  KCASDAY * 2
*                                     day
*      CHARACTER  KCASMON * 2
*                                     month
*      CHARACTER  KCASYEAR * 2
*                                     year
*      CHARACTER  KCASDOY * 3
*                                     day of year
*      CHARACTER  KCASTIME * (6)
*                                     time :
*      CHARACTER  KCASHOUR * 2
*                                     hour
*      CHARACTER  KCASMIN * 2
*                                     minute
*      CHARACTER  KCASSEC * 2
*                                     second
*-----*
*
*      CHARACTER  KCPS * (15)
*
*      CHARACTER  KCPSDATE * (9)
*                                     date/time of program start :
*      CHARACTER  KCPSDAY * 2
*                                     date :
*      CHARACTER  KCPSMON * 2
*                                     day

```

```

*          CHARACTER          KCPSYEAR * 2          month
*
*          CHARACTER          KCPSDOY * 3          year
*
*          CHARACTER          KCPSTIME * (6)       day of year
*
*          CHARACTER          KCPSHOUR * 2         time :
*
*          CHARACTER          KCPSMIN * 2         hour
*
*          CHARACTER          KCPSSSEC * 2        minute
*
*          CHARACTER          KCPSSSEC * 2        second
*
*-----*
* return information for KCOM = SI                *
*-----*
*
*          CHARACTER KCSYSINF * (49)
*
*          CHARACTER          KCAPPLNM * 8         system information :
*
*          CHARACTER          KCHOSTNM * 8         application name
*
*          CHARACTER          KCPTRMNM * 8         host name
*
*          CHARACTER          KCPRONM * 8         PTERM name
*
*          CHARACTER          KCBCAPNM * 8         processor name
*
*          CHARACTER          KCVERS * 6           BCAP application name
*
*          INTEGER           KCIVER * 2           UTM- Version
*
*          CHARACTER          KCIVAR * 1           Interface-version
*
*          CHARACTER          KCIVAR * 1           BS2000 or Sinix
*
*-----*
* return information for KCOM = PC                *
*-----*
*
*          CHARACTER          KCPREINF * (39)
*
*          CHARACTER          KCPFN * 8           predecessor information :
*
*          CHARACTER          KCPNXTAC * 8         format name
*
*          CHARACTER          KCPCVTAC * 8         next tac
*
*          CHARACTER          KCPLDATE * (9)       conversation tac
*
*          CHARACTER          KCPLDAY * 2         date of last program run :
*
*          CHARACTER          KCPLDAY * 2         day
*
*          CHARACTER          KCPLMON * 2         month
*
*          CHARACTER          KCPLYEAR * 2        year
*
*          CHARACTER          KCPLDOY * 3        day of year
*
*          CHARACTER          KCPLTIME * (6)

```

```

*           CHARACTER      KCPLHOUR * 2           time of last program run :
*           CHARACTER      KCPLMIN * 2           hour
*           CHARACTER      KCPLSEC * 2           minute
*           CHARACTER      KCPLSEC * 2           second
* _____*
* return information for KCOM = LO *
* _____*
*
* CHARACTER KCINFLOC * (65)           locale information
* CHARACTER KCLTLOC * (12)           locale of specified lterm
* CHARACTER KCLTLANG * 2             language id
* CHARACTER KCLTTERR * 2             territory id
* CHARACTER KCLTCCSN * 8             coded character set name
* CHARACTER KCAPLOC * (12)           locale of application
* CHARACTER KCAPLANG * 2             language id
* CHARACTER KCAPTERR * 2             territory id
* CHARACTER KCAPCCSN * 8             coded character set nam
* CHARACTER KCCSINFO * (25)         info about xhcs support
* CHARACTER KCDEFCCS * 8             default ccs of system/userid
* INTEGER   KCCCSNO * 1             number of supported ccs
* INTEGER   KCVAR1 * 1             iso var no of 1.  supp. ccs
* INTEGER   KCVAR2 * 1             iso var no of 2.  supp. ccs
* INTEGER   KCVAR3 * 1             iso var no of 3.  supp. ccs
* INTEGER   KCVAR4 * 1             iso var no of 4.  supp. ccs
* INTEGER   KCVAR5 * 1             iso var no of 5.  supp. ccs
* INTEGER   KCVAR6 * 1             iso var no of 6.  supp. ccs
* INTEGER   KCVAR7 * 1             iso var no of 7.  supp. ccs
* INTEGER   KCVAR8 * 1             iso var no of 8.  supp. ccs
* INTEGER   KCVAR9 * 1             iso var no of 9.  supp. ccs
* INTEGER   KCVAR10 * 1            iso var no of 10. supp. ccs
* INTEGER   KCVAR11 * 1            iso var no of 11. supp. ccs
* INTEGER   KCVAR12 * 1            iso var no of 12. supp. ccs
* INTEGER   KCVAR13 * 1

```



```
EQUIVALENCE ( KCINFF(59:), KCVAR10 )
EQUIVALENCE ( KCINFF(60:), KCVAR11 )
EQUIVALENCE ( KCINFF(61:), KCVAR12 )
EQUIVALENCE ( KCINFF(62:), KCVAR13 )
EQUIVALENCE ( KCINFF(63:), KCVAR14 )
EQUIVALENCE ( KCINFF(64:), KCVAR15 )
EQUIVALENCE ( KCINFF(65:), KCVAR16 )
*
*****
*
```

## 4.8 Data structure KCINIF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1993   +***
**                                     ALL RIGHTS RESERVED           +***
**                                     +***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM 4.0         +***
**                                     +***
*****
*
* Return Information of INIT PU Call                                *
*                                                                     *
* Include for FOR1 : KCINIF                                        *
*                                                                     *
*****
*
*      CHARACTER  KCINIF * (156)                                     *
*                                                                     *
*      _____ maximum size of structure _____ *
*      input information for KCOM = PU                               *
*      _____ *
*
*      CHARACTER  KCINPUT * (16)
*
*      INTEGER    KCINIVER * 2                                     *
*                                                                     *
*      _____ interface version (1) _____ *
*      CHARACTER  KCDATE * 1                                     *
*                                                                     *
*      _____ date and time info ( y / n ) _____ *
*      CHARACTER  KCAPPL * 1                                     *
*                                                                     *
*      _____ application ( y / n ) _____ *
*      CHARACTER  KCLOCALE * 1                                   *
*                                                                     *
*      _____ locale info ( y / n ) _____ *
*      CHARACTER  KCOSITP * 1                                   *
*                                                                     *
*      _____ OSI TP ( y / n ) _____ *
*      _____ *
*      output information for KCOM = PU                               *
*      _____ *
*
*      CHARACTER  KCOUTPUT * (140)
*
*      _____ *
*      _____ general information _____ *
*      _____ *
*
*      INTEGER    KCGPAB * 2                                     *
*                                                                     *
*      _____ generated maximal lth of spab _____ *
*      INTEGER    KCGNB * 2                                     *
*                                                                     *
*      _____ generated maximal lth of nb _____ *
*      _____ *
*      _____ date and time information _____ *
*      _____ *
*
*      CHARACTER  KCDATTIM * (48)
*
*      CHARACTER  KCADTTM * (18)
*
*      _____ date/time of application start _____

```

*	CHARACTER	KCADATE * (11)	date :
*	CHARACTER	KCADAY * 2	day
*	CHARACTER	KCAMONTH * 2	month
*	CHARACTER	KCAYEAR * 4	year
*	CHARACTER	KCADOY * 3	day of year
*	CHARACTER	KCATIME * (6)	time :
*	CHARACTER	KCAHOUR * 2	hour
*	CHARACTER	KCAMIN * 2	minute
*	CHARACTER	KCASEC * 2	second
*	CHARACTER	KCASEAS * 1	season (normal (w)/ summer (s))
*	CHARACTER	KCPDTM * (18)	date/time of program start
*	CHARACTER	KCPDATE * (11)	date :
*	CHARACTER	KCPDAY * 2	day
*	CHARACTER	KCPMONTH * 2	month
*	CHARACTER	KCPYEAR * 4	year
*	CHARACTER	KCPDOY * 3	day of year
*	CHARACTER	KCPTIME * (6)	time :
*	CHARACTER	KCPHOUR * 2	hour
*	CHARACTER	KCPMIN * 2	minute
*	CHARACTER	KCPSEC * 2	second
*	CHARACTER	KCPSEAS * 1	season (normal (w)/ summer (s))
*	CHARACTER	KCTMZONE * (12)	time zone
<hr/>			
*	application information		*
<hr/>			
*	CHARACTER	KCAPINF * (50)	system information :
*	CHARACTER	KCNMAPPL * 8	application name
*	CHARACTER	KCNMHOST * 8	host name
*	CHARACTER	KCNMPTRM * 8	PTERM name
*	CHARACTER	KCNMPRO * 8	processor name
*	CHARACTER	KCNMBCAP * 8	BCAP application name

```

*      CHARACTER    KCUTMVER    * 6
*
*      INTEGER      KCIFVER     * 2
*
*      CHARACTER    KCUTMVAR    * 1
*
*-----*
*      locale information
*-----*
*
*      CHARACTER    KCLOCINF    * (22)
*
*      CHARACTER    KCUSLOC     * (12)
*
*      CHARACTER    KCUSLANG    * 2
*
*      CHARACTER    KCUSLERR    * 2
*
*      CHARACTER    KCUSCCSN    * 8
*
*      CHARACTER    KCXHCS     * (9)
*
*      CHARACTER    KCCURCCS    * 8
*
*      INTEGER      KCDEVCAP    * 1
*
*-----*
*      OSI TP information
*-----*
*
*      CHARACTER    KCOSIINF    * (8)
*
*      CHARACTER    KCFUPOL     * 1
*
*      CHARACTER    KCFUHS     * 1
*
*      CHARACTER    KCFUCOM     * 1
*
*      CHARACTER    KCFUCHN     * 1
*
*      CHARACTER    KCENDTA     * 1
*
*      CHARACTER    KCSSEND     * 1
*
*
*
*
*      EQUIVALENCE ( KCINIF(1:) , KCINPUT , KCINIVER)
*      EQUIVALENCE ( KCINIF(3:) , KCDATE )
*      EQUIVALENCE ( KCINIF(4:) , KCAPPL )
*      EQUIVALENCE ( KCINIF(5:) , KCLOCALE )
*      EQUIVALENCE ( KCINIF(6:) , KCOSITP )
*      EQUIVALENCE ( KCINIF(17:) , KCOUTPUT , KCGPAB)
*      EQUIVALENCE ( KCINIF(19:) , KCGNB )
*      EQUIVALENCE ( KCINIF(21:) , KCDATTIM, KCADTTM , KCADATE, KCADAY)
*      EQUIVALENCE ( KCINIF(23:) , KCAMONTH )
*      EQUIVALENCE ( KCINIF(25:) , KCAYEAR )
*      EQUIVALENCE ( KCINIF(29:) , KCADYOY )
*      EQUIVALENCE ( KCINIF(32:) , KCATIME , KCAHOUR )
*      EQUIVALENCE ( KCINIF(34:) , KCASMIN )

```



```
EQUIVALENCE ( KCINIF(36:), KCASSEC )
EQUIVALENCE ( KCINIF(38:), KCASEAS )
EQUIVALENCE ( KCINIF(39:), KCPDTTM , KCPDATE, KCPDAY )
EQUIVALENCE ( KCINIF(41:), KCPMONTH )
EQUIVALENCE ( KCINIF(43:), KCPYEAR )
EQUIVALENCE ( KCINIF(47:), KCPDOY )
EQUIVALENCE ( KCINIF(50:), KCPTIME , KCPHOUR )
EQUIVALENCE ( KCINIF(52:), KCPMIN )
EQUIVALENCE ( KCINIF(54:), KCPSEC )
EQUIVALENCE ( KCINIF(56:), KCPSEAS )
EQUIVALENCE ( KCINIF(57:), KCTMZONE )
EQUIVALENCE ( KCINIF(69:), KCAPINF , KCNMAPPL )
EQUIVALENCE ( KCINIF(77:), KCNMHOST )
EQUIVALENCE ( KCINIF(85:), KCNMPTRM )
EQUIVALENCE ( KCINIF(93:), KCNMPRO )
EQUIVALENCE ( KCINIF(101:), KCNBCAP )
EQUIVALENCE ( KCINIF(109:), KCUTMVER )
EQUIVALENCE ( KCINIF(115:), KCIFVER )
EQUIVALENCE ( KCINIF(117:), KCUTMVAR )
EQUIVALENCE ( KCINIF(119:), KCLOCINF , KCUSLOC , KCUSLANG)
EQUIVALENCE ( KCINIF(121:), KCUSTERR )
EQUIVALENCE ( KCINIF(123:), KCUSCCSN )
EQUIVALENCE ( KCINIF(139:), KCXHCS , KCCURCCS )
EQUIVALENCE ( KCINIF(147:), KCDEVCAP )
EQUIVALENCE ( KCINIF(149:), KCOSIINF , KCFUPOL )
EQUIVALENCE ( KCINIF(150:), KCFUHSH )
EQUIVALENCE ( KCINIF(151:), KCFUCOM )
EQUIVALENCE ( KCINIF(152:), KCFUCHN )
EQUIVALENCE ( KCINIF(153:), KCENDTA )
EQUIVALENCE ( KCINIF(154:), KCSSEND )
```

\*

\*\*\*\*\*

\*

## 4.9 Data structure KCINPF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
*
*****
*
*   Parameter Area for input exit                                     *
*
*   Include for FOR1 : KCINPF                                       *
*
*****
*
*   CHARACTER  KCINPF * (140)
*
*   CHARACTER  KCIFCH * 8
*
*                                     first 8 characters of input
*                                     message
*
*   CHARACTER  KCIFN * 8
*
*                                     format name
*
*   CHARACTER  KCICVTAC * 8
*
*                                     conversation tac
*
*   CHARACTER  KCICVST * 2
*
*                                     conversation state
*
*   INTEGER    KCIFKEY * 2
*
*                                     F-key
*
*   INTEGER    KCIKKEY * 2
*
*                                     K-key
*
*   CHARACTER  KCICFINF * 2
*
*                                     control field information
*
*   CHARACTER  KCILTERM * 8
*
*                                     current lterm
*
*   CHARACTER  KCIUSER * 8
*
*                                     current user
*
*   CHARACTER  KCINTAC * 8
*
*                                     next tac
*
*   CHARACTER  KCINCMD * 8
*
*                                     next command
*
*   CHARACTER  KCICCD * 2
*
*                                     continuation command
*
*   CHARACTER  KCICUT * 1
*
*                                     cut tac (y/n)
*
*   CHARACTER  KCIERRCD * 4
*
*                                     error code
*
*
*   EQUIVALENCE ( KCINPF(1:) , KCIFCH   )
*   EQUIVALENCE ( KCINPF(9:) , KCIFN   )
*   EQUIVALENCE ( KCINPF(17:) , KCICVTAC )
*   EQUIVALENCE ( KCINPF(25:) , KCICVST )
*   EQUIVALENCE ( KCINPF(27:) , KCIFKEY )
*   EQUIVALENCE ( KCINPF(29:) , KCIKKEY )
*   EQUIVALENCE ( KCINPF(31:) , KCICFINF )

```

```
EQUIVALENCE ( KCINPF(33:), KCI TERM )
EQUIVALENCE ( KCINPF(41:), KCI USER )
EQUIVALENCE ( KCINPF(81:), KCINTAC , KCINCMD )
EQUIVALENCE ( KCINPF(89:), KCICCD )
EQUIVALENCE ( KCINPF(91:), KCICUT )
EQUIVALENCE ( KCINPF(93:), KCIERRCD )
```

\*

\*\*\*\*\*

\*

## 4.10 Data structure KCMSGF

```

*****
***                                     ***
***      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992      ***
***                                     ALL RIGHTS RESERVED                ***
***                                     ***
*****
***      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM ...           ***
*****
***      Layout of UTM-messages                UTM (BS2000)  V04.0      ***
***                                     KCMSGF                20.06.1996 ***
*****
*
*      CHARACTER KCMSGF * (176)
*
*      CHARACTER   MSGKOPF * (24)
*                                     MESSAGE HEADER
*      CHARACTER   KCMFIL01 * 1
*                                     FILLER
*      CHARACTER   MSGNR * 4
*                                     MESSAGE_NUMBER
*      CHARACTER   KCMFIL02 * 1
*                                     FILLER
*      CHARACTER   MSGDATE * 11
*                                     DATE OF ORIGIN
*                                     (MM/DD/YYJJJ)
*      CHARACTER   KCMFIL03 * 1
*                                     FILLER
*      CHARACTER   MSGTIME * 6
*                                     TIME OF ORIGIN (HHMMSS)
*      CHARACTER   MSGYEAR * 4
*                                     YEAR OF ORIGIN (YYYY)
*****
*      INSERTS OF MESSAGES                                     ***
*****
*      CHARACTER KXXX * (152)
*
*      COMMON /KCMSGF/ KCMFIL01, MSGNR, KCMFIL02, MSGDATE
*      COMMON /KCMSGF/ KCMFIL03, MSGTIME, KXXX
*
*      EQUIVALENCE (KCMSGF(1:), MSGKOPF , KCMFIL01)
*      EQUIVALENCE (KCMSGF(25:), KXXX)
*****
*
*      CHARACTER K001PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K001PTRM)
*                                     PTERM NAME
*      CHARACTER K001PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K001PRNM)
*                                     PROCESSOR NAME
*      CHARACTER K001BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K001BCAP)
*      CHARACTER K001LTRM * (008)
*                                     BCAM APPLICATION NAME

```

```

      EQUIVALENCE (KXXX (25:), K001LTRM)
*
      CHARACTER K001APPL * (008)
      EQUIVALENCE (KXXX (33:), K001APPL)
*
      CHARACTER K001TEXT * (112)
      EQUIVALENCE (KXXX (41:), K001TEXT)
*****
      CHARACTER K002PTRM * (008)
      EQUIVALENCE (KXXX (1:), K002PTRM)
*
      CHARACTER K002PRNM * (008)
      EQUIVALENCE (KXXX (9:), K002PRNM)
*
      CHARACTER K002BCAP * (008)
      EQUIVALENCE (KXXX (17:), K002BCAP)
*
      CHARACTER K002LTRM * (008)
      EQUIVALENCE (KXXX (25:), K002LTRM)
*
      CHARACTER K002APPL * (008)
      EQUIVALENCE (KXXX (33:), K002APPL)
*
      CHARACTER K002TEXT * (112)
      EQUIVALENCE (KXXX (41:), K002TEXT)
*****
      CHARACTER K003PTRM * (008)
      EQUIVALENCE (KXXX (1:), K003PTRM)
*
      CHARACTER K003PRNM * (008)
      EQUIVALENCE (KXXX (9:), K003PRNM)
*
      CHARACTER K003BCAP * (008)
      EQUIVALENCE (KXXX (17:), K003BCAP)
*
      CHARACTER K003LTRM * (008)
      EQUIVALENCE (KXXX (25:), K003LTRM)
*
      CHARACTER K003CMD * (008)
      EQUIVALENCE (KXXX (33:), K003CMD )
*
      CHARACTER K003TEXT * (112)
      EQUIVALENCE (KXXX (41:), K003TEXT)
*****
      CHARACTER K004PTRM * (008)
      EQUIVALENCE (KXXX (1:), K004PTRM)
*
      CHARACTER K004PRNM * (008)
      EQUIVALENCE (KXXX (9:), K004PRNM)
*
      CHARACTER K004BCAP * (008)
      EQUIVALENCE (KXXX (17:), K004BCAP)
*
      CHARACTER K004LTRM * (008)
      EQUIVALENCE (KXXX (25:), K004LTRM)
*
      CHARACTER K004USER * (008)
      EQUIVALENCE (KXXX (33:), K004USER)
*
      CHARACTER K004TEXT * (112)

```

```

EQUIVALENCE (KXXX (41:), K004TEXT)
*****
CHARACTER K005PTRM * (008)
EQUIVALENCE (KXXX (1:), K005PTRM)
*
CHARACTER K005PRNM * (008)
EQUIVALENCE (KXXX (9:), K005PRNM)
*
CHARACTER K005BCAP * (008)
EQUIVALENCE (KXXX (17:), K005BCAP)
*
CHARACTER K005LTRM * (008)
EQUIVALENCE (KXXX (25:), K005LTRM)
*
CHARACTER K005USER * (008)
EQUIVALENCE (KXXX (33:), K005USER)
*
CHARACTER K005TEXT * (112)
EQUIVALENCE (KXXX (41:), K005TEXT)
*****
CHARACTER K006PTRM * (008)
EQUIVALENCE (KXXX (1:), K006PTRM)
*
CHARACTER K006PRNM * (008)
EQUIVALENCE (KXXX (9:), K006PRNM)
*
CHARACTER K006BCAP * (008)
EQUIVALENCE (KXXX (17:), K006BCAP)
*
CHARACTER K006LTRM * (008)
EQUIVALENCE (KXXX (25:), K006LTRM)
*
CHARACTER K006USER * (008)
EQUIVALENCE (KXXX (33:), K006USER)
*
CHARACTER K006TEXT * (112)
EQUIVALENCE (KXXX (41:), K006TEXT)
*****
CHARACTER K007PTRM * (008)
EQUIVALENCE (KXXX (1:), K007PTRM)
*
CHARACTER K007PRNM * (008)
EQUIVALENCE (KXXX (9:), K007PRNM)
*
CHARACTER K007BCAP * (008)
EQUIVALENCE (KXXX (17:), K007BCAP)
*
CHARACTER K007LTRM * (008)
EQUIVALENCE (KXXX (25:), K007LTRM)
*
CHARACTER K007USER * (008)
EQUIVALENCE (KXXX (33:), K007USER)
*
CHARACTER K007TEXT * (112)
EQUIVALENCE (KXXX (41:), K007TEXT)
*****
CHARACTER K008PTRM * (008)
EQUIVALENCE (KXXX (1:), K008PTRM)
*
CHARACTER K008PRNM * (008)
EQUIVALENCE (KXXX (9:), K008PRNM)
*

```

```

      EQUIVALENCE (KXXX (9:), K008PRNM)
*
      CHARACTER K008BCAP * (008)          PROCESSOR NAME
      EQUIVALENCE (KXXX (17:), K008BCAP)
*
      CHARACTER K008LTRM * (008)         BCAM APPLICATION NAME
      EQUIVALENCE (KXXX (25:), K008LTRM)
*
      CHARACTER K008USER * (008)         LTERM NAME
      EQUIVALENCE (KXXX (33:), K008USER)
*
      CHARACTER K008TEXT * (112)        USER/LSES/OSI-ASS NAME
      EQUIVALENCE (KXXX (41:), K008TEXT)
*****
      CHARACTER K009PTRM * (008)
*
      EQUIVALENCE (KXXX (1:), K009PTRM)  PTERM NAME
      CHARACTER K009PRNM * (008)
*
      EQUIVALENCE (KXXX (9:), K009PRNM)  PROCESSOR NAME
      CHARACTER K009BCAP * (008)
*
      EQUIVALENCE (KXXX (17:), K009BCAP) BCAM APPLICATION NAME
      CHARACTER K009LTRM * (008)
*
      EQUIVALENCE (KXXX (25:), K009LTRM) LTERM NAME
      CHARACTER K009USER * (008)
*
      EQUIVALENCE (KXXX (33:), K009USER) USER/LSES/OSI-ASS NAME
      CHARACTER K009TAC * (008)
*
      EQUIVALENCE (KXXX (41:), K009TAC ) TRANSACTION CODE
      CHARACTER K009TEXT * (104)
*
      EQUIVALENCE (KXXX (49:), K009TEXT)
*****
      CHARACTER K010PTRM * (008)
*
      EQUIVALENCE (KXXX (1:), K010PTRM)  PTERM NAME
      CHARACTER K010PRNM * (008)
*
      EQUIVALENCE (KXXX (9:), K010PRNM)  PROCESSOR NAME
      CHARACTER K010BCAP * (008)
*
      EQUIVALENCE (KXXX (17:), K010BCAP) BCAM APPLICATION NAME
      CHARACTER K010LTRM * (008)
*
      EQUIVALENCE (KXXX (25:), K010LTRM) LTERM NAME
      CHARACTER K010USER * (008)
*
      EQUIVALENCE (KXXX (33:), K010USER) USER/LSES/OSI-ASS NAME
      CHARACTER K010TAC * (008)
*
      EQUIVALENCE (KXXX (41:), K010TAC ) TRANSACTION CODE
      CHARACTER K010TEXT * (104)
*
      EQUIVALENCE (KXXX (49:), K010TEXT)
*****
      CHARACTER K011PTRM * (008)
*
      EQUIVALENCE (KXXX (1:), K011PTRM)  PTERM NAME
      CHARACTER K011PRNM * (008)

```

```

EQUIVALENCE (KXXX (9:), K011PRNM)
*
CHARACTER K011BCAP * (008)          PROCESSOR NAME
EQUIVALENCE (KXXX (17:), K011BCAP)
*
CHARACTER K011LTRM * (008)          BCAM APPLICATION NAME
EQUIVALENCE (KXXX (25:), K011LTRM)
*
CHARACTER K011USER * (008)          LTERM NAME
EQUIVALENCE (KXXX (33:), K011USER)
*
CHARACTER K011ATAC * (008)          USER/LSES/OSI-ASS NAME
EQUIVALENCE (KXXX (41:), K011ATAC)
*
CHARACTER K011TEXT * (104)          ASYNCHRONOUS TAC
EQUIVALENCE (KXXX (49:), K011TEXT)
*****
CHARACTER K013PTRM * (008)
EQUIVALENCE (KXXX (1:), K013PTRM)
*
CHARACTER K013PRNM * (008)          PTERM NAME
EQUIVALENCE (KXXX (9:), K013PRNM)
*
CHARACTER K013BCAP * (008)          PROCESSOR NAME
EQUIVALENCE (KXXX (17:), K013BCAP)
*
CHARACTER K013LTRM * (008)          BCAM APPLICATION NAME
EQUIVALENCE (KXXX (25:), K013LTRM)
*
CHARACTER K013CMD * (008)           LTERM NAME
EQUIVALENCE (KXXX (33:), K013CMD )
*
CHARACTER K013TEXT * (112)          COMMAND NAME
EQUIVALENCE (KXXX (41:), K013TEXT)
*****
CHARACTER K014PTRM * (008)
EQUIVALENCE (KXXX (1:), K014PTRM)
*
CHARACTER K014PRNM * (008)          PTERM NAME
EQUIVALENCE (KXXX (9:), K014PRNM)
*
CHARACTER K014BCAP * (008)          PROCESSOR NAME
EQUIVALENCE (KXXX (17:), K014BCAP)
*
CHARACTER K014LTRM * (008)          BCAM APPLICATION NAME
EQUIVALENCE (KXXX (25:), K014LTRM)
*
CHARACTER K014USER * (008)          LTERM NAME
EQUIVALENCE (KXXX (33:), K014USER)
*
CHARACTER K014TEXT * (112)          USER/LSES/OSI-ASS NAME
EQUIVALENCE (KXXX (41:), K014TEXT)
*****
CHARACTER K015PTRM * (008)
EQUIVALENCE (KXXX (1:), K015PTRM)
*
CHARACTER K015PRNM * (008)          PTERM NAME
EQUIVALENCE (KXXX (9:), K015PRNM)
*
CHARACTER K015BCAP * (008)          PROCESSOR NAME

```



```

      EQUIVALENCE (KXXX (17:), K015BCAP)
*      CHARACTER K015LTRM * (008)          BCAM APPLICATION NAME
      EQUIVALENCE (KXXX (25:), K015LTRM)
*      CHARACTER K015USER * (008)          LTERM NAME
      EQUIVALENCE (KXXX (33:), K015USER)
*      CHARACTER K015TAC * (008)          USER/LSES/OSI-ASS NAME
      EQUIVALENCE (KXXX (41:), K015TAC )
*      CHARACTER K015FORM * (008)         TRANSACTION CODE
      EQUIVALENCE (KXXX (49:), K015FORM)
*      CHARACTER K015RCDC * (004)         FORMAT NAME (FOR K015
*                                         ONLY)
      EQUIVALENCE (KXXX (57:), K015RCDC)
*      CHARACTER K015RCF2 * (004)        KRCDC
      EQUIVALENCE (KXXX (61:), K015RCF2)
*      CHARACTER K015TEXT * (088)        SECONDARY FHS/VTSU RET
*                                         CODE
      EQUIVALENCE (KXXX (65:), K015TEXT)
*****
      CHARACTER K016PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K016PTRM)  PTRM NAME
      CHARACTER K016PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K016PRNM)  PROCESSOR NAME
      CHARACTER K016BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K016BCAP) BCAM APPLICATION NAME
      CHARACTER K016LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K016LTRM) LTERM NAME
      CHARACTER K016USER * (008)
*      EQUIVALENCE (KXXX (33:), K016USER) USER/LSES/OSI-ASS NAME
      CHARACTER K016TEXT * (112)
*****
      EQUIVALENCE (KXXX (41:), K016TEXT)
      CHARACTER K017PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K017PTRM)  PTRM NAME
      CHARACTER K017PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K017PRNM)  PROCESSOR NAME
      CHARACTER K017BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K017BCAP) BCAM APPLICATION NAME
      CHARACTER K017LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K017LTRM) LTERM NAME
      CHARACTER K017USER * (008)
*      EQUIVALENCE (KXXX (33:), K017USER) USER/LSES/OSI-ASS NAME
      CHARACTER K017TCVG * (008)
      EQUIVALENCE (KXXX (41:), K017TCVG)

```

```

*           CONVERSATION TAC
CHARACTER  K017RCCC * (003)
EQUIVALENCE (KXXX (49:), K017RCCC)
*           KCRCCC
CHARACTER  K017RCDC * (004)
EQUIVALENCE (KXXX (52:), K017RCDC)
*           KCRCDC
CHARACTER  K017RCF2 * (004)
EQUIVALENCE (KXXX (56:), K017RCF2)
*           SECONDARY FHS/VTSU RET
*           CODE
CHARACTER  K017TAC * (008)
EQUIVALENCE (KXXX (60:), K017TAC )
*           TRANSACTION CODE
CHARACTER  K017TEXT * (085)
EQUIVALENCE (KXXX (68:), K017TEXT)
*****
CHARACTER  K018PTRM * (008)
EQUIVALENCE (KXXX (1:), K018PTRM)
*           PTERM NAME
CHARACTER  K018PRNM * (008)
EQUIVALENCE (KXXX (9:), K018PRNM)
*           PROCESSOR NAME
CHARACTER  K018BCAP * (008)
EQUIVALENCE (KXXX (17:), K018BCAP)
*           BCAM APPLICATION NAME
CHARACTER  K018LTRM * (008)
EQUIVALENCE (KXXX (25:), K018LTRM)
*           LTERM NAME
CHARACTER  K018APPL * (008)
EQUIVALENCE (KXXX (33:), K018APPL)
*           APPLICATION NAME
CHARACTER  K018TEXT * (112)
EQUIVALENCE (KXXX (41:), K018TEXT)
*****
CHARACTER  K019PTRM * (008)
EQUIVALENCE (KXXX (1:), K019PTRM)
*           PTERM NAME
CHARACTER  K019PRNM * (008)
EQUIVALENCE (KXXX (9:), K019PRNM)
*           PROCESSOR NAME
CHARACTER  K019BCAP * (008)
EQUIVALENCE (KXXX (17:), K019BCAP)
*           BCAM APPLICATION NAME
CHARACTER  K019LTRM * (008)
EQUIVALENCE (KXXX (25:), K019LTRM)
*           LTERM NAME
CHARACTER  K019APPL * (008)
EQUIVALENCE (KXXX (33:), K019APPL)
*           APPLICATION NAME
CHARACTER  K019TEXT * (112)
EQUIVALENCE (KXXX (41:), K019TEXT)
*****
CHARACTER  K020PTRM * (008)
EQUIVALENCE (KXXX (1:), K020PTRM)
*           PTERM NAME
CHARACTER  K020PRNM * (008)
EQUIVALENCE (KXXX (9:), K020PRNM)
*           PROCESSOR NAME
CHARACTER  K020BCAP * (008)

```

```

      EQUIVALENCE (KXXX (17:), K020BCAP)
*      CHARACTER K020LTRM * (008)          BCAM APPLICATION NAME
      EQUIVALENCE (KXXX (25:), K020LTRM)
*
      CHARACTER K020USER * (008)
*      EQUIVALENCE (KXXX (33:), K020USER)    USER/LSES/OSI-ASS NAME
      CHARACTER K020TEXT * (112)
*      EQUIVALENCE (KXXX (41:), K020TEXT)
*****
      CHARACTER K021PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K021PTRM)      PTERM NAME
      CHARACTER K021PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K021PRNM)      PROCESSOR NAME
      CHARACTER K021BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K021BCAP)     BCAM APPLICATION NAME
      CHARACTER K021LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K021LTRM)     LTERM NAME
      CHARACTER K021TEXT * (120)
*      EQUIVALENCE (KXXX (33:), K021TEXT)
*****
      CHARACTER K022PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K022PTRM)      PTERM NAME
      CHARACTER K022PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K022PRNM)      PROCESSOR NAME
      CHARACTER K022BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K022BCAP)     BCAM APPLICATION NAME
      CHARACTER K022LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K022LTRM)     LTERM NAME
      CHARACTER K022TEXT * (120)
*      EQUIVALENCE (KXXX (33:), K022TEXT)
*****
      CHARACTER K023MSG * (074)
*      EQUIVALENCE (KXXX (1:), K023MSG)        BROADCAST MESSAGE
      CHARACTER K023TEXT * (078)
*      EQUIVALENCE (KXXX (75:), K023TEXT)
*****
      CHARACTER K024PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K024PTRM)      PTERM NAME
      CHARACTER K024PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K024PRNM)      PROCESSOR NAME
      CHARACTER K024BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K024BCAP)     BCAM APPLICATION NAME
      CHARACTER K024LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K024LTRM)     LTERM NAME
      CHARACTER K024USER * (008)

```

```

EQUIVALENCE (KXXX (33:), K024USER)
*
CHARACTER K024TEXT * (112)
EQUIVALENCE (KXXX (41:), K024TEXT)
*****
CHARACTER K025PTRM * (008)
EQUIVALENCE (KXXX (1:), K025PTRM)
*
CHARACTER K025PRNM * (008)
EQUIVALENCE (KXXX (9:), K025PRNM)
*
CHARACTER K025BCAP * (008)
EQUIVALENCE (KXXX (17:), K025BCAP)
*
CHARACTER K025LTRM * (008)
EQUIVALENCE (KXXX (25:), K025LTRM)
*
CHARACTER K025TEXT * (120)
EQUIVALENCE (KXXX (33:), K025TEXT)
*****
CHARACTER K026PTRM * (008)
EQUIVALENCE (KXXX (1:), K026PTRM)
*
CHARACTER K026PRNM * (008)
EQUIVALENCE (KXXX (9:), K026PRNM)
*
CHARACTER K026BCAP * (008)
EQUIVALENCE (KXXX (17:), K026BCAP)
*
CHARACTER K026LTRM * (008)
EQUIVALENCE (KXXX (25:), K026LTRM)
*
CHARACTER K026USER * (008)
EQUIVALENCE (KXXX (33:), K026USER)
*
CHARACTER K026TEXT * (112)
EQUIVALENCE (KXXX (41:), K026TEXT)
*****
CHARACTER K027PTRM * (008)
EQUIVALENCE (KXXX (1:), K027PTRM)
*
CHARACTER K027PRNM * (008)
EQUIVALENCE (KXXX (9:), K027PRNM)
*
CHARACTER K027BCAP * (008)
EQUIVALENCE (KXXX (17:), K027BCAP)
*
CHARACTER K027LTRM * (008)
EQUIVALENCE (KXXX (25:), K027LTRM)
*
CHARACTER K027TEXT * (120)
EQUIVALENCE (KXXX (33:), K027TEXT)
*****
CHARACTER K029PTRM * (008)
EQUIVALENCE (KXXX (1:), K029PTRM)
*
CHARACTER K029PRNM * (008)
EQUIVALENCE (KXXX (9:), K029PRNM)
*
CHARACTER K029BCAP * (008)

```

```

      EQUIVALENCE (KXXX (17:), K029BCAP)
*      CHARACTER K029LTRM * (008)          BCAM APPLICATION NAME
      EQUIVALENCE (KXXX (25:), K029LTRM)
*
      CHARACTER K029USER * (008)          LTERM NAME
      EQUIVALENCE (KXXX (33:), K029USER)
*
      CHARACTER K029TEXT * (112)         USER/LSES/OSI-ASS NAME
      EQUIVALENCE (KXXX (41:), K029TEXT)
*****
      CHARACTER K030PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K030PTRM)    PTERM NAME
      CHARACTER K030PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K030PRNM)    PROCESSOR NAME
      CHARACTER K030BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K030BCAP)   BCAM APPLICATION NAME
      CHARACTER K030LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K030LTRM)   LTERM NAME
      CHARACTER K030USER * (008)
*      EQUIVALENCE (KXXX (33:), K030USER)   USER/LSES/OSI-ASS NAME
      CHARACTER K030TEXT * (112)
*      EQUIVALENCE (KXXX (41:), K030TEXT)
*****
      CHARACTER K031PTRM * (008)
*      EQUIVALENCE (KXXX (1:), K031PTRM)    PTERM NAME
      CHARACTER K031PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K031PRNM)    PROCESSOR NAME
      CHARACTER K031BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K031BCAP)   BCAM APPLICATION NAME
      CHARACTER K031LTRM * (008)
*      EQUIVALENCE (KXXX (25:), K031LTRM)   LTERM NAME
      CHARACTER K031USER * (008)
*      EQUIVALENCE (KXXX (33:), K031USER)   USER/LSES/OSI-ASS NAME
      CHARACTER K031TEXT * (112)
*      EQUIVALENCE (KXXX (41:), K031TEXT)
*****
      CHARACTER K032CON * (008)
*      EQUIVALENCE (KXXX (1:), K032CON )    CONNECTION NAME
      CHARACTER K032PRNM * (008)
*      EQUIVALENCE (KXXX (9:), K032PRNM)    PROCESSOR NAME
      CHARACTER K032BCAP * (008)
*      EQUIVALENCE (KXXX (17:), K032BCAP)   BCAM APPLICATION NAME
      CHARACTER K032LPAP * (008)
*      EQUIVALENCE (KXXX (25:), K032LPAP)   LPAP NAME
      CHARACTER K032USER * (008)

```

```

EQUIVALENCE (KXXX (33:), K032USER)
*
CHARACTER K032RCF1 * (003)
EQUIVALENCE (KXXX (41:), K032RCF1)
*
CHARACTER K032RCF2 * (004)
EQUIVALENCE (KXXX (44:), K032RCF2)
*
CHARACTER K032TEXT * (105)
EQUIVALENCE (KXXX (48:), K032TEXT)
*****
CHARACTER K033PTRM * (008)
EQUIVALENCE (KXXX (1:), K033PTRM)
*
CHARACTER K033PRNM * (008)
EQUIVALENCE (KXXX (9:), K033PRNM)
*
CHARACTER K033BCAP * (008)
EQUIVALENCE (KXXX (17:), K033BCAP)
*
CHARACTER K033LTRM * (008)
EQUIVALENCE (KXXX (25:), K033LTRM)
*
CHARACTER K033USER * (008)
EQUIVALENCE (KXXX (33:), K033USER)
*
CHARACTER K033REST * (001)
EQUIVALENCE (KXXX (41:), K033REST)
*
CHARACTER K033TEXT * (111)
EQUIVALENCE (KXXX (42:), K033TEXT)
*****
CHARACTER K036PTRM * (008)
EQUIVALENCE (KXXX (1:), K036PTRM)
*
CHARACTER K036PRNM * (008)
EQUIVALENCE (KXXX (9:), K036PRNM)
*
CHARACTER K036BCAP * (008)
EQUIVALENCE (KXXX (17:), K036BCAP)
*
CHARACTER K036LTRM * (008)
EQUIVALENCE (KXXX (25:), K036LTRM)
*
CHARACTER K036RSLT * (001)
EQUIVALENCE (KXXX (33:), K036RSLT)
*
CHARACTER K036REAS * (001)
EQUIVALENCE (KXXX (34:), K036REAS)
*
CHARACTER K036TEXT * (118)
EQUIVALENCE (KXXX (35:), K036TEXT)
*****
CHARACTER K040WLEV * (001)
EQUIVALENCE (KXXX (1:), K040WLEV)
*
CHARACTER K040TEXT * (151)
EQUIVALENCE (KXXX (2:), K040TEXT)
*****

```

```

CHARACTER K041WLEV * (001)
EQUIVALENCE (KXXX (1:), K041WLEV)
*                               WARN LEVEL OF PAGE POOL

CHARACTER K041TEXT * (151)
EQUIVALENCE (KXXX (2:), K041TEXT)
*****

CHARACTER K043DMSE * (004)
EQUIVALENCE (KXXX (1:), K043DMSE)
*                               DMS ERROR CODE

CHARACTER K043FNAM * (054)
EQUIVALENCE (KXXX (5:), K043FNAM)
*                               FILE NAME

CHARACTER K043TEXT * (094)
EQUIVALENCE (KXXX (59:), K043TEXT)
*****

CHARACTER K045PTRM * (008)
EQUIVALENCE (KXXX (1:), K045PTRM)
*                               PTRM NAME

CHARACTER K045PRNM * (008)
EQUIVALENCE (KXXX (9:), K045PRNM)
*                               PROCESSOR NAME

CHARACTER K045BCAP * (008)
EQUIVALENCE (KXXX (17:), K045BCAP)
*                               BCAM APPLICATION NAME

CHARACTER K045LTRM * (008)
EQUIVALENCE (KXXX (25:), K045LTRM)
*                               LTRM NAME

CHARACTER K045PALT * (008)
EQUIVALENCE (KXXX (33:), K045PALT)
*                               LTRM NAME PRINT ADMIN
*                               STATION

CHARACTER K045CID * (008)
EQUIVALENCE (KXXX (41:), K045CID )
*                               PRINTER CONTROL ID

CHARACTER K045TEXT * (104)
EQUIVALENCE (KXXX (49:), K045TEXT)
*****

CHARACTER K046PTRM * (008)
EQUIVALENCE (KXXX (1:), K046PTRM)
*                               PTRM NAME

CHARACTER K046PRNM * (008)
EQUIVALENCE (KXXX (9:), K046PRNM)
*                               PROCESSOR NAME

CHARACTER K046BCAP * (008)
EQUIVALENCE (KXXX (17:), K046BCAP)
*                               BCAM APPLICATION NAME

CHARACTER K046LTRM * (008)
EQUIVALENCE (KXXX (25:), K046LTRM)
*                               LTRM NAME

CHARACTER K046PALT * (008)
EQUIVALENCE (KXXX (33:), K046PALT)
*                               LTRM NAME PRINT ADMIN
*                               STATION

CHARACTER K046CID * (008)
EQUIVALENCE (KXXX (41:), K046CID )
*                               PRINTER CONTROL ID

CHARACTER K046DPID * (008)
EQUIVALENCE (KXXX (49:), K046DPID)
*                               ASYNCHRONOUS MESSAGE ID

CHARACTER K046ERPR * (001)

```

```

*      EQUIVALENCE (KXXX (57:), K046ERPR)
*      CHARACTER K046IMSG * (032)          PRINT ERROR CODE
*      EQUIVALENCE (KXXX (58:), K046IMSG)
*
*      CHARACTER K046TEXT * (063)
*      EQUIVALENCE (KXXX (90:), K046TEXT)
*****
*      CHARACTER K049RCCC * (004)
*      EQUIVALENCE (KXXX (1:), K049RCCC)
*
*      CHARACTER K049TEXT * (148)
*      EQUIVALENCE (KXXX (5:), K049TEXT)
*****
*      CHARACTER K050APPL * (008)
*      EQUIVALENCE (KXXX (1:), K050APPL)
*
*      CHARACTER K050VERS * (008)
*      EQUIVALENCE (KXXX (9:), K050VERS)
*
*      CHARACTER K050TEXT * (136)
*      EQUIVALENCE (KXXX (17:), K050TEXT)
*****
*      CHARACTER K051APPL * (008)
*      EQUIVALENCE (KXXX (1:), K051APPL)
*
*      CHARACTER K051VERS * (008)
*      EQUIVALENCE (KXXX (9:), K051VERS)
*
*      CHARACTER K051TEXT * (136)
*      EQUIVALENCE (KXXX (17:), K051TEXT)
*****
*      CHARACTER K052TASK * (004)
*      EQUIVALENCE (KXXX (1:), K052TASK)
*
*      CHARACTER K052APPL * (008)
*      EQUIVALENCE (KXXX (5:), K052APPL)
*
*      CHARACTER K052PRGV * (004)
*      EQUIVALENCE (KXXX (13:), K052PRGV)
*
*      CHARACTER K052TEXT * (136)
*      EQUIVALENCE (KXXX (17:), K052TEXT)
*****
*      CHARACTER K053CNTR * (006)
*      EQUIVALENCE (KXXX (1:), K053CNTR)
*
*      CHARACTER K053TEXT * (146)
*      EQUIVALENCE (KXXX (7:), K053TEXT)
*****
*      CHARACTER K055ATAC * (008)
*      EQUIVALENCE (KXXX (1:), K055ATAC)
*
*      CHARACTER K055RCCC * (003)
*      EQUIVALENCE (KXXX (9:), K055RCCC)
*
*      CHARACTER K055RCDC * (004)
*      EQUIVALENCE (KXXX (12:), K055RCDC)

```



```

*          KCRCDC
CHARACTER K055USER * (008)
EQUIVALENCE (KXXX (16:), K055USER)
*          USER/LSES/OSI-ASS NAME
CHARACTER K055LTRM * (008)
EQUIVALENCE (KXXX (24:), K055LTRM)
*          LTERM NAME
CHARACTER K055TEXT * (121)
EQUIVALENCE (KXXX (32:), K055TEXT)
*****
CHARACTER K056TASK * (004)
EQUIVALENCE (KXXX (1:), K056TASK)
*          TSN OF UTM TASK
CHARACTER K056TEXT * (148)
EQUIVALENCE (KXXX (5:), K056TEXT)
*****
CHARACTER K058TASK * (004)
EQUIVALENCE (KXXX (1:), K058TASK)
*          TSN OF UTM TASK
CHARACTER K058TEXT * (148)
EQUIVALENCE (KXXX (5:), K058TEXT)
*****
CHARACTER K060TRMA * (006)
EQUIVALENCE (KXXX (1:), K060TRMA)
*          TERM APPLICATION REASON
CHARACTER K060TEXT * (146)
EQUIVALENCE (KXXX (7:), K060TEXT)
*****
CHARACTER K061FNAM * (054)
EQUIVALENCE (KXXX (1:), K061FNAM)
*          FILE NAME
CHARACTER K061TEXT * (098)
EQUIVALENCE (KXXX (55:), K061TEXT)
*****
CHARACTER K063PTRM * (008)
EQUIVALENCE (KXXX (1:), K063PTRM)
*          PTRM NAME
CHARACTER K063PRNM * (008)
EQUIVALENCE (KXXX (9:), K063PRNM)
*          PROCESSOR NAME
CHARACTER K063BCAP * (008)
EQUIVALENCE (KXXX (17:), K063BCAP)
*          BCAM APPLICATION NAME
CHARACTER K063LTRM * (008)
EQUIVALENCE (KXXX (25:), K063LTRM)
*          LTERM NAME
CHARACTER K063FMTN * (008)
EQUIVALENCE (KXXX (33:), K063FMTN)
*          FORMAT NAME
CHARACTER K063RCF1 * (004)
EQUIVALENCE (KXXX (41:), K063RCF1)
*          KCRCDC
CHARACTER K063RCF2 * (004)
EQUIVALENCE (KXXX (45:), K063RCF2)
*          SECONDARY FHS/VTSU RET
*          CODE
CHARACTER K063TEXT * (104)
EQUIVALENCE (KXXX (49:), K063TEXT)
*****
CHARACTER K064PTRM * (008)

```

```

EQUIVALENCE (KXXX (1:), K064PTRM)
*                                     PTERM NAME
CHARACTER K064PRNM * (008)
EQUIVALENCE (KXXX (9:), K064PRNM)
*                                     PROCESSOR NAME
CHARACTER K064BCAP * (008)
EQUIVALENCE (KXXX (17:), K064BCAP)
*                                     BCAM APPLICATION NAME
CHARACTER K064LTRM * (008)
EQUIVALENCE (KXXX (25:), K064LTRM)
*                                     LTERM NAME
CHARACTER K064DEVC * (001)
EQUIVALENCE (KXXX (33:), K064DEVC)
*                                     DEVICE TYPE
CHARACTER K064FIL1 * (001)
EQUIVALENCE (KXXX (34:), K064FIL1)
*                                     APPLICATION STATE
CHARACTER K064FIL2 * (001)
EQUIVALENCE (KXXX (35:), K064FIL2)
*                                     LTERM STATE
CHARACTER K064FIL3 * (002)
EQUIVALENCE (KXXX (36:), K064FIL3)
*                                     PTERM STATE
CHARACTER K064VTRC * (004)
EQUIVALENCE (KXXX (38:), K064VTRC)
*                                     VTSU OR ASECO RETURN CODE
CHARACTER K064IMSG * (032)
EQUIVALENCE (KXXX (42:), K064IMSG)
*                                     FIRST PART OF INPUT
*                                     MESSAGE
CHARACTER K064REAS * (001)
EQUIVALENCE (KXXX (74:), K064REAS)
*                                     REASON
CHARACTER K064CBRC * (004)
EQUIVALENCE (KXXX (75:), K064CBRC)
*                                     VTSUCB RETURN CODE
CHARACTER K064TEXT * (074)
EQUIVALENCE (KXXX (79:), K064TEXT)
*****
CHARACTER K065PTRM * (008)
EQUIVALENCE (KXXX (1:), K065PTRM)
*                                     PTERM NAME
CHARACTER K065PRNM * (008)
EQUIVALENCE (KXXX (9:), K065PRNM)
*                                     PROCESSOR NAME
CHARACTER K065BCAP * (008)
EQUIVALENCE (KXXX (17:), K065BCAP)
*                                     BCAM APPLICATION NAME
CHARACTER K065LTRM * (008)
EQUIVALENCE (KXXX (25:), K065LTRM)
*                                     LTERM NAME
CHARACTER K065FIL1 * (001)
EQUIVALENCE (KXXX (33:), K065FIL1)
*                                     BCAM REQUEST OR ANNO TYPE
*                                     / UTM ANNO TYPE
CHARACTER K065FIL2 * (004)
EQUIVALENCE (KXXX (34:), K065FIL2)
*                                     BCAM INFOWORD
CHARACTER K065TEXT * (115)
EQUIVALENCE (KXXX (38:), K065TEXT)

```

```

*****
CHARACTER K069PTRM * (008)
EQUIVALENCE (KXXX (1:), K069PTRM)
*
CHARACTER K069PRNM * (008)
EQUIVALENCE (KXXX (9:), K069PRNM)
*
CHARACTER K069BCAP * (008)
EQUIVALENCE (KXXX (17:), K069BCAP)
*
CHARACTER K069LTRM * (008)
EQUIVALENCE (KXXX (25:), K069LTRM)
*
CHARACTER K069COTM * (004)
EQUIVALENCE (KXXX (33:), K069COTM)
*
*
CHARACTER K069REAS * (001)
EQUIVALENCE (KXXX (37:), K069REAS)
*
*
CHARACTER K069REA6 * (001)
EQUIVALENCE (KXXX (38:), K069REA6)
*
*
CHARACTER K069TEXT * (114)
EQUIVALENCE (KXXX (39:), K069TEXT)
*****
CHARACTER K070PTRM * (008)
EQUIVALENCE (KXXX (1:), K070PTRM)
*
CHARACTER K070PRNM * (008)
EQUIVALENCE (KXXX (9:), K070PRNM)
*
CHARACTER K070BCAP * (008)
EQUIVALENCE (KXXX (17:), K070BCAP)
*
CHARACTER K070LTRM * (008)
EQUIVALENCE (KXXX (25:), K070LTRM)
*
CHARACTER K070USER * (008)
EQUIVALENCE (KXXX (33:), K070USER)
*
CHARACTER K070COTM * (004)
EQUIVALENCE (KXXX (41:), K070COTM)
*
*
CHARACTER K070CPTM * (004)
EQUIVALENCE (KXXX (45:), K070CPTM)
*
*
CHARACTER K070TEXT * (104)
EQUIVALENCE (KXXX (49:), K070TEXT)
*****
CHARACTER K072STMT * (011)
EQUIVALENCE (KXXX (1:), K072STMT)
*
CHARACTER K072TEXT * (141)
EQUIVALENCE (KXXX (12:), K072TEXT)
*****

```



```

*          CHARACTER K079TEXT * (150)          REASON
EQUIVALENCE (KXXX (3:), K079TEXT)
*****
*          CHARACTER K081IMSG * (005)
EQUIVALENCE (KXXX (1:), K081IMSG)
*          NUMBER OF TERMINAL INPUT
*          MESSAGES
*          CHARACTER K0810MSG * (005)
EQUIVALENCE (KXXX (6:), K0810MSG)
*          NUMBER OF TERMINAL OUTPUT
*          MESSAGES
*          CHARACTER K081CONU * (005)
EQUIVALENCE (KXXX (11:), K081CONU)
*          NUMBER OF CONNECTED USERS
*          CHARACTER K081ATAC * (005)
EQUIVALENCE (KXXX (16:), K081ATAC)
*          NUMBER OF UNPROCESSED
*          ASYNCHRONOUS TACS
*          CHARACTER K081LWRT * (005)
EQUIVALENCE (KXXX (21:), K081LWRT)
*          NUMBER OF USLOG FILE
*          WRITES
*          CHARACTER K081HITR * (003)
EQUIVALENCE (KXXX (26:), K081HITR)
*          CACHE HIT RATE
*          CHARACTER K081WTBF * (003)
EQUIVALENCE (KXXX (29:), K081WTBF)
*          CACHE WAITS FOR BUFFER
*          CHARACTER K081TEXT * (121)
EQUIVALENCE (KXXX (32:), K081TEXT)
*****
*          CHARACTER K086PTRM * (008)
EQUIVALENCE (KXXX (1:), K086PTRM)
*          PTERM NAME
*          CHARACTER K086PRNM * (008)
EQUIVALENCE (KXXX (9:), K086PRNM)
*          PROCESSOR NAME
*          CHARACTER K086BCAP * (008)
EQUIVALENCE (KXXX (17:), K086BCAP)
*          BCAM APPLICATION NAME
*          CHARACTER K086LTRM * (008)
EQUIVALENCE (KXXX (25:), K086LTRM)
*          LTERM NAME
*          CHARACTER K086USER * (008)
EQUIVALENCE (KXXX (33:), K086USER)
*          USER/LSES/OSI-ASS NAME
*          CHARACTER K086SYSD * (002)
EQUIVALENCE (KXXX (41:), K086SYSD)
*          SYSTEM SENSE DATA
*          CHARACTER K086USSD * (002)
EQUIVALENCE (KXXX (43:), K086USSD)
*          USER SENSE DATA
*          CHARACTER K086FMH7 * (080)
EQUIVALENCE (KXXX (45:), K086FMH7)
*          ERROR RECOVERY PROCEDURE
*          MESSAGE
*          CHARACTER K086AGUS * (008)
EQUIVALENCE (KXXX (125:), K086AGUS)
*          JOB-SUBMITTING USER

```

```

CHARACTER K086TEXT * (020)
EQUIVALENCE (KXXX (133:), K086TEXT)
*****
CHARACTER K088LSES * (008)
EQUIVALENCE (KXXX (1:), K088LSES)
*
LSES NAME
CHARACTER K088RSES * (008)
EQUIVALENCE (KXXX (9:), K088RSES)
*
RSES NAME
CHARACTER K088LPAP * (008)
EQUIVALENCE (KXXX (17:), K088LPAP)
*
LPAP NAME
CHARACTER K088SRFG * (004)
EQUIVALENCE (KXXX (25:), K088SRFG)
*
SAVED SESSION STATE
CHARACTER K088PSQN * (004)
EQUIVALENCE (KXXX (29:), K088PSQN)
*
SAVED PET SEQUENCE NUMBER
CHARACTER K088ESQS * (004)
EQUIVALENCE (KXXX (33:), K088ESQS)
*
SAVED SEQUENCE NUMBER
CHARACTER K088EBSS * (004)
EQUIVALENCE (KXXX (37:), K088EBSS)
*
SAVED BRACKET STATE
CHARACTER K088ESQR * (005)
EQUIVALENCE (KXXX (41:), K088ESQR)
*
*
ACTUAL REQUEST SEQUENCE
NUMBER
CHARACTER K088ESRR * (005)
EQUIVALENCE (KXXX (46:), K088ESRR)
*
*
ACTUAL RESPONSE SEQUENCE
NUMBER
CHARACTER K088EBSR * (004)
EQUIVALENCE (KXXX (51:), K088EBSR)
*
ACTUAL BRACKET STATE
CHARACTER K088TEXT * (098)
EQUIVALENCE (KXXX (55:), K088TEXT)
*****
CHARACTER K089GNDA * (003)
EQUIVALENCE (KXXX (1:), K089GNDA)
*
*
GENERATION DATE
ASYNCHRONOUS MESSAGE
CHARACTER K089GNTI * (008)
EQUIVALENCE (KXXX (4:), K089GNTI)
*
*
GENERATION TIME
ASYNCHRONOUS MESSAGE
CHARACTER K089DEST * (008)
EQUIVALENCE (KXXX (12:), K089DEST)
*
*
DESTINATION OF
ASYNCHRONOUS MSG
CHARACTER K089GNUS * (008)
EQUIVALENCE (KXXX (20:), K089GNUS)
*
*
USER NAME OF ASYNCHRON.
MESSAGE GENERATION
CHARACTER K089USER * (008)
EQUIVALENCE (KXXX (28:), K089USER)
*
USER/LSES/OSI-ASS NAME
CHARACTER K089DLDA * (003)
EQUIVALENCE (KXXX (36:), K089DLDA)
*
DAY OF KDCS CALL PADM

```

```

*           CHARACTER K089DLTI * (008)           DL/DA
*           EQUIVALENCE (KXXX (39:), K089DLTI)
*
*           TIME OF KDCS CALL PADM
*           DL/DA
*           CHARACTER K089CHAI * (003)
*           EQUIVALENCE (KXXX (47:), K089CHAI)
*
*           CHAINED MESSAGE
*           INFORMATION
*           CHARACTER K089TEXT * (103)
*           EQUIVALENCE (KXXX (50:), K089TEXT)
*****
*           CHARACTER K090DEST * (008)
*           EQUIVALENCE (KXXX (1:), K090DEST)
*
*           DESTINATION OF
*           ASYNCHRONOUS MSG
*           CHARACTER K090USER * (008)
*           EQUIVALENCE (KXXX (9:), K090USER)
*
*           USER/LSES/OSI-ASS NAME
*           CHARACTER K090DLDA * (003)
*           EQUIVALENCE (KXXX (17:), K090DLDA)
*
*           DAY OF KDCS CALL PADM
*           DL/DA
*           CHARACTER K090DLTI * (008)
*           EQUIVALENCE (KXXX (20:), K090DLTI)
*
*           TIME OF KDCS CALL PADM
*           DL/DA
*           CHARACTER K090TEXT * (125)
*           EQUIVALENCE (KXXX (28:), K090TEXT)
*****
*           CHARACTER K091PTRM * (008)
*           EQUIVALENCE (KXXX (1:), K091PTRM)
*
*           PTERM NAME
*           CHARACTER K091PRNM * (008)
*           EQUIVALENCE (KXXX (9:), K091PRNM)
*
*           PROCESSOR NAME
*           CHARACTER K091BCAP * (008)
*           EQUIVALENCE (KXXX (17:), K091BCAP)
*
*           BCAM APPLICATION NAME
*           CHARACTER K091LTRM * (008)
*           EQUIVALENCE (KXXX (25:), K091LTRM)
*
*           LTERM NAME
*           CHARACTER K091USER * (008)
*           EQUIVALENCE (KXXX (33:), K091USER)
*
*           USER/LSES/OSI-ASS NAME
*           CHARACTER K091ASRC * (004)
*           EQUIVALENCE (KXXX (41:), K091ASRC)
*
*           ASECO RETURN CODE (CHIP
*           CARD MODULE)
*           CHARACTER K091TEXT * (108)
*           EQUIVALENCE (KXXX (45:), K091TEXT)
*****
*           CHARACTER K092PTRM * (008)
*           EQUIVALENCE (KXXX (1:), K092PTRM)
*
*           PTERM NAME
*           CHARACTER K092PRNM * (008)
*           EQUIVALENCE (KXXX (9:), K092PRNM)
*
*           PROCESSOR NAME
*           CHARACTER K092BCAP * (008)
*           EQUIVALENCE (KXXX (17:), K092BCAP)

```

```

*          CHARACTER K092LTRM * (008)          BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (25:), K092LTRM)
*
*          CHARACTER K092USER * (008)          LTERM NAME
          EQUIVALENCE (KXXX (33:), K092USER)
*
*          CHARACTER K092PAS1 * (020)          USER/LSES/OSI-ASS NAME
          EQUIVALENCE (KXXX (41:), K092PAS1)
*
*          CHARACTER K092PAS2 * (020)          SPACE FOR PASSWORD
          EQUIVALENCE (KXXX (61:), K092PAS2)
*
*          CHARACTER K092PAS3 * (020)          SPACE FOR PASSWORD
          EQUIVALENCE (KXXX (81:), K092PAS3)
*
*          CHARACTER K092TEXT * (052)          SPACE FOR PASSWORD
          EQUIVALENCE (KXXX (101:), K092TEXT)
*****
*          CHARACTER K093PTRM * (008)          PTERM NAME
          EQUIVALENCE (KXXX (1:), K093PTRM)
*
*          CHARACTER K093PRNM * (008)          PROCESSOR NAME
          EQUIVALENCE (KXXX (9:), K093PRNM)
*
*          CHARACTER K093BCAP * (008)          BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (17:), K093BCAP)
*
*          CHARACTER K093LTRM * (008)          LTERM NAME
          EQUIVALENCE (KXXX (25:), K093LTRM)
*
*          CHARACTER K093USER * (008)          USER/LSES/OSI-ASS NAME
          EQUIVALENCE (KXXX (33:), K093USER)
*
*          CHARACTER K093HSTA * (002)          HEIGHT OF STACK
          EQUIVALENCE (KXXX (41:), K093HSTA)
*
*          CHARACTER K093MSTA * (002)          MAXIMUM STACK HEIGHT
          EQUIVALENCE (KXXX (43:), K093MSTA)
*
*          CHARACTER K093TEXT * (108)
          EQUIVALENCE (KXXX (45:), K093TEXT)
*****
*          CHARACTER K094PTRM * (008)          PTERM NAME
          EQUIVALENCE (KXXX (1:), K094PTRM)
*
*          CHARACTER K094PRNM * (008)          PROCESSOR NAME
          EQUIVALENCE (KXXX (9:), K094PRNM)
*
*          CHARACTER K094BCAP * (008)          BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (17:), K094BCAP)
*
*          CHARACTER K094LTRM * (008)          LTERM NAME
          EQUIVALENCE (KXXX (25:), K094LTRM)
*
*          CHARACTER K094USER * (008)          USER/LSES/OSI-ASS NAME
          EQUIVALENCE (KXXX (33:), K094USER)
*
*          CHARACTER K094RCF1 * (003)
          EQUIVALENCE (KXXX (41:), K094RCF1)

```



```

*                                     RETURN CODE 1
  CHARACTER K094TEXT * (109)
  EQUIVALENCE (KXXX (44:), K094TEXT)
*****
  CHARACTER K097PTRM * (008)
  EQUIVALENCE (KXXX (1:), K097PTRM)
*                                     PTERM NAME
  CHARACTER K097PRNM * (008)
  EQUIVALENCE (KXXX (9:), K097PRNM)
*                                     PROCESSOR NAME
  CHARACTER K097BCAP * (008)
  EQUIVALENCE (KXXX (17:), K097BCAP)
*                                     BCAM APPLICATION NAME
  CHARACTER K097LTRM * (008)
  EQUIVALENCE (KXXX (25:), K097LTRM)
*                                     LTERM NAME
  CHARACTER K097USER * (008)
  EQUIVALENCE (KXXX (33:), K097USER)
*                                     USER/LSES/OSI-ASS NAME
  CHARACTER K097TEXT * (112)
  EQUIVALENCE (KXXX (41:), K097TEXT)
*****
  CHARACTER K098PTRM * (008)
  EQUIVALENCE (KXXX (1:), K098PTRM)
*                                     PTERM NAME
  CHARACTER K098PRNM * (008)
  EQUIVALENCE (KXXX (9:), K098PRNM)
*                                     PROCESSOR NAME
  CHARACTER K098BCAP * (008)
  EQUIVALENCE (KXXX (17:), K098BCAP)
*                                     BCAM APPLICATION NAME
  CHARACTER K098LTRM * (008)
  EQUIVALENCE (KXXX (25:), K098LTRM)
*                                     LTERM NAME
  CHARACTER K098USER * (008)
  EQUIVALENCE (KXXX (33:), K098USER)
*                                     USER/LSES/OSI-ASS NAME
  CHARACTER K098RCF1 * (004)
  EQUIVALENCE (KXXX (41:), K098RCF1)
*                                     RETURN CODE 1
  CHARACTER K098RCF2 * (004)
  EQUIVALENCE (KXXX (45:), K098RCF2)
*                                     RETURN CODE 2
  CHARACTER K098TEXT * (104)
  EQUIVALENCE (KXXX (49:), K098TEXT)
*****
  CHARACTER K101PTRM * (008)
  EQUIVALENCE (KXXX (1:), K101PTRM)
*                                     PTERM NAME
  CHARACTER K101PRNM * (008)
  EQUIVALENCE (KXXX (9:), K101PRNM)
*                                     PROCESSOR NAME
  CHARACTER K101BCAP * (008)
  EQUIVALENCE (KXXX (17:), K101BCAP)
*                                     BCAM APPLICATION NAME
  CHARACTER K101LTRM * (008)
  EQUIVALENCE (KXXX (25:), K101LTRM)
*                                     LTERM NAME
  CHARACTER K101USER * (008)
  EQUIVALENCE (KXXX (33:), K101USER)

```

```

*                                     USER/LSES/OSI-ASS NAME
  CHARACTER K101TEXT * (112)
  EQUIVALENCE (KXXX (41:), K101TEXT)
*****
  CHARACTER K104UTMD * (007)
  EQUIVALENCE (KXXX (1:), K104UTMD)
*                                     UTM-D EVENT
  CHARACTER K104LSES * (008)
  EQUIVALENCE (KXXX (8:), K104LSES)
*                                     LSES NAME
  CHARACTER K104LPAP * (008)
  EQUIVALENCE (KXXX (16:), K104LPAP)
*                                     LPAP NAME
  CHARACTER K104AGUS * (008)
  EQUIVALENCE (KXXX (24:), K104AGUS)
*                                     JOB-SUBMITTING USER
  CHARACTER K104OCVS * (001)
  EQUIVALENCE (KXXX (32:), K104OCVS)
*                                     OLD CONVERSATION STATE
  CHARACTER K104OTAS * (001)
  EQUIVALENCE (KXXX (33:), K104OTAS)
*                                     OLD TRANSACTION STATE
  CHARACTER K104ACTI * (006)
  EQUIVALENCE (KXXX (34:), K104ACTI)
*                                     SYSTEM ACTION
  CHARACTER K104NCVS * (001)
  EQUIVALENCE (KXXX (40:), K104NCVS)
*                                     NEW CONVERSATION STATE
  CHARACTER K104NTAS * (001)
  EQUIVALENCE (KXXX (41:), K104NTAS)
*                                     NEW TRANSACTION STATE
  CHARACTER K104TEXT * (111)
  EQUIVALENCE (KXXX (42:), K104TEXT)
*****
  CHARACTER K105LSES * (008)
  EQUIVALENCE (KXXX (1:), K105LSES)
*                                     LSES NAME
  CHARACTER K105LPAP * (008)
  EQUIVALENCE (KXXX (9:), K105LPAP)
*                                     LPAP NAME
  CHARACTER K105AGUS * (008)
  EQUIVALENCE (KXXX (17:), K105AGUS)
*                                     JOB-SUBMITTING USER
  CHARACTER K105SYST * (004)
  EQUIVALENCE (KXXX (25:), K105SYST)
*                                     SYSTEM
  CHARACTER K105TEXT * (124)
  EQUIVALENCE (KXXX (29:), K105TEXT)
*****
  CHARACTER K106PTRM * (008)
  EQUIVALENCE (KXXX (1:), K106PTRM)
*                                     PTRM NAME
  CHARACTER K106PRNM * (008)
  EQUIVALENCE (KXXX (9:), K106PRNM)
*                                     PROCESSOR NAME
  CHARACTER K106BCAP * (008)
  EQUIVALENCE (KXXX (17:), K106BCAP)
*                                     BCAM APPLICATION NAME
  CHARACTER K106LTRM * (008)
  EQUIVALENCE (KXXX (25:), K106LTRM)

```

```

*          CHARACTER K106USER * (008)          LTERM NAME
EQUIVALENCE (KXXX (33:), K106USER)
*
*          CHARACTER K106DEVC * (001)          USER/LSES/OSI-ASS NAME
EQUIVALENCE (KXXX (41:), K106DEVC)
*
*          CHARACTER K106FIL1 * (001)          DEVICE TYPE
EQUIVALENCE (KXXX (42:), K106FIL1)
*
*          CHARACTER K106FIL2 * (001)          APPLICATION STATE
EQUIVALENCE (KXXX (43:), K106FIL2)
*
*          CHARACTER K106FIL3 * (002)          LTERM STATE
EQUIVALENCE (KXXX (44:), K106FIL3)
*
*          CHARACTER K106VTRC * (004)          PTERM STATE
EQUIVALENCE (KXXX (46:), K106VTRC)
*
*          CHARACTER K106CBRC * (004)          VTSU OR ASECO RETURN CODE
EQUIVALENCE (KXXX (50:), K106CBRC)
*
*          CHARACTER K106MSG * (032)          VTSUCB RETURN CODE
EQUIVALENCE (KXXX (54:), K106MSG)
*
*          CHARACTER K106FMTN * (008)          FIRST PART OF OUTPUT
EQUIVALENCE (KXXX (86:), K106FMTN)          MESSAGE
*
*          CHARACTER K106CCSN * (008)          FORMAT NAME
EQUIVALENCE (KXXX (94:), K106CCSN)
*
*          CHARACTER K106TEXT * (051)          CCSNAME
EQUIVALENCE (KXXX (102:), K106TEXT)
*****
*          CHARACTER K107TTYP * (008)          TERMINAL TYPE
EQUIVALENCE (KXXX (1:), K107TTYP)
*
*          CHARACTER K107TEXT * (144)
EQUIVALENCE (KXXX (9:), K107TEXT)
*****
*          CHARACTER K108PTRM * (008)          PTERM NAME
EQUIVALENCE (KXXX (1:), K108PTRM)
*
*          CHARACTER K108PRNM * (008)          PROCESSOR NAME
EQUIVALENCE (KXXX (9:), K108PRNM)
*
*          CHARACTER K108BCAP * (008)          BCAM APPLICATION NAME
EQUIVALENCE (KXXX (17:), K108BCAP)
*
*          CHARACTER K108LTRM * (008)          LTERM NAME
EQUIVALENCE (KXXX (25:), K108LTRM)
*
*          CHARACTER K108USER * (008)          USER/LSES/OSI-ASS NAME
EQUIVALENCE (KXXX (33:), K108USER)
*
*          CHARACTER K108ASRC * (004)          ASECO RETURN CODE (CHIP
EQUIVALENCE (KXXX (41:), K108ASRC)          CARD MODULE)
*
*

```

```

CHARACTER K108TEXT * (108)
EQUIVALENCE (KXXX (45:), K108TEXT)
*****
CHARACTER K109PTRM * (008)
EQUIVALENCE (KXXX (1:), K109PTRM)
*
CHARACTER K109PRNM * (008)
EQUIVALENCE (KXXX (9:), K109PRNM)
*
CHARACTER K109BCAP * (008)
EQUIVALENCE (KXXX (17:), K109BCAP)
*
CHARACTER K109LTRM * (008)
EQUIVALENCE (KXXX (25:), K109LTRM)
*
CHARACTER K109USER * (008)
EQUIVALENCE (KXXX (33:), K109USER)
*
CHARACTER K109ASRC * (004)
EQUIVALENCE (KXXX (41:), K109ASRC)
*
CHARACTER K109ADFN * (016)
EQUIVALENCE (KXXX (45:), K109ADFN)
*
CHARACTER K109TEXT * (092)
EQUIVALENCE (KXXX (61:), K109TEXT)
*****
CHARACTER K115PTRM * (008)
EQUIVALENCE (KXXX (1:), K115PTRM)
*
CHARACTER K115PRNM * (008)
EQUIVALENCE (KXXX (9:), K115PRNM)
*
CHARACTER K115BCAP * (008)
EQUIVALENCE (KXXX (17:), K115BCAP)
*
CHARACTER K115LTRM * (008)
EQUIVALENCE (KXXX (25:), K115LTRM)
*
CHARACTER K115SNPT * (008)
EQUIVALENCE (KXXX (33:), K115SNPT)
*
CHARACTER K115SNPR * (008)
EQUIVALENCE (KXXX (41:), K115SNPR)
*
CHARACTER K115SNLT * (008)
EQUIVALENCE (KXXX (49:), K115SNLT)
*
CHARACTER K115CCC * (001)
EQUIVALENCE (KXXX (57:), K115CCC )
*
CHARACTER K115REAS * (001)
EQUIVALENCE (KXXX (58:), K115REAS)
*
CHARACTER K115ANNO * (032)
EQUIVALENCE (KXXX (59:), K115ANNO)
*

```

PTERM NAME  
PROCESSOR NAME  
BCAM APPLICATION NAME  
LTERM NAME  
USER/LSES/OSI-ASS NAME  
ASECO RETURN CODE (CHIP  
CARD MODULE)  
ADF NAME  
PTERM NAME  
PROCESSOR NAME  
BCAM APPLICATION NAME  
LTERM NAME  
MUX SESSION PTERM NAME  
MUX SESSION PROCESSOR  
NAME  
MUX SESSION LTERM NAME  
CONTXT MACRO: CONDITION  
CODE IN PCR FORMAT  
REASON  
ANNO RECEIVED

```

CHARACTER K115TEXT * (062)
EQUIVALENCE (KXXX (91:), K115TEXT)
*****
CHARACTER K116PTRM * (008)
EQUIVALENCE (KXXX (1:), K116PTRM)
*
CHARACTER K116PRNM * (008)
EQUIVALENCE (KXXX (9:), K116PRNM)
*
CHARACTER K116BCAP * (008)
EQUIVALENCE (KXXX (17:), K116BCAP)
*
CHARACTER K116LTRM * (008)
EQUIVALENCE (KXXX (25:), K116LTRM)
*
CHARACTER K116SNPT * (008)
EQUIVALENCE (KXXX (33:), K116SNPT)
*
CHARACTER K116SNPR * (008)
EQUIVALENCE (KXXX (41:), K116SNPR)
*
CHARACTER K116SNLT * (008)
EQUIVALENCE (KXXX (49:), K116SNLT)
*
CHARACTER K116USER * (008)
EQUIVALENCE (KXXX (57:), K116USER)
*
CHARACTER K116REAS * (001)
EQUIVALENCE (KXXX (65:), K116REAS)
*
CHARACTER K116TEXT * (087)
EQUIVALENCE (KXXX (66:), K116TEXT)
*****
CHARACTER K117PTRM * (008)
EQUIVALENCE (KXXX (1:), K117PTRM)
*
CHARACTER K117PRNM * (008)
EQUIVALENCE (KXXX (9:), K117PRNM)
*
CHARACTER K117BCAP * (008)
EQUIVALENCE (KXXX (17:), K117BCAP)
*
CHARACTER K117LTRM * (008)
EQUIVALENCE (KXXX (25:), K117LTRM)
*
CHARACTER K117SNPT * (008)
EQUIVALENCE (KXXX (33:), K117SNPT)
*
CHARACTER K117SNPR * (008)
EQUIVALENCE (KXXX (41:), K117SNPR)
*
CHARACTER K117SNLT * (008)
EQUIVALENCE (KXXX (49:), K117SNLT)
*
CHARACTER K117USER * (008)
EQUIVALENCE (KXXX (57:), K117USER)
*
CHARACTER K117REAS * (001)

```

```

      EQUIVALENCE (KXXX (65:), K117REAS)
*
      CHARACTER K117TEXT * (087)
      EQUIVALENCE (KXXX (66:), K117TEXT)
*****
      CHARACTER K119OSLP * (008)
      EQUIVALENCE (KXXX (1:), K119OSLP)
*
      CHARACTER K119USER * (008)
      EQUIVALENCE (KXXX (9:), K119USER)
*
      CHARACTER K119TAC * (008)
      EQUIVALENCE (KXXX (17:), K119TAC )
*
      CHARACTER K119DIA1 * (004)
      EQUIVALENCE (KXXX (25:), K119DIA1)
*
      CHARACTER K119DIA2 * (004)
      EQUIVALENCE (KXXX (29:), K119DIA2)
*
      CHARACTER K119DIA3 * (004)
      EQUIVALENCE (KXXX (33:), K119DIA3)
*
      CHARACTER K119TEXT * (116)
      EQUIVALENCE (KXXX (37:), K119TEXT)
*****
      CHARACTER K120PTRM * (008)
      EQUIVALENCE (KXXX (1:), K120PTRM)
*
      CHARACTER K120PRNM * (008)
      EQUIVALENCE (KXXX (9:), K120PRNM)
*
      CHARACTER K120BCAP * (008)
      EQUIVALENCE (KXXX (17:), K120BCAP)
*
      CHARACTER K120LTRM * (008)
      EQUIVALENCE (KXXX (25:), K120LTRM)
*
      CHARACTER K120USER * (008)
      EQUIVALENCE (KXXX (33:), K120USER)
*
      CHARACTER K120TEXT * (112)
      EQUIVALENCE (KXXX (41:), K120TEXT)
*****
      CHARACTER K121PTRM * (008)
      EQUIVALENCE (KXXX (1:), K121PTRM)
*
      CHARACTER K121PRNM * (008)
      EQUIVALENCE (KXXX (9:), K121PRNM)
*
      CHARACTER K121BCAP * (008)
      EQUIVALENCE (KXXX (17:), K121BCAP)
*
      CHARACTER K121LTRM * (008)
      EQUIVALENCE (KXXX (25:), K121LTRM)
*
      CHARACTER K121USER * (008)
      EQUIVALENCE (KXXX (33:), K121USER)
*
      CHARACTER K121PAS1 * (020)

```

```

      EQUIVALENCE (KXXX (41:), K121PAS1)
*      CHARACTER K121PAS2 * (020)          SPACE FOR PASSWORD
      EQUIVALENCE (KXXX (61:), K121PAS2)
*      CHARACTER K121PAS3 * (020)          SPACE FOR PASSWORD
      EQUIVALENCE (KXXX (81:), K121PAS3)
*      CHARACTER K121NUMD * (002)          SPACE FOR PASSWORD
      EQUIVALENCE (KXXX (101:), K121NUMD)
*      NUMBER DAYS PASSWORD
*      VALID
      CHARACTER K121TEXT * (050)
      EQUIVALENCE (KXXX (103:), K121TEXT)
*****
      CHARACTER K123LTRM * (008)
      EQUIVALENCE (KXXX (1:), K123LTRM)
*      LTERM NAME
      CHARACTER K123TAC * (008)
      EQUIVALENCE (KXXX (9:), K123TAC )
*      TRANSACTION CODE
      CHARACTER K123USER * (008)
      EQUIVALENCE (KXXX (17:), K123USER)
*      USER/LSES/OSI-ASS NAME
      CHARACTER K123TEXT * (128)
      EQUIVALENCE (KXXX (25:), K123TEXT)
*****
      CHARACTER K124RCXA * (004)
      EQUIVALENCE (KXXX (1:), K124RCXA)
*      RETURNCODE XAP-TP
*      STARTFUNCTIONS
      CHARACTER K124PHAX * (014)
      EQUIVALENCE (KXXX (5:), K124PHAX)
*      INIT or START/RESTART of
*      XAP-TP
      CHARACTER K124TEXT * (134)
      EQUIVALENCE (KXXX (19:), K124TEXT)
*****
      CHARACTER K125PTRM * (008)
      EQUIVALENCE (KXXX (1:), K125PTRM)
*      PTRM NAME
      CHARACTER K125PRNM * (008)
      EQUIVALENCE (KXXX (9:), K125PRNM)
*      PROCESSOR NAME
      CHARACTER K125BCAP * (008)
      EQUIVALENCE (KXXX (17:), K125BCAP)
*      BCAM APPLICATION NAME
      CHARACTER K125LTRM * (008)
      EQUIVALENCE (KXXX (25:), K125LTRM)
*      LTERM NAME
      CHARACTER K125USER * (008)
      EQUIVALENCE (KXXX (33:), K125USER)
*      USER/LSES/OSI-ASS NAME
      CHARACTER K125TEXT * (112)
      EQUIVALENCE (KXXX (41:), K125TEXT)
*****
      CHARACTER K126SATR * (004)
      EQUIVALENCE (KXXX (1:), K126SATR)
*      SAT RETURNCODE
      CHARACTER K126TEXT * (148)

```

```

EQUIVALENCE (KXXX (5:), K126TEXT)
*****
CHARACTER K128CON * (008)
EQUIVALENCE (KXXX (1:), K128CON )
*
CHARACTER K128PRNM * (008)
EQUIVALENCE (KXXX (9:), K128PRNM)
*
CHARACTER K128BCAP * (008)
EQUIVALENCE (KXXX (17:), K128BCAP)
*
CHARACTER K128LPAP * (008)
EQUIVALENCE (KXXX (25:), K128LPAP)
*
CHARACTER K128LSES * (008)
EQUIVALENCE (KXXX (33:), K128LSES)
*
CHARACTER K128REAS * (001)
EQUIVALENCE (KXXX (41:), K128REAS)
*
CHARACTER K128RCDC * (004)
EQUIVALENCE (KXXX (42:), K128RCDC)
*
CHARACTER K128TAC * (008)
EQUIVALENCE (KXXX (46:), K128TAC )
*
CHARACTER K128TEXT * (099)
EQUIVALENCE (KXXX (54:), K128TEXT)
*****
CHARACTER K130TPRI * (001)
EQUIVALENCE (KXXX (1:), K130TPRI)
*
CHARACTER K130TASK * (004)
EQUIVALENCE (KXXX (2:), K130TASK)
*
CHARACTER K130TEXT * (147)
EQUIVALENCE (KXXX (6:), K130TEXT)
*****
CHARACTER K135PTRM * (008)
EQUIVALENCE (KXXX (1:), K135PTRM)
*
CHARACTER K135PRNM * (008)
EQUIVALENCE (KXXX (9:), K135PRNM)
*
CHARACTER K135BCAP * (008)
EQUIVALENCE (KXXX (17:), K135BCAP)
*
CHARACTER K135LTRM * (008)
EQUIVALENCE (KXXX (25:), K135LTRM)
*
CHARACTER K135UPCR * (001)
EQUIVALENCE (KXXX (33:), K135UPCR)
*
CHARACTER K135UPCS * (002)
EQUIVALENCE (KXXX (34:), K135UPCS)
*
CHARACTER K135UPCP * (004)
EQUIVALENCE (KXXX (36:), K135UPCP)
*
CHARACTER K135TEXT * (113)

```



```

*****
EQUIVALENCE (KXXX (40:), K135TEXT)
*****
CHARACTER K137FNAM * (054)
EQUIVALENCE (KXXX (1:), K137FNAM)
*
CHARACTER K137TEXT * (098)
EQUIVALENCE (KXXX (55:), K137TEXT)
*****
CHARACTER K138FNAM * (054)
EQUIVALENCE (KXXX (1:), K138FNAM)
*
CHARACTER K138TEXT * (098)
EQUIVALENCE (KXXX (55:), K138TEXT)
*****
CHARACTER K139FNAM * (054)
EQUIVALENCE (KXXX (1:), K139FNAM)
*
CHARACTER K139TEXT * (098)
EQUIVALENCE (KXXX (55:), K139TEXT)
*****
CHARACTER K140PTRM * (008)
EQUIVALENCE (KXXX (1:), K140PTRM)
*
CHARACTER K140PRNM * (008)
EQUIVALENCE (KXXX (9:), K140PRNM)
*
CHARACTER K140BCAP * (008)
EQUIVALENCE (KXXX (17:), K140BCAP)
*
CHARACTER K140LTRM * (008)
EQUIVALENCE (KXXX (25:), K140LTRM)
*
CHARACTER K140MXP1 * (004)
EQUIVALENCE (KXXX (33:), K140MXP1)
*
*
CHARACTER K140MXP2 * (004)
EQUIVALENCE (KXXX (37:), K140MXP2)
*
*
CHARACTER K140TEXT * (112)
EQUIVALENCE (KXXX (41:), K140TEXT)
*****
CHARACTER K141PTRM * (008)
EQUIVALENCE (KXXX (1:), K141PTRM)
*
CHARACTER K141PRNM * (008)
EQUIVALENCE (KXXX (9:), K141PRNM)
*
CHARACTER K141BCAP * (008)
EQUIVALENCE (KXXX (17:), K141BCAP)
*
CHARACTER K141LTRM * (008)
EQUIVALENCE (KXXX (25:), K141LTRM)
*
CHARACTER K141MXP1 * (004)
EQUIVALENCE (KXXX (33:), K141MXP1)
*
*
CHARACTER K141TEXT * (116)

```

```

EQUIVALENCE (KXXX (37:), K141TEXT)
*****
CHARACTER K142PTRM * (008)
EQUIVALENCE (KXXX (1:), K142PTRM)
*
CHARACTER K142PRNM * (008)
EQUIVALENCE (KXXX (9:), K142PRNM)
*
CHARACTER K142BCAP * (008)
EQUIVALENCE (KXXX (17:), K142BCAP)
*
CHARACTER K142LTRM * (008)
EQUIVALENCE (KXXX (25:), K142LTRM)
*
CHARACTER K142MXPT * (008)
EQUIVALENCE (KXXX (33:), K142MXPT)
*
CHARACTER K142MXPR * (008)
EQUIVALENCE (KXXX (41:), K142MXPR)
*
CHARACTER K142MXLT * (008)
EQUIVALENCE (KXXX (49:), K142MXLT)
*
CHARACTER K142TEXT * (096)
EQUIVALENCE (KXXX (57:), K142TEXT)
*****
CHARACTER K143PTRM * (008)
EQUIVALENCE (KXXX (1:), K143PTRM)
*
CHARACTER K143PRNM * (008)
EQUIVALENCE (KXXX (9:), K143PRNM)
*
CHARACTER K143BCAP * (008)
EQUIVALENCE (KXXX (17:), K143BCAP)
*
CHARACTER K143LTRM * (008)
EQUIVALENCE (KXXX (25:), K143LTRM)
*
CHARACTER K143STS1 * (002)
EQUIVALENCE (KXXX (33:), K143STS1)
*
*
*
CHARACTER K143STS2 * (002)
EQUIVALENCE (KXXX (35:), K143STS2)
*
*
*
CHARACTER K143STS3 * (002)
EQUIVALENCE (KXXX (37:), K143STS3)
*
*
*
CHARACTER K143STS4 * (002)
EQUIVALENCE (KXXX (39:), K143STS4)
*
*
*
CHARACTER K143TEXT * (112)
EQUIVALENCE (KXXX (41:), K143TEXT)
*****
CHARACTER K144PTRM * (008)
EQUIVALENCE (KXXX (1:), K144PTRM)
*

```

```

CHARACTER K144PRNM * (008)
EQUIVALENCE (KXXX (9:), K144PRNM)
*
PROCESSOR NAME
CHARACTER K144BCAP * (008)
EQUIVALENCE (KXXX (17:), K144BCAP)
*
BCAM APPLICATION NAME
CHARACTER K144LTRM * (008)
EQUIVALENCE (KXXX (25:), K144LTRM)
*
LTERM NAME
CHARACTER K144DEVC * (001)
EQUIVALENCE (KXXX (33:), K144DEVC)
*
DEVICE TYPE
CHARACTER K144FIL1 * (001)
EQUIVALENCE (KXXX (34:), K144FIL1)
*
APPLICATION STATE
CHARACTER K144FIL2 * (001)
EQUIVALENCE (KXXX (35:), K144FIL2)
*
LTERM STATE
CHARACTER K144FIL3 * (002)
EQUIVALENCE (KXXX (36:), K144FIL3)
*
PTERM STATE
CHARACTER K144VTRC * (004)
EQUIVALENCE (KXXX (38:), K144VTRC)
*
VTSU OR ASECO RETURN CODE
CHARACTER K144CBRC * (004)
EQUIVALENCE (KXXX (42:), K144CBRC)
*
VTSUCB RETURN CODE
CHARACTER K1440MSG * (032)
EQUIVALENCE (KXXX (46:), K1440MSG)
*
FIRST PART OF OUTPUT
*
MESSAGE
CHARACTER K144FMTN * (008)
EQUIVALENCE (KXXX (78:), K144FMTN)
*
FORMAT NAME
CHARACTER K144CCSN * (008)
EQUIVALENCE (KXXX (86:), K144CCSN)
*
CCSNNAME
CHARACTER K144TEXT * (059)
EQUIVALENCE (KXXX (94:), K144TEXT)
*****
CHARACTER K145PTRM * (008)
EQUIVALENCE (KXXX (1:), K145PTRM)
*
PTERM NAME
CHARACTER K145PRNM * (008)
EQUIVALENCE (KXXX (9:), K145PRNM)
*
PROCESSOR NAME
CHARACTER K145BCAP * (008)
EQUIVALENCE (KXXX (17:), K145BCAP)
*
BCAM APPLICATION NAME
CHARACTER K145LTRM * (008)
EQUIVALENCE (KXXX (25:), K145LTRM)
*
LTERM NAME
CHARACTER K145USER * (008)
EQUIVALENCE (KXXX (33:), K145USER)
*
USER/LSES/OSI-ASS NAME
CHARACTER K145TEXT * (112)
EQUIVALENCE (KXXX (41:), K145TEXT)
*****
CHARACTER K146BCMO * (004)
EQUIVALENCE (KXXX (1:), K146BCMO)

```

```

*          CHARACTER K146BCMR * (004)          BCMM-OPCODE
          EQUIVALENCE (KXXX (5:), K146BCMR)
*
*          CHARACTER K146STDH * (008)          BCMM-RETURNCODE
          EQUIVALENCE (KXXX (9:), K146STDH)
*
*          CHARACTER K146TASK * (004)          BS2000 STANDARDHEADER
          EQUIVALENCE (KXXX (17:), K146TASK)
*
*          CHARACTER K146BCAP * (008)          TSN OF UTM TASK
          EQUIVALENCE (KXXX (21:), K146BCAP)
*
*          CHARACTER K146TEXT * (124)         BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (29:), K146TEXT)
*****
*          CHARACTER K147PTRM * (008)
          EQUIVALENCE (KXXX (1:), K147PTRM)
*
*          CHARACTER K147PRNM * (008)         PTERM NAME
          EQUIVALENCE (KXXX (9:), K147PRNM)
*
*          CHARACTER K147BCAP * (008)         PROCESSOR NAME
          EQUIVALENCE (KXXX (17:), K147BCAP)
*
*          CHARACTER K147LTRM * (008)         BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (25:), K147LTRM)
*
*          CHARACTER K147USER * (008)         LTERM NAME
          EQUIVALENCE (KXXX (33:), K147USER)
*
*          CHARACTER K147TEXT * (112)        USER/LSES/OSI-ASS NAME
          EQUIVALENCE (KXXX (41:), K147TEXT)
*****
*          CHARACTER K150PTRM * (008)
          EQUIVALENCE (KXXX (1:), K150PTRM)
*
*          CHARACTER K150PRNM * (008)         PTERM NAME
          EQUIVALENCE (KXXX (9:), K150PRNM)
*
*          CHARACTER K150BCAP * (008)         PROCESSOR NAME
          EQUIVALENCE (KXXX (17:), K150BCAP)
*
*          CHARACTER K150LTRM * (008)         BCAM APPLICATION NAME
          EQUIVALENCE (KXXX (25:), K150LTRM)
*
*          CHARACTER K150RSOA * (032)        LTERM NAME
          EQUIVALENCE (KXXX (33:), K150RSOA)
*
*          CHARACTER K150RSO0 * (001)        RSO ANNO
          EQUIVALENCE (KXXX (65:), K150RSO0)
*
*          CHARACTER K150RSOM * (007)        RSO ACTION
          EQUIVALENCE (KXXX (66:), K150RSOM)
*
*          CHARACTER K150RSOR * (004)        RSO ERROR MESSAGE
          EQUIVALENCE (KXXX (73:), K150RSOR)
*
*          CHARACTER K150RSO2 * (004)        RSO RETURNCODE
          EQUIVALENCE (KXXX (77:), K150RSO2)

```

```

*           RSO ASYN RETURNCODE
CHARACTER  K150TEXT * (072)
EQUIVALENCE (KXXX (81:), K150TEXT)
*****
CHARACTER  K151IDEF * (008)
EQUIVALENCE (KXXX (1:), K151IDEF)
*           RETURNCODE OF INVERSE
*           KDCDEF
CHARACTER  K151DMSE * (004)
EQUIVALENCE (KXXX (9:), K151DMSE)
*           DMS ERROR CODE
CHARACTER  K151FNAM * (054)
EQUIVALENCE (KXXX (13:), K151FNAM)
*           FILE NAME
CHARACTER  K151TEXT * (086)
EQUIVALENCE (KXXX (67:), K151TEXT)
*****
CHARACTER  K152COND * (003)
EQUIVALENCE (KXXX (1:), K152COND)
*           CONDITION
CHARACTER  K152MTYP * (004)
EQUIVALENCE (KXXX (4:), K152MTYP)
*           MESSAGE TYPE
CHARACTER  K152OSLP * (008)
EQUIVALENCE (KXXX (8:), K152OSLP)
*           OSI-LPAP NAME
CHARACTER  K152USER * (008)
EQUIVALENCE (KXXX (16:), K152USER)
*           USER/LSES/OSI-ASS NAME
CHARACTER  K152LTAC * (008)
EQUIVALENCE (KXXX (24:), K152LTAC)
*           TAC OR LTAC
CHARACTER  K152AAIS * (004)
EQUIVALENCE (KXXX (32:), K152AAIS)
*           ATOMIC ACTION IDENTIFIER
*           SIZE
CHARACTER  K152AAID * (064)
EQUIVALENCE (KXXX (36:), K152AAID)
*           ATOMIC ACTION IDENTIFIER
CHARACTER  K152TEXT * (053)
EQUIVALENCE (KXXX (100:), K152TEXT)
*****
CHARACTER  P001XPFU * (020)
EQUIVALENCE (KXXX (1:), P001XPFU)
*           CALLED OSI-TP FUNCTION
CHARACTER  P001XPRES * (004)
EQUIVALENCE (KXXX (21:), P001XPRES)
*           OSI-TP RETURN CODE
CHARACTER  P001XPER * (004)
EQUIVALENCE (KXXX (25:), P001XPER)
*           OSI-TP ERROR CODE
CHARACTER  P001XP1I * (004)
EQUIVALENCE (KXXX (29:), P001XP1I)
*           OSI-TP ADDITIONAL
*           INFORMATION 1
CHARACTER  P001XP2I * (004)
EQUIVALENCE (KXXX (33:), P001XP2I)
*           OSI-TP ADDITIONAL
*           INFORMATION 2
CHARACTER  P001XPCO * (004)

```

```

EQUIVALENCE (KXXX (37:), P001XPC0)
*                                     MESSAGE CORRELATOR NUMBER
CHARACTER P001TEXT * (112)
EQUIVALENCE (KXXX (41:), P001TEXT)
*****
CHARACTER P002XPFU * (020)
EQUIVALENCE (KXXX (1:), P002XPFU)
*                                     CALLED OSI-TP FUNCTION
CHARACTER P002ACPN * (008)
EQUIVALENCE (KXXX (21:), P002ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P002OSLP * (008)
EQUIVALENCE (KXXX (29:), P002OSLP)
*                                     OSI-LPAP NAME
CHARACTER P002XPRE * (004)
EQUIVALENCE (KXXX (37:), P002XPRE)
*                                     OSI-TP RETURN CODE
CHARACTER P002XPER * (004)
EQUIVALENCE (KXXX (41:), P002XPER)
*                                     OSI-TP ERROR CODE
CHARACTER P002XP1I * (004)
EQUIVALENCE (KXXX (45:), P002XP1I)
*                                     OSI-TP ADDITIONAL
*                                     INFORMATION 1
CHARACTER P002XP2I * (004)
EQUIVALENCE (KXXX (49:), P002XP2I)
*                                     OSI-TP ADDITIONAL
*                                     INFORMATION 2
CHARACTER P002XPC0 * (004)
EQUIVALENCE (KXXX (53:), P002XPC0)
*                                     MESSAGE CORRELATOR NUMBER
CHARACTER P002TEXT * (096)
EQUIVALENCE (KXXX (57:), P002TEXT)
*****
CHARACTER P003ACPN * (008)
EQUIVALENCE (KXXX (1:), P003ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P003XPRJ * (004)
EQUIVALENCE (KXXX (9:), P003XPRJ)
*                                     OSI-TP ASSOCIATION REASON
*                                     FOR REJECT
CHARACTER P003XPLT * (004)
EQUIVALENCE (KXXX (13:), P003XPLT)
*                                     OSI-TP INVALID LENGTH
CHARACTER P003TEXT * (136)
EQUIVALENCE (KXXX (17:), P003TEXT)
*****
CHARACTER P004ACPN * (008)
EQUIVALENCE (KXXX (1:), P004ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P004OSLP * (008)
EQUIVALENCE (KXXX (9:), P004OSLP)
*                                     OSI-LPAP NAME
CHARACTER P004XPRJ * (004)
EQUIVALENCE (KXXX (17:), P004XPRJ)
*                                     OSI-TP ASSOCIATION REASON
*                                     FOR REJECT
CHARACTER P004TEXT * (132)
EQUIVALENCE (KXXX (21:), P004TEXT)
*****

```

```

CHARACTER P005ACPN * (008)
EQUIVALENCE (KXXX (1:), P005ACPN)
* ACCESS-POINT-NAME
CHARACTER P005XPNS * (008)
EQUIVALENCE (KXXX (9:), P005XPNS)
* OSI-TP N-SEL OF PARTNER
CHARACTER P005XPTS * (008)
EQUIVALENCE (KXXX (17:), P005XPTS)
* OSI-TP T-SEL OF PARTNER
CHARACTER P005XPLS * (004)
EQUIVALENCE (KXXX (25:), P005XPLS)
* OSI-TP LENGTH S-SEL OF
* PARTNER
CHARACTER P005XPCS * (016)
EQUIVALENCE (KXXX (29:), P005XPCS)
* OSI-TP S-SEL OF PARTNER
* (CHAR)
CHARACTER P005XPHS * (016)
EQUIVALENCE (KXXX (45:), P005XPHS)
* OSI-TP S-SEL OF PARTNER
* (HEX)
CHARACTER P005XPLP * (004)
EQUIVALENCE (KXXX (61:), P005XPLP)
* OSI-TP LENGTH P-SEL OF
* PARTNER
CHARACTER P005XPCP * (016)
EQUIVALENCE (KXXX (65:), P005XPCP)
* OSI-TP P-SEL OF PARTNER
* (CHAR)
CHARACTER P005XPHP * (016)
EQUIVALENCE (KXXX (81:), P005XPHP)
* OSI-TP P-SEL OF PARTNER
* (HEX)
CHARACTER P005TEXT * (056)
EQUIVALENCE (KXXX (97:), P005TEXT)
*****
CHARACTER P006ACPN * (008)
EQUIVALENCE (KXXX (1:), P006ACPN)
* ACCESS-POINT-NAME
CHARACTER P006OSLP * (008)
EQUIVALENCE (KXXX (9:), P006OSLP)
* OSI-LPAP NAME
CHARACTER P006XP00 * (004)
EQUIVALENCE (KXXX (17:), P006XP00)
* OSI-TP OBJECT IDENTIFIER
* 0
CHARACTER P006XP10 * (004)
EQUIVALENCE (KXXX (21:), P006XP10)
* OSI-TP OBJECT IDENTIFIER
* 1
CHARACTER P006XP20 * (004)
EQUIVALENCE (KXXX (25:), P006XP20)
* OSI-TP OBJECT IDENTIFIER
* 2
CHARACTER P006XP30 * (004)
EQUIVALENCE (KXXX (29:), P006XP30)
* OSI-TP OBJECT IDENTIFIER
* 3
CHARACTER P006XP40 * (004)
EQUIVALENCE (KXXX (33:), P006XP40)

```

```

*                                     OSI-TP OBJECT IDENTIFIER
*                                     4
CHARACTER P006XP50 * (004)
EQUIVALENCE (KXXX (37:), P006XP50)
*                                     OSI-TP OBJECT IDENTIFIER
*                                     5
CHARACTER P006XP60 * (004)
EQUIVALENCE (KXXX (41:), P006XP60)
*                                     OSI-TP OBJECT IDENTIFIER
*                                     6
CHARACTER P006XP70 * (004)
EQUIVALENCE (KXXX (45:), P006XP70)
*                                     OSI-TP OBJECT IDENTIFIER
*                                     7
CHARACTER P006XP80 * (004)
EQUIVALENCE (KXXX (49:), P006XP80)
*                                     OSI-TP OBJECT IDENTIFIER
*                                     8
CHARACTER P006XP90 * (004)
EQUIVALENCE (KXXX (53:), P006XP90)
*                                     OSI-TP OBJECT IDENTIFIER
*                                     9
CHARACTER P006TEXT * (096)
EQUIVALENCE (KXXX (57:), P006TEXT)
*****
CHARACTER P007ACPN * (008)
EQUIVALENCE (KXXX (1:), P007ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P007OSLP * (008)
EQUIVALENCE (KXXX (9:), P007OSLP)
*                                     OSI-LPAP NAME
CHARACTER P007XPRES * (004)
EQUIVALENCE (KXXX (17:), P007XPRES)
*                                     OSI-TP RETURN CODE
CHARACTER P007XPER * (004)
EQUIVALENCE (KXXX (21:), P007XPER)
*                                     OSI-TP ERROR CODE
CHARACTER P007XP1I * (004)
EQUIVALENCE (KXXX (25:), P007XP1I)
*                                     OSI-TP ADDITIONAL
*                                     INFORMATION 1
CHARACTER P007XP2I * (004)
EQUIVALENCE (KXXX (29:), P007XP2I)
*                                     OSI-TP ADDITIONAL
*                                     INFORMATION 2
CHARACTER P007XPCO * (004)
EQUIVALENCE (KXXX (33:), P007XPCO)
*                                     MESSAGE CORRELATOR NUMBER
CHARACTER P007TEXT * (116)
EQUIVALENCE (KXXX (37:), P007TEXT)
*****
CHARACTER P008ACPN * (008)
EQUIVALENCE (KXXX (1:), P008ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P008OSLP * (008)
EQUIVALENCE (KXXX (9:), P008OSLP)
*                                     OSI-LPAP NAME
CHARACTER P008XPOS * (004)
EQUIVALENCE (KXXX (17:), P008XPOS)
*                                     OSI-TP ASSOCIATION

```



```

*                                     REFERENCE
CHARACTER P008TEXT * (132)
EQUIVALENCE (KXXX (21:), P008TEXT)
*****
CHARACTER P009ACPN * (008)
EQUIVALENCE (KXXX (1:), P009ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P0090SLP * (008)
EQUIVALENCE (KXXX (9:), P0090SLP)
*                                     OSI-LPAP NAME
CHARACTER P009XPRJ * (004)
EQUIVALENCE (KXXX (17:), P009XPRJ)
*                                     OSI-TP ASSOCIATION REASON
*                                     FOR REJECT
CHARACTER P009XPLT * (004)
EQUIVALENCE (KXXX (21:), P009XPLT)
*                                     OSI-TP INVALID LENGTH
CHARACTER P009XPOS * (004)
EQUIVALENCE (KXXX (25:), P009XPOS)
*                                     OSI-TP ASSOCIATION
*                                     REFERENCE
CHARACTER P009TEXT * (124)
EQUIVALENCE (KXXX (29:), P009TEXT)
*****
CHARACTER P010ACPN * (008)
EQUIVALENCE (KXXX (1:), P010ACPN)
*                                     ACCESS-POINT-NAME
CHARACTER P0100SLP * (008)
EQUIVALENCE (KXXX (9:), P0100SLP)
*                                     OSI-LPAP NAME
CHARACTER P010XPNS * (008)
EQUIVALENCE (KXXX (17:), P010XPNS)
*                                     OSI-TP N-SEL OF PARTNER
CHARACTER P010XPTS * (008)
EQUIVALENCE (KXXX (25:), P010XPTS)
*                                     OSI-TP T-SEL OF PARTNER
CHARACTER P010XPLS * (004)
EQUIVALENCE (KXXX (33:), P010XPLS)
*                                     OSI-TP LENGTH S-SEL OF
*                                     PARTNER
CHARACTER P010XPCS * (016)
EQUIVALENCE (KXXX (37:), P010XPCS)
*                                     OSI-TP S-SEL OF PARTNER
*                                     (CHAR)
CHARACTER P010XPHS * (016)
EQUIVALENCE (KXXX (53:), P010XPHS)
*                                     OSI-TP S-SEL OF PARTNER
*                                     (HEX)
CHARACTER P010XPLP * (004)
EQUIVALENCE (KXXX (69:), P010XPLP)
*                                     OSI-TP LENGTH P-SEL OF
*                                     PARTNER
CHARACTER P010XPCP * (016)
EQUIVALENCE (KXXX (73:), P010XPCP)
*                                     OSI-TP P-SEL OF PARTNER
*                                     (CHAR)
CHARACTER P010XPHP * (016)
EQUIVALENCE (KXXX (89:), P010XPHP)
*                                     OSI-TP P-SEL OF PARTNER
*                                     (HEX)

```



```

CHARACTER P012XPCT * (004)
EQUIVALENCE (KXXX (1:), P012XPCT)
*
CHARACTER P012XPCC * (004)
EQUIVALENCE (KXXX (5:), P012XPCC)
*
CHARACTER P012XPCV * (004)
EQUIVALENCE (KXXX (9:), P012XPCV)
*
CHARACTER P012XPBC * (004)
EQUIVALENCE (KXXX (13:), P012XPBC)
*
CHARACTER P012XPCO * (004)
EQUIVALENCE (KXXX (17:), P012XPCO)
*
CHARACTER P012TEXT * (132)
EQUIVALENCE (KXXX (21:), P012TEXT)
*****
CHARACTER P013ACPN * (008)
EQUIVALENCE (KXXX (1:), P013ACPN)
*
CHARACTER P013OSLP * (008)
EQUIVALENCE (KXXX (9:), P013OSLP)
*
CHARACTER P013XPCR * (004)
EQUIVALENCE (KXXX (17:), P013XPCR)
*
CHARACTER P013XPSR * (004)
EQUIVALENCE (KXXX (21:), P013XPSR)
*
CHARACTER P013XPND * (004)
EQUIVALENCE (KXXX (25:), P013XPND)
*
CHARACTER P013XP1B * (005)
EQUIVALENCE (KXXX (29:), P013XP1B)
*
CHARACTER P013XP2B * (005)
EQUIVALENCE (KXXX (34:), P013XP2B)
*
CHARACTER P013XP3B * (005)
EQUIVALENCE (KXXX (39:), P013XP3B)
*
CHARACTER P013XP4B * (005)
EQUIVALENCE (KXXX (44:), P013XP4B)
*
CHARACTER P013XP5B * (005)
EQUIVALENCE (KXXX (49:), P013XP5B)
*
CHARACTER P013XPOS * (004)
EQUIVALENCE (KXXX (54:), P013XPOS)
*
CHARACTER P013TEXT * (095)

```

CMX ERROR TYPE

CMX ERROR CLASS

CMX ERROR VALUE

BCAM INFOWORD

MESSAGE CORRELATOR NUMBER

ACCESS-POINT-NAME

OSI-LPAP NAME

OSI-TP NEGATIVE  
CONFIRMATION RESULTOSI-TP RESULT SOURCE FROM  
PARTNEROSI-TP NEGATIVE  
DIAGNOSTICSOSI-TP CCR V2 NOT  
AVAILABLEOSI-TP PROTOCOL VERSION  
INCOMPATIBILITYOSI-TP CONTENTION WINNER  
ASSIGNMENT REJECTEDOSI-TP BID MANDATORY  
REJECTED

OSI-TP NO REASON GIVEN

OSI-TP ASSOCIATION  
REFERENCE

```

EQUIVALENCE (KXXX (58:), P013TEXT)
*****
CHARACTER P014XPFU * (020)
EQUIVALENCE (KXXX (1:), P014XPFU)
*
CHARACTER P014ACPN * (008)
EQUIVALENCE (KXXX (21:), P014ACPN)
*
CHARACTER P014OSLP * (008)
EQUIVALENCE (KXXX (29:), P014OSLP)
*
CHARACTER P014XP2I * (004)
EQUIVALENCE (KXXX (49:), P014XP2I)
*
CHARACTER P014XPCO * (004)
EQUIVALENCE (KXXX (57:), P014XPCO)
*
CHARACTER P014TEXT * (092)
EQUIVALENCE (KXXX (61:), P014TEXT)
*****
CHARACTER P015XPFU * (020)
EQUIVALENCE (KXXX (1:), P015XPFU)
*
CHARACTER P015ACPN * (008)
EQUIVALENCE (KXXX (21:), P015ACPN)
*
CHARACTER P015OSLP * (008)
EQUIVALENCE (KXXX (29:), P015OSLP)
*
CHARACTER P015XPLN * (004)
EQUIVALENCE (KXXX (37:), P015XPLN)
*
CHARACTER P015XPSR * (004)
EQUIVALENCE (KXXX (41:), P015XPSR)
*
CHARACTER P015XPND * (004)
EQUIVALENCE (KXXX (45:), P015XPND)
*
CHARACTER P015XPIN * (004)
EQUIVALENCE (KXXX (49:), P015XPIN)
*
CHARACTER P015XP1I * (004)
EQUIVALENCE (KXXX (53:), P015XP1I)

```

```

*
*
* CHARACTER P015XP2I * (004)
EQUIVALENCE (KXXX (57:), P015XP2I)
*
* OSI-TP ADDITIONAL
INFORMATION 1
*
* CHARACTER P015XPOS * (004)
EQUIVALENCE (KXXX (61:), P015XPOS)
*
* OSI-TP ADDITIONAL
INFORMATION 2
*
* CHARACTER P015XPCO * (004)
EQUIVALENCE (KXXX (65:), P015XPCO)
*
* OSI-TP ASSOCIATION
REFERENCE
*
* CHARACTER P015XPCO * (004)
EQUIVALENCE (KXXX (65:), P015XPCO)
*
* MESSAGE CORRELATOR NUMBER
*
* CHARACTER P015TEXT * (084)
EQUIVALENCE (KXXX (69:), P015TEXT)
*****
*
* CHARACTER P016ACPN * (008)
EQUIVALENCE (KXXX (1:), P016ACPN)
*
* ACCESS-POINT-NAME
*
* CHARACTER P016OSLP * (008)
EQUIVALENCE (KXXX (9:), P016OSLP)
*
* OSI-LPAP NAME
*
* CHARACTER P016XPLN * (004)
EQUIVALENCE (KXXX (17:), P016XPLN)
*
* OSI-TP LINK
*
* CHARACTER P016XPND * (004)
EQUIVALENCE (KXXX (21:), P016XPND)
*
* OSI-TP NEGATIVE
DIAGNOSTICS
*
* CHARACTER P016XPOS * (004)
EQUIVALENCE (KXXX (25:), P016XPOS)
*
* OSI-TP ASSOCIATION
REFERENCE
*
* CHARACTER P016TEXT * (124)
EQUIVALENCE (KXXX (29:), P016TEXT)
*****
*
* CHARACTER P017XPPD * (004)
EQUIVALENCE (KXXX (1:), P017XPPD)
*
* OSI-TP PDU TYPE
*
* CHARACTER P017XP1D * (004)
EQUIVALENCE (KXXX (5:), P017XP1D)
*
* OSI-TP DIAGNOSTIC
INFORMATION 1
*
* CHARACTER P017XP2D * (004)
EQUIVALENCE (KXXX (9:), P017XP2D)
*
* OSI-TP DIAGNOSTIC
INFORMATION 2
*
* CHARACTER P017XP3D * (004)
EQUIVALENCE (KXXX (13:), P017XP3D)
*
* OSI-TP DIAGNOSTIC
INFORMATION 3
*
* CHARACTER P017TEXT * (136)
EQUIVALENCE (KXXX (17:), P017TEXT)
*****
*
* CHARACTER P018ACPN * (008)
EQUIVALENCE (KXXX (1:), P018ACPN)
*
* ACCESS-POINT-NAME
*
* CHARACTER P018OSLP * (008)
EQUIVALENCE (KXXX (9:), P018OSLP)
*
* OSI-LPAP NAME

```

```

CHARACTER P018XPPT * (004)
EQUIVALENCE (KXXX (17:), P018XPPT)
*
OSI-TP PRIITIVE TYPE
CHARACTER P018XPFS * (010)
EQUIVALENCE (KXXX (21:), P018XPFS)
*
OSI-TP FSM NAME
CHARACTER P018TEXT * (122)
EQUIVALENCE (KXXX (31:), P018TEXT)
*****
CHARACTER P019ACPN * (008)
EQUIVALENCE (KXXX (1:), P019ACPN)
*
ACCESS-POINT-NAME
CHARACTER P019OSLP * (008)
EQUIVALENCE (KXXX (9:), P019OSLP)
*
OSI-LPAP NAME
CHARACTER P019XPAP * (020)
EQUIVALENCE (KXXX (17:), P019XPAP)
*
OSI-TP APDU TYPE
CHARACTER P019XP3I * (040)
EQUIVALENCE (KXXX (37:), P019XP3I)
*
OSI-TP ADDITIONAL
*
INFORMATION 3
CHARACTER P019TEXT * (076)
EQUIVALENCE (KXXX (77:), P019TEXT)
*****
*****

```

## 4.11 Data structure KCPADF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +***
**                                     +***
**      ALL RIGHTS RESERVED                                           +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +***
**
**
**      Return Information of PADM Call                                *
**
**      Include for FOR1 : KCPADF                                    *
**
*****+***
**
**      CHARACTER  KCPADF * (44)
**
**      CHARACTER  KCRETPAD * (44)                                maxim.length of information
**
-----*
**      return information for KCOM = AI                            *
-----*
**
**      CHARACTER  KCACKINF * (44)                                acknowledge information :
**      CHARACTER  KCACKCID * 8                                  printer control id
**      CHARACTER  KCGENUID * 8                                  user id
**      CHARACTER  KCDPUTID * 8                                  DPUT id
**      CHARACTER  KCGENTIM * (9)                                generation time of message :
**      CHARACTER  KCGENDOY * 3                                  day of year
**      CHARACTER  KCGENHR * 2                                  hour
**      CHARACTER  KCGENMIN * 2                                  minute
**      CHARACTER  KCGENSEC * 2                                  second
**      CHARACTER  KCSTTIM * (9)                                desired start time (DPUT) :
**      CHARACTER  KCSTDOY * 3                                  day of year
**      CHARACTER  KCSTHR * 2                                  hour
**      CHARACTER  KCSTMIN * 2                                  minute
**      CHARACTER  KCSTSEC * 2                                  second
**      CHARACTER  KCPOSMSG * 1                                  positive acknowl. job
**

```





## 4.12 Data structure KCPAF

```

*****+***
**                                     +***
**      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992 +***
**                                     +***
**      ALL RIGHTS RESERVED                                     +***
**                                     +***
*****+***
**      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0   +***
**
**
**      Parameter Area for UTM  ( KDCSPAB )
**
**      Include for FOR1 : KCPAF
**
*****
**
**      CHARACTER  KCSPA * (44)
**
**-----**
**      parameter area                                     ( 42 bytes )
**-----**
**
**      CHARACTER  KCOP * 4
**                                     operation code
**      CHARACTER  KCOM * 2
**                                     operation modification
**      INTEGER    KCLA * 2
**                                     length of data area
**      INTEGER    KCLCAPA * 2
**                                     length of ca-program area
**      INTEGER    KCLM * 2
**                                     length of message
**      INTEGER    KCLSPA * 2
**                                     length of parameter area
**      CHARACTER  KCRN * 8
**                                     reference name
**                                     TAC/LTERM/storege area
**                                     TLS-BLOCK )
**      CHARACTER  KCFN * 8
**                                     format name
**      CHARACTER  KCLT * 8
**                                     logical terminal name
**      CHARACTER  KCUS * 8
**                                     user name
**      CHARACTER  KCPA * 8
**                                     partner application name
**      INTEGER    KCDF * 2
**                                     device feature
**      INTEGER    KCLI * 2
**                                     length of init area
**      CHARACTER  KCEXTENT * (14)
**                                     extent of parameter area
**      CHARACTER  KCDPUT * (14)
**                                     data for DPUT call :
**      CHARACTER  KCMOD * 1

```

```

*
*
* CHARACTER      KCDAY * 3          A=absolute, R=relative time
*                                         SPACE= no time
*
* CHARACTER      KCHOUR * 2         day
*
* CHARACTER      KCMIN * 2          minute
*
* CHARACTER      KCSEC * 2           second
*
* CHARACTER      KCAPRO * (14)      data for APRO call :
*
* CHARACTER      KCPI * 8           conversation identification
*
* CHARACTER      KCOF * 1           osi-tp function
*
* CHARACTER      KCPADM * (14)      data for PADM call :
*
* CHARACTER      KCACT * 3          KCOM=CS —> action
*
* CHARACTER      KCADRLT * 8        KCOM=CA —> LTERM name
*
* CHARACTER      KCSGCL * (14)      data for SIGN CL call :
*
* CHARACTER      KCLANGID * 2       language id
*
* CHARACTER      KCTERRID * 2       territory id
*
* CHARACTER      KCCSNAME * 8       coded character set name
*
* CHARACTER      KCMCOM * (42)      data for MCOM call :
*
* CHARACTER      KCPOS * 8          destination in positive case
*
* CHARACTER      KCNEG * 8          destination in negative case
*
* CHARACTER      KCCOMID * 8        complex identification
*
*
* EQUIVALENCE ( KCSPA ( 1: 4 ) , KCOP      )
* EQUIVALENCE ( KCSPA ( 5: 6 ) , KCOM      )
* EQUIVALENCE ( KCSPA ( 7: 8 ) , KCLA      , KCLCAPA )
* EQUIVALENCE ( KCSPA ( 9:10 ) , KCLM      , KCLSPA )
* EQUIVALENCE ( KCSPA ( 11:18 ) , KCRN     )
* EQUIVALENCE ( KCSPA ( 19:26 ) , KCFN      , KCLT      )
* EQUIVALENCE ( KCSPA ( 19:26 ) , KCUS      , KCPA      )
* EQUIVALENCE ( KCSPA ( 27:28 ) , KCDF      , KCLI      )
* EQUIVALENCE ( KCSPA ( 29:42 ) , KCEXTENT , KCDPUT )
* EQUIVALENCE ( KCSPA ( 29:29 ) , KCMOD     )
* EQUIVALENCE ( KCSPA ( 30:32 ) , KCDAY     )
* EQUIVALENCE ( KCSPA ( 33:34 ) , KCHOUR    )
* EQUIVALENCE ( KCSPA ( 35:36 ) , KCMIN     )
* EQUIVALENCE ( KCSPA ( 37:38 ) , KCSEC     )
* EQUIVALENCE ( KCSPA ( 29:42 ) , KCAPRO    )
* EQUIVALENCE ( KCSPA ( 29:36 ) , KCPI      )
* EQUIVALENCE ( KCSPA ( 37:37 ) , KCOF      )
* EQUIVALENCE ( KCSPA ( 29:42 ) , KCPADM    )
* EQUIVALENCE ( KCSPA ( 29:31 ) , KCACT     )
* EQUIVALENCE ( KCSPA ( 32:39 ) , KCADRLT   )
* EQUIVALENCE ( KCSPA ( 29:42 ) , KCSGCL   )

```

```
      EQUIVALENCE ( KCSPA ( 29:30 ) , KCLANGID )
      EQUIVALENCE ( KCSPA ( 31:32 ) , KCTERRID )
      EQUIVALENCE ( KCSPA ( 33:40 ) , KCCSNAME )
      EQUIVALENCE ( KCSPA ( 1:42 ) , KCMCOM )
      EQUIVALENCE ( KCSPA ( 19:26 ) , KCPOS )
      EQUIVALENCE ( KCSPA ( 27:34 ) , KCNEG )
      EQUIVALENCE ( KCSPA ( 35:42 ) , KCCOMID )
*
*
      COMMON /KDCSPAB/ KCSPA
*
*****
*
```



---

# Index

#format 16  
\*format 16  
+format 16

## A

additional data areas 6

## C

command section 10  
COMMON area  
    KDCKB 11  
    KDCSPAB 11  
COMP  
    =FOR1 13  
    =ILCS 13  
compiler-dependent points 15  
creating formats 16

## D

data declarations 3, 5  
data structure  
    KCAPROF 59  
    KCATF 61  
    KCCAF 64  
    KCCFF 68  
    KCDADF 78  
    KCDFF 80  
    KCINFF 81  
    KCINIF 86  
    KCINPF 90  
    KCMMSGF 92  
    KCPADF 135  
    KCPAF 137  
data structures 4, 12  
    for Fortran programs 8

Fortran 59

Distributed Transaction Processing (DTP) 1

DPUT call 25

### E

entry name 4

EQUIVALENCE statements 5

event exit 14

### F

formats 16

    create 16

Fortran

    data structures 59

    examples 21

    language restrictions 20

Fortran addressing aids 15

Fortran program unit 3

### I

INCLUDE element 4

### K

KCAPROF 59

KCATF 61

KCCAF 64

KCCFF 68

KCDADF 78

KCDFF 80

KCINFF 81

KCINIF 86

KCINPF 90

KCMMSGF 92

KCPADF 135

KCPAF 137

KDCROOT 10

KDCROOT table module 10

KDCS calls 12

    examples 21

KDCS communication area 6

KDCS INIT call 10

KDCS parameter area 5, 12

- 
- L**  
language restrictions 20  
linkage program 10
- M**  
main routine 10  
    compile 15  
MGET call 22  
MPUT call 24
- N**  
name prefixes 20
- P**  
program skeleton 10  
program structure 3  
program termination 13  
program unit name 4
- R**  
restrictions  
    Fortran language elements 20  
RETURN statement 3, 10, 11
- S**  
shareable data areas 6  
shareable modules 17  
SPAB 5  
STOP statement 10, 11  
STXIT events 13  
STXIT handling 13  
SUBROUTINE statement 4  
subroutines 3
- U**  
UTM application  
    example 26  
UTM Fortran application  
    link 15  
UTM functions 12  
UTM main routine 10  
UTM program unit 3





---

# Contents

<b>1</b>	<b>Preface</b> .....	<b>1</b>
1.1	Summary of contents and target group .....	2
<b>2</b>	<b>Structure of Fortran program units</b> .....	<b>3</b>
2.1	Fortran program units as subroutines .....	3
2.2	Subroutine name as the entry name .....	4
2.3	Data section of a Fortran program unit (data declarations) .....	5
2.3.1	Communication area and KDCS parameter area .....	5
2.3.2	Additional shareable data areas .....	6
2.3.3	Data structures for UTM Fortran programs .....	8
2.4	Command section of a Fortran program unit .....	10
2.4.1	Program skeleton .....	10
2.4.2	Calling UTM functions (KDCS calls) .....	12
2.4.3	STXIT handling .....	13
2.5	Event exits .....	14
2.6	Special points relating to Fortran .....	15
2.6.1	Compiling the KDCROOT main routine (FOR1) .....	15
2.6.2	Linking a UTM Fortran application (FOR1) .....	15
2.6.3	Linking a UTM Fortran application (Fortran90) .....	16
2.6.4	Creating formats with IFG .....	16
2.6.5	Shareable modules .....	17
2.6.6	Restrictions affecting the Fortran language elements under UTM .....	20
2.6.7	Fortran-specific version dependencies .....	20
<b>3</b>	<b>Examples in Fortran</b> .....	<b>21</b>
3.1	Examples of individual KDCS calls .....	21
	MGET call .....	22
	MPUT call .....	24
	DPUT call .....	25
3.2	Example of a complete UTM application .....	26
<b>4</b>	<b>Data structures for Fortran</b> .....	<b>59</b>
4.1	Data structure KCAPROF .....	59
4.2	Data structure KCATF .....	61
4.3	Data structure KCCAF .....	64
4.4	Data structure KCCFF .....	68

## Contents

---

4.5	Data structure KCDADF .....	78
4.6	Data structure KCDFE .....	80
4.7	Data structure KCINFF .....	81
4.8	Data structure KCINIF .....	86
4.9	Data structure KCINPF .....	90
4.10	Data structure KCMSGF .....	92
4.11	Data structure KCPADF .....	135
4.12	Data structure KCPAF .....	137
	<b>Index .....</b>	<b>141</b>

---

# *open*UTM V4.0 (BS2000/OSD)

## Supplement for FORTRAN

### *Target group*

Programmers of UTM FORTRAN applications

### *Contents*

Translation of the KDCS program interface into the FORTRAN language, and all the information required by programmers of UTM FORTRAN applications

### *Applications*

BS2000 transaction processing

**Edition: February 1997**

**File: UTM\_FOR.PDF**

BS2000 is registered trademarks of Siemens Nixdorf Informationssysteme AG.

Copyright © Siemens Nixdorf Informationssysteme AG, 1997.

All rights, including rights of translation, reproduction by printing, copying or similar methods, even of parts, are reserved.

Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Delivery subject to availability; right of technical modifications reserved.





## Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at

[http://ts.fujitsu.com/...](http://ts.fujitsu.com/)

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter

[http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009