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.
Universal	... 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.
Transaction	... because *open*UTM guarantees complete global transaction security in accordance with the classical ACID properties of atomicity, consistency, isolation and durability.
Monitor	... 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 *open*UTM applications in Pascal-XT in their work. It is a supplement to the *open*UTM manual "Programming Applications with KDCS for COBOL, C and C++".

A basic knowledge of the operating system and *open*UTM, as well as of the core manual "Programming Applications with KDCS for COBOL, C and C++" is required. For more detailed information, the *open*UTM 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 Pascal-XT program units.

It provides sample programs written in Pascal-XT for individual KDCS calls and for the event service MSGTAC, as well as an example for a complete *open*UTM application.

The Pascal-XT data structures are listed in chapter “Data structures for Pascal-XT” on page 67ff).

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 Pascal-XT program units

This chapter tells you

- how to write a Pascal-XT program unit as a subroutine
- what UTM Pascal-XT packages are, as well as how to program a KDCS call in Pascal-XT
- what special features and restrictions apply to Pascal-XT program units.

2.1 Pascal-XT program units as subroutines

The following topics are dealt with in this section:

- specification and implementation of UTM Pascal-XT program units
- package specification
- constants and data structures for UTM Pascal-XT programs
- compilers, runtime systems and generation options

2.1.1 Specification and implementation of UTM Pascal-XT program units

UTM program units, including event exits, are subroutines of the UTM main routine. This has the following consequences for the structure of these programs:

– UTM Pascal-XT program units must be written as ENTRY procedures and must therefore be implemented in packages.

– The names of the ENTRY procedures (procedure identifiers) must, in their first 8 characters, match the program unit names specified in the KDCDEF statement PROGRAM.

– The parameter list of the ENTRY procedures must contain at least two formal parameters (for the communication area and the standard primary working area SPAB); if additional storage areas are used which were defined with the KDCDEF statement AREA, the parameter list must also contain the appropriate formal parameters. These parameters in the ENTRY procedure declarations must be defined as variable parameters ("VAR"...).

You can implement more than one ENTRY procedure (program unit) in the same package.

For compatibility purposes, and in order to work with error-free entries, it is advisable to import the data structures and constants defined by UTM from the package specifications of the library SYSLIB.UTM.040.PASC.

The data structures and constants are described in detail in the following section. Descriptions of how to use them in the individual calls can be found in the Core Manual "Programming Applications with KDCS for COBOL, C and C++" .

2.1.2 Package specification

In the package specification you declare the constants, data structures and procedures that are "visible on the outside", i.e. at least

– the structures of the communication area (KB) and the standard primary working area SPAB (and, if required, other UTM storage areas defined at generation with the KDCDEF statement AREA), as type declarations

– the program units you wish to implement in this package, as ENTRY procedures

– file variables you have entered in the program parameter list, as variable declarations.

2.1.3 Constants and data structures for UTM Pascal-XT programs

In order to structure data areas, packages containing these constants and data structures are supplied with *open*UTM. The table below gives you an overview of the scope and function of these predefined packages.

Package name **Most important data types**	**Contents and meaning**
KCKBL	Defines the following: – the KDCS operation codes. These are defined as symbolic constants. Using these constants guarantees the validity of the operation codes.
KCKB	– the communication area header, the structure of which is defined by UTM. This contains: - current data of the service and program - return information following a call to UTM. This KB header can be followed by a program area, which you must define yourself (see examples).
KCPAL	– the KDCS parameter area, which accepts the parameters of a call to UTM. Generally, the parameter area is placed at the start of the standard primary working area (SPAB), the structure of which you must define yourself (see examples).
KCATL (FIELD_ATTRIBUTE_PACKAGE)	Defines the symbolic names which can be used to modify the attribute fields of formats when working with +formats. Constant names are defined for a number of standard attribute combinations.
KCDFL	Defines the KDCS screen functions. When supplying values for the KCDF field of the KDCS parameter area, you can use these constant definitions if you request specific functions of a terminal (practicable only if the partner is a terminal).

Package name **Most important data types**	**Contents and meaning**
KCINL KCINFL	Data structure for the information supplied by the UTM call. If you define a global message area (for a program unit), you should declare KCINFL as one of the variants for this RECORD type.
KCMSL	Provides data structures for the UTM messages. You need these if you wish to interpret messages in an MSGTAC routine.
KCINPL	Data structure for the event exit INPUT. KCINPL contains input and output parameters; the output parameters determine the effect of the input.
KCDADL	Data structure for the DADM call. You should place this data structure over the message area if you want to use the call DADM RQ.
KCPADL	Data structure for the PADM call.You should place this data structure over the message area if you want to use the call PADM AI or PI.
KCAPROL	Defines an optional second parameter area for the APRO function call. KCAPROL is used for selecting specific OSI TP function combinations.
KCINIL	Defines a second parameter area for the INIT PU function call. In this parameter area UTM returns the information requested with INIT PU.
KCCFL	Defines the second parameter transferred by UTM with the event exit INPUT. In this parameter UTM passes the contents of the control fields of screen formats to the program unit. For this reason this second parameter is also known as the control fields area.

The packages you wish to use must be imported into your package by means of a WITH clause. Names in the imported packages which you wish to employ directly, i.e. without qualifying them fully, must be included in a USE clause.

Examples of the declarations for your data areas can be found in the examples in the following section and on page 33ff.

A list of these packages is provided in chapter “Data structures for Pascal-XT” on page 67ff.

2.2 Compilers, runtime systems and generation options

The following table shows the compilers, runtime systems and generation options which you can use to create Pascal-XT program units and execute them in a UTM user program.

The first column of the table contains all the compiler versions that can be used to create the object modules of the program unit.
The second column contains the runtime system that you may have to use when working with the corresponding compiler.
The third column contains the value of the COMP parameter of the UTM generation statement PROGRAM that has to be specified when using the corresponding compiler.

Pascal-XT compiler	Runtime system	COMP=
V2.1	V2.1	PASCAL-XT
V2.1	V2.2	ILCS
V2.2	V2.2	ILCS

Only one Pascal-XT runtime system may be linked into a UTM application.

Pascal-XT uses the ILCS interface. This enables program units from two or more source codes of different programming languages to be linked up. A list of all compilers and runtime systems that permit mixing can be found in the Release Notes.

2.3 Naming conventions

The names of the declared ENTRY procedures must be unique in their first 8 characters, as only the first 8 characters are entered in the external address table of the generated object modules.

The name of the package must also differ from the names of the ENTRY procedures in its first 8 characters, so as to avoid conflicts when linking the application.

All names beginning with "KDC", "KC" or "I" are reserved.

All other names can be assigned freely in accordance with the rules of the Pascal-XT language.

2.4 Declarations

The following topics are described in this section:

– declarations of the ENTRY procedures
– type declarations
– data areas as Pascal-XT packages

2.4.1 Declaration for the ENTRY procedures

You must declare every program unit as an ENTRY procedure as follows:

```
entry procedure tpname (var kb: kckbc; var spab: kcspab
                        [;var p1: id_p1; ... var pn: id_pn]);
```

tpname	is the name of the program unit specified in the PROGRAM statement; it must be unique in the first 8 characters.
kb	is the name of the communication area (of the type *kckbc*; see above).
spab	is the name of the standard primary working area (of the type *kcspab*; see above).
p1 ... pn	are the names of the additional data areas defined with the AREA statement and translated as an Assembler CSECT; their structures must have been declared beforehand with the names *id p1* ... *id pn*. The order of these formal parameters must correspond to the order of the AREA statements. All areas up to the last one used here must be included in the parameter list. If none of these areas is used in the program unit *tpname*, the entry can be omitted.

2.4.2 Type declarations

The parameters currently used to call a UTM program unit are structured data fields. You must, therefore, declare appropriate data types for the formal parameters of the ENTRY procedures.
These comprise at least the communication area (KB) and standard primary working area (SPAB). The first part of these structures is defined by *open*UTM. The declarations for these fixed parts are contained in the package specification KCKBL, which is supplied in the library SYSLIB.UTM.040.PASC under the name "KCKBLS". To make program units compatible and enhance their readability, it is advisable to import these declarations (in a WITH clause.)

Example

```
with Kckbl;                                              1)
from Kckbl use kckb, kcpal;                              2)
  ...
package application;
   ...
type
  kckbc = record                                         3)
    kb head: kckb;                                       4)
    kb any: packed array[1..22] of char;                 5)
    kb pt of ddepart: packed array [1..2] of char;       5)
    kb destination: packed array [1..2] of char;         5)
    kb flightday: packed array [1..5] of char;           5)
    kb flightno1: packed array [1..5] of char;           5)
    kb flightno2: packed array [1..5] of char;           5)
  end;
  iforma = record                                        6)
     ...
  end;

  kcspab = record                                        7)
    kcpac: kcpal;                                        8)
    nb: iforma;                                          6)
  end;
end.
```

1) The package specification KCKBL is imported.

2) The names given here can be used without selectors (e.g. KCPAL instead of KCKBL.KCPAL).

3) Type declaration for the communication area.

4) Communication area header (structure defined by UTM).

5) Application-specific program area.

6) Data structure of a previously created screen format.

7) Structure of the SPAB.

8) Parameter area.

In addition to the control area and the SPAB, you can also define up to 99 further storage areas as procedure parameters for the ENTRY procedures. These can then be used as public data areas within a UTM application. These areas can be located in an application-global or application-local common memory pool, or be statically linked to the main routine.

You create the object module required for linking as an Assembler CSECT, whose length you define with DS statements. (You can define static presettings with the aid of DC statements.) The type declaration describing the structure of this area must either be contained in the package specification of the program units which use this area or be imported from another package specification.

Example

```
VREC    CSECT
        DC     H'4'
        DC     CL2'  '
        DS     2044C
        END
```

During generation, you must define this area by means of the AREA statement (not the PROGRAM statement). The type of area (local or global) is also defined here.

When a program unit is called, the areas thus defined are transferred in the order of the AREA statements. When declaring the ENTRY procedures, you must take this order into account in the procedure parameter list. If these areas are located in common memory pools, the program units alone are responsible for synchronizing accesses.

Note

This function is not part of DIN standard 66 265.

2.4.3 Data areas as Pascal-XT packages

The package concept of Pascal-XT also offers the option of defining common data areas as global variables in separate packages.

The package specification contains:

– type and variable declarations for those areas which you wish to access directly from your program units

– declarations for private pointer types for areas whose structure you wish to conceal

– declarations for access procedures for the private data types (and, if required, declarations for other procedures and functions for processing these areas).

The package body contains:

– the declaration for the domain type of the private pointer types

– the blocks of the procedures and functions declared in the package specification;

– in the outmost block of the body, statements (if required) for initializing the package variables.

You import the specifications of these packages into the packages for your program units using a WITH clause (and, if required, USE clauses). Here you must observe the following rules:

– at the time of generation, the areas thus defined must not be named in either a PROGRAM statement or an AREA statement

– these areas are **not** transferred as current parameters when a program unit is called

If such areas are only used locally, you can link these packages together with the ROOT module. However, you can also place such packages in a (local or global) common memory pool (see also page 19).

Rules for forming package names

In this context you must observe the rules followed by the Pascal-XT compiler in forming package names:

– The package name is abbreviated to 7 characters; if it is shorter than 7 characters, it is padded to this length with "#".

– The eighth character is used to differentiate the modules, e.g. "C“ for the code module and "D“ for the data module.

– Underscore characters are replaced by "#".

Example

```
package AREAS;
type
  vrec = record
    record length: short integer;
    dummy: short integer;
    data: packed array[1..2044] of char;
  end;

var
  area1 : vrec;
  area2 : vrec;
end.

package body AREAS;
begin
  with area1 do begin record length := 4; dummy := #4040 end;
  with area2 do begin record length := 4; dummy := #4040 end;
end.
```

In this example the names generated by the Pascal-XT compiler are:

```
AREAC          for the code module
AREAD          for the data module
```

You must link in both modules or load them into the memory pool.

2.5 Package body

The package body can contain not only the outmost blocks of the ENTRY procedures which you have declared in the package specification, but also constant, type and variable declarations, as well as procedures and functions which you only wish to use locally within this package (e.g. INLINE procedures to simplify the formulation of the KDCS calls).

When implementing the ENTRY procedures, you need only to follow a few transaction processing rules, as described in detail in the section dealing with the structure and use of UTM programs in the Core Manual "Programming Applications with KDCS for COBOL, C and C++" . These concern:

– reentrant capability
– strict dialog (in dialog programs).

You do not need to bother about reentrancy for shared code. Pascal-XT programs and procedures are always reentrant.

Calling UTM functions

Communication between your program units and the UTM main routine takes place exclusively by calling the external procedure "KDCS".

When this procedure is called, UTM expects at least one current parameter, the address of the parameter area, which you must supply prior to calling KDCS by entering the operation code and, if required, additional values.

For most KDCS operations, the address of the message area is also requested as a second parameter.

In Pascal-XT (in contrast to COBOL), it is not possible to declare or call procedures with a variable number of parameters. In addition, the data types of the current and formal parameters must be compatible (see the "Pascal-XT" Reference Manual). There are two ways of doing this:

– You can declare a data type (e.g. *kcnb*) for the message area as the RECORD type; you then formulate as variants of this RECORD type all the message structures you use. You then declare the KDCS procedure as follows:

```
procedure KDCS (var p: kckbl.kcpal; var nb: kcnb); COBOL;
```

(kcpal is the RECORD type declared for the parameter area in the KCKBL package).

– You can write special procedures in Pascal-XT for the various KDCS operations you use. In the declaration part of these procedures you must declare the KDCS procedure.

Example

```
function INIT: boolean;
   procedure KDCS (var p: kckbl.kcpal); COBOL;
begin
   spab.kcpac.kcop := kckbl.init;
   kdcs (spab.kcpac);
   init := (kckbc.kb_header.kcrccc = '000');
end (* INIT*);

function GET_LINE (line: string): boolean;
   procedure KDCS (var p: kckbl.kcpal; var nb: string); COBOL;
begin
   with spab.kcpac
   do begin
      kcop := kckbl.mget;
      kcmf := '        '; (* activate line mode *)
   end;
   kdcs (spab.kcpac, line);
   put_line := (kckbc.kb_header.kcrccc = '000');
end (*GET_LINE*);
```

The type and variable names in this example are taken from the examples in the previous section.

Note

The directive COBOL is mandatory in the declaration for the KDCS procedure. This is because *open*UTM requires the current parameters to be transferred in a COBOL-compatible form when the KDCS procedure is called, and this is only possible in conjunction with the COBOL directive.

2.6 Event exits

The event exits INPUT, START, SHUT and VORGANG ("service") must not contain any KDCS calls. They must be exited via the normal end of procedure.

Event exit START

If the START program unit detects an error (e.g. an attempt to open a non-existent file) and the start has to be terminated for this reason, the event exit must be able to deal with an abnormal program termination (TERM macro with relevant return code).
To this end, you must write an Assembler procedure, which you call in this case (as an external or internal procedure; see the "Pascal-XT" User Guide). Calling SET_RETURN_CODE from the predefined package BS2000CALLS is not sufficient here.

Event exit SHUT

The event exit SHUT is called

– on termination of an application program

– when PEND ER occurs, as a result of an STXIT event in a Pascal-XT program unit which is not handled (in an exception section)

– when application program exchange is terminated.

2.7 Special points relating to Pascal-XT

This section tells you:

– what you have to bear in mind in connection with linkage
– which modules can be loaded as shareable and how to generate shared code
– how to create and use Pascal-XT addressing aids
– how to work in extended line mode

2.7.1 Notes on linkage

Two options are available: either link the Pascal-XT runtime system statically to the UTM application program or make the shareable part shareable.

When statically linking the Pascal runtime system, you should bear in mind that the CSECT names ILMSINI and IMLEND in the Pascal runtime system and in the common runtime environment (CRTE) are identical. As a result, additional measures are necessary when linking the UTM application program. These vary depending on whether you are using BINDER or TSOSLNK.

BINDER

The following linkage section is required when linking the UTM application program before the runtime systems are linked in:

```
//INCLUDE-MOD  LIB = $<userid1>.PASLIB-XT
      //        , ELEMENT =                                         -
      //   ( IP@#MA2C ,                                             -
      //     IMLDATA  , IMLDCOS  , IMLDEXP  , IMLDLOG  ,            -
      //     IMLDSIN  , IMLDSQR  , IMLEND   , IMLSINI  )
//MODIFY-SYMBOL-VISIBILITY  SYMBOL-NAME = ( IMLSINI , IMLEND )      -
      //                  ,VISIBLE     = NO
```

This is followed by:

```
// RESOLVE-BY-AUTOLINK   LIB=$<userid2>.SYSLNK.CRTE.PARTIAL-BIND
// RESOLVE-BY-AUTOLINK   LIB=$<userid1>.PASLIB-XT
// RESOLVE-BY-AUTOLINK   LIB=$<userid2>.SYSLNK.UTM.040.SPLRTS
```

TSOSLNK

In this case you have to prelink a module (PASPART), whose CSECTs (IMLEND and IMLSINI) have to be renamed with the aid of LMS.
When linking the UTM application program you should remember that, before the runtime system is linked in with a RESOLVE statement, you have to add the module (PASPART) to the UTM application program using an INCLUDE statement.

```
/START-PROGRAM $TSOSLNK
MODULE    PASPART     ,LET=Y
MODULE    PASPART     ,CMAP=ALL
MODULE    PASPART     ,LIB=<private-lib>
COMMENT
  LINK-SYMBOLS *KEEP
COMMENT
  NCAL
COMMENT
INCLUDE    IP@#MA2C              ,$<userid>.PASLIB-XT
INCLUDE  ( IMLDSQR , IMLDCOS , IMLDSIN , IMLDLOG ) ,$<userid>.PASLIB-XT
INCLUDE  ( IMLDEXP , IMLDATA , IMLSINI , IMLEND  ) ,$<userid>.PASLIB-XT
BIND
/
/START-LMS
/  SEND-ST 'O <private-lib>   ,MODE=UPDATE'
/  SEND-ST 'MOD-ELEM ELEMENT=*LIB(ELEM=PASPART,TYPE=R)'
/  SEND-ST 'REN IMLSINI  , NEW=PASDUMY1'
/  SEND-ST 'REN IMLEND   , NEW=PASDUMY2'
/  SEND-ST 'END-MOD'
/  SEND-ST 'END'
/
```

When linking the UTM application program, the statement sequence for linking in the runtime systems looks like this:

```
INCLUDE    PASPART,<private-lib>
RESOLVE           ,$<userid3>.SYSLNK.UTM.040.SPLRTS
RESOLVE           ,$<userid2>.PASLIB-XT
RESOLVE           ,$<userid1>.SYSLNK.CRTE.PARTIAL-BIND
```

2.7.2 Shareable modules

The following modules can be loaded as shareable:

- program units (more precisely: the code modules of the packages in which the program units are implemented)
- formats
- the formatting routine MFHSROUT
- the database connection module, provided it is shareable
- the message module KCSMSGS
- the code and data modules of the packages with which you have defined shareable data areas
- the Pascal-XT runtime system
  You should bear in mind that the runtime system contains an external reference to an ILCS data module (IT0INITS) and that this reference has to be provided for (see “Example of the generation of shared code” on page 20).

You may only declare the data modules to be shareable if read access only is permitted.

*open*UTM supports the following methods of loading modules as shareable:

- shareable modules are loaded in class 3/4 memory (up to and including BS2000/OSD V2.0, with ADD-SHARED-PROGRAM)
- shareable modules are loaded as nonprivileged subsystems
- shareable modules are loaded in the common memory pool in user memory (class 6 memory); see the *open*UTM-Manual "Concepts and Functions").

Shareable parts of the Pascal-XT runtime system can be loaded as a subsystem or in a common memory pool generated by UTM (see “Example of the generation of shared code” on page 20).

When compiling a package, the Pascal-XT compiler always generates three modules. The names of these modules are formed in accordance with the rules laid down in “Rules for forming package names” on page 12. These are the following three modules:

- a code module, e.g. with the name ECHO##C, which is always shareable and reentrant.
- a data module, e.g. with the name ECHO##D, which is **not** reentrant. This also contains the names of the ENTRY procedures defined in this package (abbreviated to 8 characters); underscore characters in the names of the ENTRY procedures are replaced by "#".
- a test table module, e.g. with the name ECHO##T.
  The debugging aid PATH enables the symbolic debugging of a Pascal-XT program unit in interactive mode. Precisely how this is done is described in the "Pascal-XT" User Guide.

All modules generated when compiling the source program have to be stored in a module library, including those parts of the program units which are not shareable.

Example of the generation of shared code

The following example is designed to show how to create shared code for program units and the Pascal-XT runtime system. The example contains two variants: one uses the BLS (Binder-Loader-Starter) interface, the other works without BLS.

The example reproduced under “Procedure for creating shared code” on page 23 shows the creation of shared code for the program unit "ECHO". The program unit contains two entry points (ECHOSYN and ECHOASYN).

In addition to the program unit, an Assembler source PASDUMMY is required in order to deal with the external reference to the ILCS procedure IT0INITS in the Pascal runtime system. The Assembler source PASDUMMY consists of the following lines:

```
IT0INITS CSECT PUBLIC
*
IT0INITS AMODE ANY
IT0INITS RMODE ANY
*
         PRINT GEN
*
         LA    15,1
         L     0,76(6)
         BR    14
*
         END   IT0INITS
```

UTM supplies this Assembler source in the library SYSLIB.UTM.040.EXAMPLE.

The shared code (BLS: public slice of PASCAL-XT-PROGRAM-UNITS-RTS;
non-BLS: module PASHARED) is loaded in a common memory pool "MPOOL001".

Depending on whether or not the BLS interface is used, the KDCDEF generation segment looks like this:

BLS:

```
LOAD-MODULE PASCAL-XT-PROGRAM-UNITS-RTS                          -
                     , LOAD-MODE   = (POOL , MPOOL001 , STARTUP )-
                     , LIB         = &(SHARED-MODULE-LIB)        -
                     , VERSION     = 001
PROGRAM     ECHOSYN  , LOAD-MODULE = PASCAL-XT-PROGRAM-UNITS-RTS -
                     , COMP        = ILCS
PROGRAM     ECHOASYN , LOAD-MODULE = PASCAL-XT-PROGRAM-UNITS-RTS -
                     , COMP        = ILCS
*
TAC ECHOSYN  , TYPE = D , PROGRAM = ECHOSYN
TAC ECHOASYN , TYPE = A , PROGRAM = ECHOASYN
```

Non-BLS:

```
DEFAULT PROGRAM LIB = &(PRIVATE-MODULE-LIB)
PROGRAM ECHOSYN  , COMP = ILCS , LOAD = STARTUP
PROGRAM ECHOASYN , COMP = ILCS , LOAD = STARTUP
MODULE PASHARED, LIB = &(SHARED-MODULE-LIB), LOAD = ( POOL , MPOOL001 )
    ENTRY   ECHO###C                        , LOAD = ( POOL , MPOOL001 )
    ENTRY   IP@#RT2C                        , LOAD = ( POOL , MPOOL001 )
*
TAC ECHOSYN  , TYPE = D , PROGRAM = ECHOSYN
TAC ECHOASYN , TYPE = A , PROGRAM = ECHOASYN
```

In a generation that does not support the BLS interface, the ENTRY IP@#RT2C must always be specified.

The names "&(PRIVATE-MODULE-LIB)" and "&(SHARED-MODULE-LIB)" correspond to the names in the following procedure.

After the KDCDEF generation run the Assembler source PASDUMMY is assembled. Then the Pascal sources are compiled.

Following this, the resultant objects are linked. The result varies depending on whether or not BLS is used:

BLS:

The modules are linked to form an LLM (link and load module) with the name PASCAL-XT-PROGRAM-UNITS-RTS, which is split into a public slice and a private slice.

Non-BLS:

The modules are linked to form a private part (PAPRIVAT) and a shared part (PASHARED), both with the associated runtime system modules.
"ECHO###D" of the sample source and of the data parts of the associated UTM-Pascal packages is additionally linked to the private part of the runtime system modules (all IP@*xxxx*D, where *xxxx* is freely selectable).

The shared part (PASHARED) consists of the following components:

– the code runtime system modules (all IP@*xxxx*C, where *xxxx* is freely selectable)

– runtime system functions for certain mathematical routines (all IML*xxxx*, where *xxxx* is freely selectable)

– the shared parts of the program units (ECHO###C, KCKBL##C)

– the module IT0ENTR from the library SYSLNK.CRTE.IT0ENTR;
the IT0ENTR version must match the linked-in CRTE.

The CSECTs IT0ENTR, IT0INITS, IMLSINI and IMLEND must be made invisible for subsequent link operations in order to prevent duplicates being detected or external references being resolved incorrectly in the course of dynamic linkage. For object modules, this is done by renaming the CSECTs with the aid of LMS after linkage.
In the case of LLMs it is enough to make the CSECTs invisible with the aid of the linkage editor.

Notes

When using the BLS interface, it is not necessary to link the program units together using the Pascal-XT runtime system. All that needs to be done is link all Pascal-XT program units dynamically (if necessary, in other load modules) to the UTM application program.
If there is a Pascal-XT runtime system module and additional load modules with Pascal-XT modules, then during generation you have to define the load module with the Pascal-XT runtime system as the first load module in the KDCDEF input. The load module with the Pascal-XT runtime system must not be replaced during operation.

If you are not working with BLS, all modules of the Pascal-XT program units of the UTM application program must be linked either to the private part (non-shareable modules) or to the shared part (shareable modules).
If the UTM application program is linked with the linkage editor TSOSLNK and the TSOSLNK-PROGRAM statement, the dynamically linked modules may only leave the following V-type constants open (the corresponding CSECTs and ENTRYs are in root code): KDCS, KDCFHS, KDCSCUR, KDCDATF and KDCERRE.
If you want to use an additional CSECT or ENTRY, e.g. a statically linked Assembler subroutine, you have to use the MODULE statement instead of the TSOSLNK-PROGRAM statement and start the UTM application program with the following statements:

```
START-PROGRAM  FROM-FILE=*MODULE(                        -

                  LIBRARY  = <lib-name>                  -
                 ,ELEMENT  = <element-name>              -
                 ,RUN-MODE = *STD                        -
                 )
```

Procedure for creating shared code

The following procedure has been designed so that it can be used with or without the BLS interface, as required.

```
/SET-PROCEDURE-OPTIONS                                  -
/       ,DATA-ESC             = STD                     -
/       ,IMPLICIT-DECLARATION = NO
/
/ " ---------------------------------------------------------------- "
/ "                                                                  "
/ " Procedure:  MAKE-PASCAL-SHARED-CODE                              "
/ "                                                                  "
/ "                                                                  "
/ " Purpose:                                                         "
/ "   This procedure generates one private and shareable part by     "
/ "   a user-owned program unit and the Pascal-XT runtime system.    "
/ "   The shareable part may be loaded in a common memory pool by    "
/ "   using UTM generation statements.                               "
/ "                                                                  "
/ " Requirements:                                                    "
/ "    ILCS must be initialized ( This task is done by UTM ).        "
/ "                                                                  "
/ "    An assembler program unit, which is called PASDUMMY. It must  "
/ "    contain a ITOINITS csect, which must be used by the           "
/ "    shareable part of the Pascal-XT runtime system to simulate    "
/ "    the ILCS ITOINITS module, if ILCS is already initialized.     "
/ "                                                                  "
/ "    A program unit with SPEC=ECHOS and BODY=ECHOB, which is       "
/ "    contained with the dummy source in the &SOURCE-LIB. This      "
/ "    program unit contins the entries ECHOSYN and ECHOASYN         "
/ "    (UTM known by the KDCDEF statements PROGRAM ).                "
/ "                                                                  "
/ "    This procedure runs under SDF-P, LMS ( from version 2.0A )    "
/ "    Assembh ( V1.2A ) and TSOSLNK. If you want to run this        "
/ "    procedure in an other environment, you have to adapt this     "
/ "    procedure.                                                    "
/ "                                                                  "
/ " Calling:                                                         "
/ "    CALL-PROCEDURE MAKE-PASCAL-SHARED-CODE,P-P=(ACTION = ... )    "
/ "     ACTION = A compiles the private PASDUMMY module (ITOINITS).  "
/ "            = C compiles the Pascal-XT sources.                   "
/ "            = L binds the runtime system, the private ITOINITS    "
/ "                module and the Pascal-XT program unit. The        "
/ "                bind section delivers a private part (PAPRIVAT)   "
/ "                and a shareable part (PASHARED). (BLS = 'N'.)     "
/ "                For the case BLS = 'Y' the procedure provide      "
/ "                the LLM PASCAL-XT-PROGRAM-UNITS-RTS with a        "
/ "                private and public slice.                         "
/ "     BLS =  Y/N                                                   "
/ "                                                                  "
/ "     ITOENTR-LIB = The ITOENTR modul containing within the CRTE   "
/ "              library. It must be the same CRTE, which is binding "
/ "              with the UTM application.                           "
/ "     MODULE-LIB = PLAM-library, which saves the assembler object  "
/ "              and the compiled Pascal-XT program unit.            "
/ "     PASCAL-COMPILER = Filename of the  Pascal-XT compiler.       "
/ "     PASCAL-RTS      = Filename of the Pascal-XT runtime system   "
/ "                       library.                                   "
/ "     PASCAL-UTM-PACK = Filename of the Pascal-XT packages         "
/ "                       consisting of the UTM data structures.     "
/ "     PRIVATE-MODULE-LIB = PLAM library, in which the private part "
/ "              (PAPRIVAT) is written by the TSOSLNK (BLS = 'N').   "
/ "              If BLS = 'Y' this library is not used.              "
/ "     SHARED-MODULE-LIB  = PLAM library, in which is written the   "
/ "              shareable part (PASHARED) by the TSOSLNK (BLS='Y'). "
/ "              In the case  BLS = 'Y' the LLM                      "
/ "              PASCAL-XT-PROGRAM-UNITS-RTS is written in this      "
/ "              library.                                            "
/ "     SOURCE-LIB = This PLAM library consist of the PASDUMMY and   "
/ "                    the Pascal-XT sources.                        "
/ " ---------------------------------------------------------------- "
/
/BEGIN-PARAMETER-DECLARATION
/ DECL-PAR  ACTION                   ( 'ACL'                  , STRING  )
/ DECL-PAR  BLS                      ( 'Y'                    , STRING  )
/ DECL-PAR  ITOENTR-LIB         ( '$TSOS.SYSLNK.CRTE.ITOENTR', STRING  )
/ DECL-PAR  MODULE-LIB               ( 'LIB.TP.PRELINK.PIN'   , STRING  )
/ DECL-PAR  PASCAL-COMPILER          ( '$PASCAL.PASCAL-XT'    , STRING  )
/ DECL-PAR  PASCAL-RTS               ( '$PASCAL.PASLIB-XT'    , STRING  )
/ DECL-PAR  PASCAL-UTM-PACK     ( '$UTM.SYSLIB.UTM.040.PASC' , STRING  )
/ DECL-PAR  PRIVATE-MODULE-LIB       ( 'PIN.R.LIB'            , STRING  )
/ DECL-PAR  SHARED-MODULE-LIB        ( 'LIB.TP.SHARE.PIN'     , STRING  )
/ DECL-PAR  SOURCE-LIB               ( 'PIN.SRC.LIB'          , STRING  )
/END-PARAMETER-DECLARATION
/
/
/  ACTION = UPPER-CASE( ACTION )
/
/  TCHNG OFLOW = ACK
/
/  WORK:   BEGIN-BLOCK   DATA-INSERTION = YES
/
/  IF ( WILDCARD( ACTION, '*A*' )  )
/
/
```

```
/START-ASSEMBH
/   SEND-ST 'COMPILE  SOURCE=*LIB-ELEM(LIB=&(SOURCE-LIB),ELEM=PASDUMMY)-
/                     ,MODULE-LIBRARY=&(MODULE-LIB)'
/   SEND-ST 'END'
/
/  END-IF
/
/
/  IF ( WILDCARD( ACTION, '*C*' )  )
/
/ASS-SYSDTA *SYSCMD
/START-PROG      &(PASCAL-COMPILER)
/  SEND-ST 'DEF-PROJ ##.ECHO-DIRECTORY'
/  SEND-ST 'MC LISTING=*DUMMY,MOD-LIB=&(MODULE-LIB),DEBUG=ON'
/  SEND-ST 'C (&(PASCAL-UTM-PACK),KCKBLS)'
/  SEND-ST 'C (&(PASCAL-UTM-PACK),KCKBLB)'
/  SEND-ST 'C (&(SOURCE-LIB),ECHOS)'
/  SEND-ST 'C (&(SOURCE-LIB),ECHOB)'
/  SEND-ST 'END'
/
/
/
/  END-IF
/
/  IF ( WILDCARD( ACTION, '*L*' )  )
/
/  IF ( BLS == 'Y' )
/
/  START-PROGRAM                                                    -
/            FROM-FILE = $BINDER
//START-LLM-CREATION                                                -
//     INTERNAL-NAME    = PASCAL-XT-PROGRAM-UNITS-RTS               -
//    ,INTERNAL-VERSION = 001                                       -
//    ,SLICE-DEFINITION = BY-ATTR( PUBLIC = YES )
//MODIFY-MAP-DEFAULT  PROGRAM-MAP = PAR( DEFINITION = ALL           -
//                                      ,INVERT   = ALL             -
//                                      ,REFERENC = ALL          )  -
//                                      ,UNRESOLVED  = SORTED( WX=NO) -
//                                      ,SORTED-PRO  = YES          -
//                                      ,DUPLICATE   = YES          -
//                                      ,OUTPUT      = LIST.LINK.PASCAL
//REMARK
//REMARK +-----------------------  Public  slice  --------------------+
//REMARK
//REMARK Own shared code modules
//   INCLUDE-MOD  LIB = &(MODULE-LIB)                               -
//               ,ELE = (  ECHO###C , KCKBL##C )
//REMARK
```

```
//REMARK RTS shared code modules
//   INCLUDE-MOD  LIB = &(PASCAL-RTS) , ELEMENT =                       -
      //   ( IP@$DM2C , IP@$ED2C , IP@$LI2C , IP@$LO2C ,                -
      //     IP@$ME2C , IP@$ML2C , IP@$SF2C , IP@$ST2C ,                -
      //     IP@$SY2C , IP@#CN2C , IP@#ER2C , IP@#HE2C ,                -
      //     IP@#IL2C , IP@#IN2C , IP@#LO2C , IP@#MA2C ,                -
      //     IP@#MM2C , IP@#OP2C , IP@#OU2C , IP@#PA2C ,                -
      //     IP@#PT2C , IP@#RE2C , IP@#RT2C , IP@#ST2C ,                -
      //     IP@#TX2C )
//   INCLUDE-MOD  LIB = &(PASCAL-RTS) , ELEMENT =                       -
      //   ( IMLDATA  , IMLDCOS  , IMLDEXP  , IMLDLOG  ,                -
      //     IMLDSIN  , IMLDSQR  , IMLEND   , IMLSINI  )
//MODIFY-SYMBOL-VISIBILITY  SYMBOL-NAME = ( IMLSINI , IMLEND )          -
      //                   ,VISIBLE     = NO
//REMARK
//REMARK +---------------- SUB-LLM : ILCS  -------------------------+
//BEGIN-SUB-LLM SUB-LLM-NAME = ILCS
//   INCLUDE-MOD  LIB = &(ITOENTR-LIB)                                  -
      //         ,ELE = ITOENTR
//   INCLUDE-MOD  LIB = &(MODULE-LIB)                                   -
      //         ,ELE = ITOINITS
//REMARK +------------- Hide all ILCS definitions -------------------+
//REMARK
//MODIFY-SYMBOL-VISIBILITY  , VISIBLE = NO
//END-SUB-LLM
//REMARK +---------------   End of SUB-LLM : ILCS  ------------------+
//   MODIFY-SYMBOL-ATTR SYMBOL-NAME =                                   -
   //   (  IMLDATA  , IMLDCOS , IMLDEXP , IMLDLOG , IMLDSIN             -
   //     , IMLDSQR  , IMLEND  , IMLSINI , IL#     , IP@#PA2C           -
   //     , ITOENTR                                             )       -
   //   , PUBLIC = YES
//REMARK
//REMARK +-----------------------  Private slice  --------------------+
//REMARK
//REMARK Own private modules
//   INCLUDE-MOD  LIB = &(MODULE-LIB)                                   -
      //         ,ELE = ( ECHO###D , KCKBL##D )
//REMARK
//REMARK RTS private modules
//   INCLUDE-MOD  LIB = &(PASCAL-RTS) , ELEMENT =                       -
      //   ( IP@$DM2D , IP@$ED2D , IP@$LI2D , IP@$LO2D ,                -
      //     IP@$ME2D , IP@$ML2D , IP@$SF2D , IP@$ST2D ,                -
      //     IP@$SY2D , IP@#CN2D , IP@#ER2D , IP@#HE2D ,                -
      //     IP@#IL2D , IP@#IN2D , IP@#LO2D , IP@#MA2D ,                -
      //     IP@#MM2D , IP@#OP2D , IP@#OU2D , IP@#PA2D ,                -
      //     IP@#PT2D , IP@#RE2D , IP@#RT2D , IP@#ST2D ,                -
      //     IP@#TX2D )
//REMARK
```

```
//REMARK +-------------------------------------------------------------+
//REMARK
//REMARK
//  SAVE-LLM             LIB                 = &SHARED-MODULE-LIB      -
     //                , ELEM                = *INTERNAL               -
     //                , FOR-BS2000-VERSION = FROM-V10
//REMARK
//STEP
//END
/
/  ELSE  "  NOT BLS   ----------------------------------------------- "
/
/ ASSIGN-SYSLST LIST.LINK.PASCAL
/
/  TSOS:   BEGIN-BLOCK DATA-INSERTION = YES
/
/
/         START-PROGRAM                                               -
/            FROM-FILE = $TSOSLNK
   MODULE    PASHARED  ,LET     = YES
   MODULE    PASHARED  ,CMAP    = ALL
   MODULE    PASHARED  ,XREF    = YES
   MODULE    PASHARED  ,LIB     = &SHARED-MODULE-LIB
COMMENT
    LINK-SYMBOLS KEEP=( ECHO###C , IP@#RT2C )
COMMENT
    NCAL
COMMENT
  INCLUDE  ( ECHO###C , KCKBL##C ) ,&(MODULE-LIB)
  INCLUDE  ( IP@$DM2C , IP@$ED2C , IP@$LI2C , IP@$LO2C ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@$ME2C , IP@$ML2C , IP@$SF2C , IP@$ST2C ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@$SY2C , IP@#CN2C , IP@#ER2C , IP@#HE2C ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#IL2C , IP@#IN2C , IP@#LO2C , IP@#MA2C ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#MM2C , IP@#OP2C , IP@#OU2C , IP@#PA2C ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#PT2C , IP@#RE2C , IP@#RT2C , IP@#ST2C ) ,&(PASCAL-RTS)
  INCLUDE    IP@#TX2C                                    ,&(PASCAL-RTS)
  INCLUDE  ( IMLDATA  , IMLDCOS  , IMLDEXP  , IMLDLOG  ) ,&(PASCAL-RTS)
  INCLUDE  ( IMLDSIN  , IMLDSQR  , IMLEND   , IMLSINI  ) ,&(PASCAL-RTS)
  INCLUDE  ITOENTR    , &(ITOENTR-LIB)
  INCLUDE  ITOINITS   , &(MODULE-LIB)
COMMENT
  BIND
/
/
/START-LMS
/  SEND-ST 'O LIB.TP.SHARE.PIN,MODE=UPDATE'
/  SEND-ST 'MOD-ELEM ELEMENT=*LIB(ELEM=PASHARED,TYPE=R)'
/  SEND-ST 'REN ITOENTR  , NEW=PASDUMY1'
/  SEND-ST 'REN ITOINITS , NEW=PASDUMY2'
/  SEND-ST 'REN IMLSINI  , NEW=PASDUMY3'
/  SEND-ST 'REN IMLEND   , NEW=PASDUMY4'
/  SEND-ST 'END-MOD'
/  SEND-ST 'END'
/
/
/         START-PROGRAM                                    -
/            FROM-FILE = $TSOSLNK
   MODULE    PAPRIVAT  ,LET     = YES
   MODULE    PAPRIVAT  ,CMAP    = ALL
   MODULE    PAPRIVAT  ,XREF    = YES
   MODULE    PAPRIVAT  ,LIB     = &(PRIVATE-MODULE-LIB)
COMMENT
    LINK-SYMBOLS KEEP=( ECHOSYN , ECHOASYN )
COMMENT
    NCAL
COMMENT
  INCLUDE  ( ECHO###D , KCKBL##D ) ,&(MODULE-LIB)
  INCLUDE  ( IP@$DM2D , IP@$ED2D , IP@$LI2D , IP@$LO2D ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@$ME2D , IP@$ML2D , IP@$SF2D , IP@$ST2D ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@$SY2D , IP@#CN2D , IP@#ER2D , IP@#HE2D ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#IL2D , IP@#IN2D , IP@#LO2D , IP@#MA2D ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#MM2D , IP@#OP2D , IP@#OU2D , IP@#PA2D ) ,&(PASCAL-RTS)
  INCLUDE  ( IP@#PT2D , IP@#RE2D , IP@#RT2D , IP@#ST2D ) ,&(PASCAL-RTS)
  INCLUDE    IP@#TX2D                                    ,&(PASCAL-RTS)
COMMENT
    BIND
/
/  ASS-SYSLST *PRIMARY
/
/ END-BLOCK TSOS
/
/ END-IF    "   End of linking wit TSOSLNK  ----------------------- "
/
/ END-IF    "   End of linking ------------------------------------ "
/
/ END-BLOCK WORK
/
/IF-BLOCK-ERROR
/END-IF
/
/
/  TCHNG OFLOW = ACK
/
/EXIT-PROC
```

2.7.3 Formatting

Creating formats with IFG

The "IFG" manual explains in detail how to create formats with IFG. When these formats are created for use with *open*UTM, 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
  - for #formats: separate attribute blocks and field contents
  - for *formats: unaligned, no attribute fields
  - for +formats: unaligned, with attribute fields

  Addressing aids for Pascal-XT can only be unaligned.
- Declare *only one* addressing aid.

  Example

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

  ```
  with FORMA;
  from FORMA use T_FORMA;
  package programunit;
     ...
  type
     ...
     kdcnb = record
        ...
        format_a : T_FORMA;
        ...
     end;
  ...
  end.
  ```

  Here FORMA is the format name defined with IFG and T_ is the name prefix for the type declaration. When using this format, specify the format name as "*FORMA" in the field KCMF of the MPUT, FPUT or DPUT call (this gives you addressing aids without attribute fields) or as "+FORMA" (for addressing aids with attribute fields).
- When defining addressing aids please note that, in the case of MGET and FGET calls, UTM removes the transaction code from the message at the start of the service, unless this is prevented by an INPUT exit. If the first field in the format contains the transaction code, you must take this into account in the addressing aids for input formatting. You can either

  - use IFG to define a separate format for input which does not contain the TAC field (for the MGET call you must, of course, enter the name of the entire format in the parameter field), or
  - declare different variants for the structure of the message area.

- When preparing formats for use, you should enter them in the format application file (format library), the name of which must be specified in the FHS start parameters.

- During format preparation, IFG generates for each format a package with the same name as the format. The specification contains the type declaration for the structure of the format. The package body is empty.
  You must compile both parts before use, so that the name of the package is known in your project file. You must enter the generated modules in your module library.
  Once a format has been modified, recompilation is not necessary. (The package contains only one type declaration; consequently only the standard code is generated for initializing a package.)

  Notes

  - For the attribute fields, IFG uses the data type T_FIELD_ATTRIBUTE_SET. This data type is declared in the package FIELD_ATTRIBUTE_PACKAGE, which is supplied with UTM.

  - The other fields in a format are defined as PACKED ARRAY [1..n] OF CHAR (where n = field length). You can replace PACKED ARRAY with ARRAY. The standard procedures PACK and UNPACK can then be used for the transport of data to or from these fields.

  - IFG uses the following naming conventions, which you must not modify:

```
T formatname        for the type declaration of a format
A fieldname         for the attribute fields
DUMMY 000n          for unknown fields accessible to the
                    program.
```

Modifying KDCS attributes

In order to support programming, *open*UTM offers all supported combinations in the package FIELD_ATTRIBUTE_PACKAGE. If X’0000’ is specified in an attribute field, the attributes are taken from the format creation.

Positioning the cursor

UTM offers you two options for positioning the cursor when outputting a format. You specify which option you require using the FHS start parameter CURSOR=ATTR or CURSOR=NOATTR (see the "FHS" manual).

In the case of CURSOR=NOATTR, you must modify the addressing aids for your formats. In the declarations for the fields which you wish to use for cursor positioning, you must replace PACKED ARRAY with ARRAY.
You must observe the rules governing assignment compatibility of character string types in Pascal-XT; use the standard procedures PACK and UNPACK for data transport.

You declare the following procedure:

```
procedure KDCSCUR (VAR c: char); COBOL;
```

The following call enables you to place the cursor at any position in output fields:

```
KDCSCUR (fieldname[i]);
```

where "fieldname" is a field declared in your format; "i" must be within the array boundaries .

Extended line mode

When using terminals in line mode, it is possible to structure the output message with logical control characters.
In line mode, all the control characters of the TIAM access method are permitted.

When working in extended line mode, you must define the control characters yourself. To find out which control characters exist and what values they are to be assigned, see the "TIAM" manual (e.g. in the description of the VTCSET macro). An example is given on page 131.

3 Examples in Pascal-XT

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

3.1 Examples of individual KDCS calls

This section contains coding examples for the following KDCS calls:

– MGET
– MPUT
– DPUT
– APRO with MPUT for distributed processing

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

In the examples given, the names defined in the predefined packages are used for the data structures and constants (see page 5 and chapter “Data structures for Pascal-XT” on page 67ff). The names of the application-specific structures are based on the examples on page 9ff (German abbreviations: *kb* = communication area; *kbk* = communication area header; *spab* = standard primary working area; kcpal = parameter area; *nb* = message area).

MGET call

An active service can be interrupted by an input consisting of a short message created with F2 and containing 10 characters of additional data. This input is to activate a special function.

```
...
...
INLINE PROCEDURE SPECIAL_MGET;            1)
BEGIN
   WITH SPAB.KCPAL DO BEGIN
     KCOP := KCKBL.MGET;
     KCLA := 10;
     KCMF := '        ';
   END;
   WITH SPAB DO KDCS (KCPAL, NB);
END;
...
...
   WITH SPAB.KCPAL DO BEGIN
     KCOP := KCKBL.MGET;
   END;
   WITH SPAB DO KDCS (KCPAL, NB);
   IF KB.KBK.KCRCCC = '21Z'            2)
   THEN SPECIAL_MGET;
   IF KB.KBK.KCRCCC <> '000' THEN MGET_ERROR;
       .
```

1) Another MGET is required for the 10 characters.

2) A special function is queried.

MPUT call

1. A complete 80-byte message is to be sent.

```
    .
    .
    .
WITH SPAB.KCPAL DO BEGIN
  KCOP := KCKBL.MPUT;
  KCOM := 'NE';
  KCLM := 80;
  KCRN := '        ';
  KCMF := '        ';
  KCDF := 0;
END;
WITH SPAB DO KDCS (KCPAL, NB);
IF KB.KBK.KCRCCC <> '000'
THEN MPUT_ERROR;
```

2. The final message in a service (formatted and 500 bytes long in our example) is to be sent to a format terminal. The name of the format is "*PIC15". The screen should be cleared beforehand.

```
    .
    .
    .
WITH SPAB.KCPAL DO BEGIN
  KCOP := KCKBL.MPUT;
  KCOM := 'NE';
  KCLM := 500;
  KCRN := '        ';
  KCMF := '*PIC15  ';
  KCDF := KCDFL.KCREPL;               1)
END;
WITH SPAB DO KDCS (KCPAL, NB);
IF KB.KBK.KCRCCC <> '000'
     THEN MPUT_ERROR;
```

1) REPLACE is performed by default when you change from one format to another.The output is made in order to preclude errors due to undefined field contents.

3. In a *format "PIC10", which according to the last input at the terminal still exists, all variables (i.e. overwritable fields) are to be deleted as the response. The protected fields are to remain intact.

```
  .
  .
  .
WITH SPAB.KCPAL DO BEGIN
  KCOP := KCKBL.MPUT;
  KCOM := 'NE';
  KCLM := ZEROES;
  KCRN := SPACES;
  KCMF := '*PIC10  ';
  KCDF := KCDFL.KCERAS;
END;
WITH SPAB DO KDCS (KCPAL, NB);
IF KB.KBK.KCRCCC <> '000'
     THEN MPUT_ERROR;
```

DPUT call

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

```
     .
     .
     .
with spab.kcpal do begin
    kcop := 'DPUT';
    kcom := 'NE';
    kclm := 11;
    kcmf := '        ';
    kcdf := 0;
    kcrn := 'DEEDAY  ';
    kcmod := 'A';
    kctag :='157'; kcstd := '12'; kcmin := '00'; kcsek := '00';
end;
with spab do kdcs (kcpal, nb);
if kb.kbk.kcrccc <> '000' then dput_error;
```

2. An asynchronous message of 80 characters is to be sent after 1 hour (relative time entry) to the data display terminal "DDT1". The screen function "audible alarm" (BEL) is also to be triggered.

```
     .
     .
     .
with spab.kcpal do begin
    KCOP := kckbl.DPUT;
    KCOM := 'NE';
    KCLM := 80;
    KCRN := 'DDT1    ';
    KCMF := '        ';
    KCDF := kcdfl.KCALARM;
    KCMOD := 'R';
    KCTAG := KCKBL.PIC_999 ('0':3);
    KCSTD := KCKBL.PIC_99 ('0','1');
    KCMIN := KCKBL.PIC_99 ('0':2);
    KCSEK := KCKBL.PIC_99 ('0','0');
 end;
 with spab do kdcs (kcpal, nb);
 if kb.kbk.kcrccc <> '000' then dput_error;
```

APRO call with MPUT in distributed processing

The dialog service with the transaction code "LTAC1" of the job-receiving application "PARTNER1" is to be addressed from the job-submitting service (double-step addressing). At the same time, the job-receiving service is to be assigned the service ID ">SEID1". An MPUT message, length 100, is then sent in line mode to the partner application.

```
        .
        .
with spab.kcpal do begin
   kcop := kckbl.apro; kcom := 'DM'; kclm := 0;
   kcrn := 'LTAC1   ';
   kcpa := 'PARTNER1';
   kcpi := '>SEID1  ';
end;
kdcs (kcpal ...);
if kb.kbk.kcrccc <> '000'
then apro_error;
        .
        .
        .
with spab.kcpal do begin
  kcop := kckbl.mput;
  kcom := 'NE';
  kclm := 100;
  kcrn := '>SEID1  ';
  kcmf := '        ';
  kcdf := 0;
end;
with spab do kdcs (kcpal, nb);
if kb.kbk.kcrccc <> '000'
then mput_error;
```

3.2 Example of an asynchronous MSGTAC program unit

The MSGTAC program unit DATPRO is intended to prevent unauthorized users from signing on to a UTM application. If more than 3 KDCSIGN attempts are made at an LTERM partner with an invalid user ID, password or ID card, the connection to this terminal is to be cleared down.

For the preparatory actions, see the Core Manual "Programming Applications with KDCS for COBOL, C and C++" .
In this example, UTM is installed under the user ID $UTM.

Implementing the MSGTAC program unit

The MSGTAC program unit DATPRO counts the number of failed attempts to sign on in a TLS. If UTM accepts a KDCSIGN (i.e. message K008 or K033 is output), this TLS is deleted again.
If, following 3 invalid KDCSIGN attempts, the fourth KDCSIGN attempt is also invalid, the corresponding terminal is to be disconnected via "asynchronous administration". This takes place with the contents "PTERM=*pterm*,PRO=*proname*, ACT=DIS" (see also the *open*UTM-Manual "Generating and Handling Applications").
The administration command is then logged in the user log file with LPUT and the TLS is deleted.

Each K message is read by the MSGTAC program unit using FGET. When one K message has been "processed", the next K message is read immediately with FGET within the same program unit run.

Package specification

```
with kckbl;
from kckbl use
   kckb, kcpal, redefines;
with kcmsl;
from kcmsl use
   kcmsgl;
package DATPRO_ADAPTER;

type
   pchar8 = packed array [1..8] of char;
   char80 = array [1..80] of char;
   kcdata = packed array [1..sizeof (string) - 2] of char;

   kckbc = record
      kbk: kckb;
      prb: char; (* program area; not used *)
   end;

   kcnb = record
      case redefines of
         v1 : (str    (0): string);
         v2 : (ccc    (2): kcdata);
         v3 : (txt    (2): char80);
         else: (hack_no (2): integer);
   end;

   kcspab = record
      pf           : kcpal;
      hacker_lterm : pchar8;
      nb           : kcnb;
      msg          : kcmsgl;
   end;


   entry procedure DATPRO (var kb: kckbc; var spab: kcspab);

end.
```

Package body

```
with kckbl;
from kckbl use
   kckb, kcpal;
with kcmsl;
from kcmsl use
   kcmsgl;
package body DATPRO_ADAPTER;

const
   no_format = pchar8(' ':8);
   go_pend_rset = 1;
   hack_max = 3;
   id_hack_tls = pchar8('T','L','S','H','A','C','K',' ');

procedure DATPRO (var kb: kckbc; var spab: kcspab);

var
   errorline : string;
   pt,
   pp : pchar8;

   procedure work;
      procedure kdcs (var pf: kcpal; var data: kcdata); cobol;
   begin
      with spab, spab.nb, spab.pf, spab.msg, kb.kbk
      do begin
         kcop := 'GTDA';
         kcla := sizeof (hack_no);
         kcrn := hacker_lterm;
         kdcs (pf, ccc);
         if kcrccc <> '000'
         then return;
         if kcrlm = 0
         then begin
            if not ((msgno = 'K008') or (msgno = 'K033'))
            then begin (* no other TLS *)
               kcop := 'PTDA';
               kcla := sizeof (hack_no);
               hack_no := 1;
               kcrn := id_hack_tls;
               kclt := hacker_lterm;
               kdcs (pf, ccc);
            end;
         end
         else begin
            if ((msgno = 'K008') or (msgno = 'K033'))
```

```
then begin (* ok; delete TLS *)
   kcop := 'PTDA';
   kcla := 0;
   kcrn := id_hack_tls;
   kdcs (pf, ccc);
end
else begin (* check_no*)
   hack_no := hack_no + 1;
   if hack_no <= hack_max
   then begin (* may continue trying *)
      kcop := 'PTDA';
      kcla := sizeof (hack_no);
      kcrn := id_hack_tls;
      kclt := hacker_lterm;
      kdcs (pf, ccc);
      return;
   end
   else begin (* DISCONNECT !! *)
      if msgno = 'K004'
      then with k004
      do begin
         pt := ptrm; pp := prnm;
      end
      else
      if msgno = 'K006'
      then with k006
      do begin
         pt := ptrm; pp := prnm;
      end
      else
      with k031
      do begin
         pt := ptrm; pp := prnm;
      end;
      writestring (str, 'PTERM=(', pt,
                        '),PRONAM=', pp,
                        ',ACTION=DIS');
   (*P_FPUT*)
      kcop := 'FPUT';
      kcom := 'NE';
      kcrn := 'KDCPTRMA';
      kclm := length (str);
      kcmf := no_format;
      kcdf := 0;
      kdcs (pf, ccc);
      if kcrccc <> '000'
      then return;
   (*P_LPUT*)  (* log on USER-LOGGING *)
```

```
                    kcop := 'LPUT';
                    kcla := length (str);
                    kdcs (pf, ccc);
                    if kcrccc <> '000'
                    then return;
                  (*P_PTDA*)  (* delete TLS *)
                    kcop := 'PTDA';
                    kcla := 0;
                    kcrn := id_hack_tls;
                    kclt := hacker_lterm;
                    kdcs (pf, ccc);
                  end (*disconnect*)
                end (* check_no*);
              end;
            end (*with*);
          end (*work*);
          procedure init;
            procedure kdcs (var pf: kcpal); cobol;
          begin
            with spab.pf
            do begin
              kcop := 'INIT';
              kclkbprg := 0;
              kclpab := sizeof (kcspab);
            end;
            kdcs (spab.pf);
            if kb.kbk.kcrccc <> '000'
            then raise (go_pend_rset);
          end;

          function  fget: boolean;
            procedure kdcs (var pf: kcpal; var msg: kcmsgl); cobol;
          begin
            with spab.pf
            do begin
              kcop := 'FGET';
              kcla := sizeof (kcmsgl);
              kcmf := no_format;
            end;
            kdcs (spab.pf, spab.msg);
            with kb.kbk
            do if kcrccc <> '000'
              then if kcrccc = '10Z'
                   then begin
                     fget := false;
                     return
                   end
                   else raise (go_pend_rset);
            with spab, spab.msg, spab.nb
            do begin
              if msgno = 'K004' (* invalid user ID *)
              then hacker_lterm := k004.ltrm
              else
              if msgno = 'K006' (* invalid password *)
              then hacker_lterm := k006.ltrm
              else
              if msgno = 'K008' (* kdcsign accepted *)
              then hacker_lterm := k008.ltrm
              else
              if msgno = 'K031' (* wrong card *)
              then hacker_lterm := k031.ltrm
              else
              if msgno = 'K033' (* start format *)
              then hacker_lterm := k033.ltrm
              else begin
                spab.pf.kcop := msgno;
                raise (go_pend_rset);
              end;
            end;
            work;
            if kb.kbk.kcrccc <> '000'
            then raise (go_pend_rset);
            fget := true; (* message present *)
          end (*fget*);
          procedure kdcs (var pf: kcpal); cobol;

          procedure emsg (message: string);
            procedure kdcs (var pf: kcpal; var msg: char80); cobol;

          begin
            with spab.pf, spab.nb
            do begin
              kcop := 'LPUT';
              kcla := sizeof (char80);
              unpack (message, txt, 1);
            end;
            kdcs (spab.pf, spab.nb.txt);
          end (*emsg*);

       begin (*DATPRO*)
          init;
          while fget do ; (* as long as messages are present *)
          (*PEND_ANF*)
          with spab.pf
          do begin
            kcop := 'PEND';
      kcom := 'FI';
   end;
   kdcs (spab.pf);

exception (* handling of errors *)
   with spab, spab.pf, kb.kbk
   do begin
      case error_number of
         go_pend_rset:
            begin
               (*PEND_RSET*)
               writestring (errorline,'ERROR IN PROGUNIT DATPRO; ',
                                         'CONV./TAC   ', kctacvg,
                                         '/', kctacal, ' WG. ', kcop,
                                         '  (RC: ', kcrccc, kcrckz, kcrcdc,
                                         ')');
               kcop := 'RSET';
               kdcs (spab.pf);
               (*PEND_RSET_LPUT*)
               emsg (errorline);
               (*PEND_RSET_ANF*)
               kcop := 'PEND';
               kcom := 'FI';
               kdcs (spab.pf);
            end;
         else : raise (error_number); (* runtime error etc. *)
      end;
   end;
end (*DATPRO*);

begin (*evaluation*)
end.
```

3.3 Example of a complete UTM application

The following example of a complete UTM application deals with address management.

This sample application can be used to manage address data located in an ISAM file.
For this purpose, the application supplies the following functions, which can be called by entering the appropriate TAC in the field provided.
The same format is used for input and output.

TAC	**Function**	
1	Display	displays an address from the file. The search criterion (ISAM key) is the last name and 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. There must not already be an address with the same search criterion (see above).
3	Modify	modifies an address entry. The address must already exist in the file.
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 DISPLAY, transaction codes 2, 3 and 4 the program unit MODIFY. Each of these program units then branches to the program unit FIMAPAPA. This program unit serves as a START and SHUT event exit and contains the subroutines that perform input/output to the address file.
The program unit BADTACS is called automatically by UTM whenever an invalid TAC is entered. Following the connection setup to the application and a successful KDCSIGN, UTM immediately outputs the format (start format). Interaction with the user then proceeds in strict dialog, i.e. when a TAC and the ISAM 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 to show how to program with UTM. The ISAM file access operations are not backed up by the UTM transaction concept. For a "genuine" application, it is advisable to use a database system or LEASY. For the sake of simplicity, DB-specific program units have not been included in this example.

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

```
DISPLAY program unit
    INIT call
    MGET call
    Subroutine call FIMAPAPA: read file address
    MPUT call
    PEND call
```

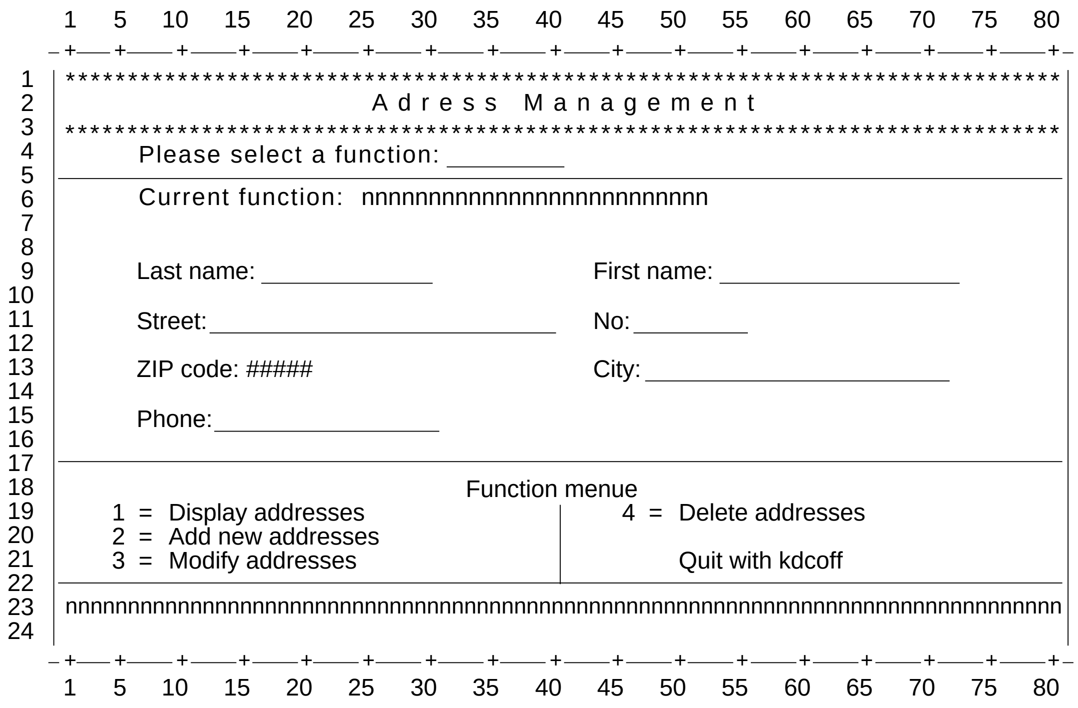
Structure diagram of the DISPLAY program unit

```
MODIFY program unit
    INIT call
    MGET call
    Subroutine call FIMAPAPA: write, overwrite or
    delete address, according to TAC
    MPUT call
    PEND call
```

Structure diagram of the MODIFY program unit

For the sake of completeness, the Pascal-XT program is immediately followed by the generation of this application. The precise meaning of the individual operands and statements can be found in the *open*UTM-Manual "Generating and Handling Applications".

The figure below shows the format used for this application:



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

The IFG attribute list for the *format "FORMA" and the package specification and package body generated for the format are reproduced on the following pages.

The IFG attribute list for the *format "FORMA"

```
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
                           UPPERCASE ONLY
                           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               019  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                           TALICS
                           ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
                           (*)
 06 026 FUNCTION      026  OUTPUT FIELD, PROTECTED, NORMAL, ACCESSIBLE TO PROGRAM
                           ITALICS
                           AUTO INPUT
                           ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
                           (*)
 09 007               010  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                           ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
 09 017 LASTNAME      014  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                           UPPERCASE ONLY
                           ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
 09 043               011  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                           ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
 09 054 FST           002  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                           UPPERCASE ONLY
                           ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                           ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
                           START OF GROUP   LASTNAME:  FIRSTNAME
 09 056               001  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                           ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
```

```
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
09 057 FSTREST       018  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                          UPPERCASE ONLY
                          ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
                          END OF GROUP
11 007               007  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
11 014 STREET        025  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                          UPPERCASE ONLY
                          ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
11 039               007  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '

11 046 HOUSENO       010  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                          UPPERCASE ONLY
                          ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
13 007               009  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
13 016 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               005  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
13 048 CITY          027  INPUT FIELD, UNPROTECTED, BRIGHT, ACCESSIBLE TO PROGRAM
                          UPPERCASE ONLY
                          ALIGNMENT / FILL CHARACTER FOR INPUT: LEFT     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : LEFT     / ' '
15 007               006  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
15 013 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     / ' '
19 005               058  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
                          ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                          ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
20 005               036  TEXT FIELD, PROTECTED, NORMAL, NOT ACCESSIBLE TO PROGRAM
```

```
                                ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                                ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
21 005                     058  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
                                UPPERCASE ONLY
                                ALIGNMENT / FILL CHARACTER FOR INPUT: NONE     / ' '
                                ALIGNMENT / FILL CHARACTER OUTPUT   : NONE     / ' '
                                (*)
```

Package specification and package body

The package specification and body generated by IFG for the format are shown below.

```
PACKAGE FORMA   ;

(* USER AREA LENGTH: 0233 *)
TYPE T_FORMA    =
  RECORD
    TAC                                (0000) : PACKED ARRAY
                                                (.01..08.) OF CHAR;
    FUNCTION                           (0008) : PACKED ARRAY
                                                (.01..26.) OF CHAR;
    LASTNAME                          (0034) : PACKED ARRAY
                                                (.01..14.) OF CHAR;
    FIRSTNAME                         (0048) :
      RECORD
        FST                                (0000) : PACKED ARRAY
                                                    (.01..02.) OF CHAR;
        DUMMY_0001                         (0002) : PACKED ARRAY
                                                    (.01..18.) OF CHAR;
      END;
    STREET                             (0068) : PACKED ARRAY
                                                (.01..25.) OF CHAR;
    HOUSENO                            (0093) : PACKED ARRAY
                                                (.01..10.) OF CHAR;
    ZIP                                (0103) : PACKED ARRAY
                                                (.01..05.) OF CHAR;
    CITY                               (0108) : PACKED ARRAY
                                                (.01..27.) OF CHAR;
    PHONE                              (0135) : PACKED ARRAY
                                                (.01..18.) OF CHAR;
    MESSAGETEXT                        (0153) : PACKED ARRAY
                                                (.01..80.) OF CHAR;
  END;

END.



PACKAGE BODY FORMA   ;
BEGIN
END.
```

Note

The package specification and body for the format(s) used must be compiled using the Pascal-XT compiler and the modules generated (code and data modules) must be copied into the object module library of the UTM application before they can be used for compiling the program units.

Package specification for the address management example

```
with kckbl;
from kckbl use kckb, kcpal, redefines;
with forma;
from forma use t_forma;
package EXAMPLE_1 (addrfile);

type

   (* structure of a record in the address file *)

   id_addressrecord = record
      lastname (000) : packed array [1..14] of char;
      fst      (014) : packed array [1..02] of char;
      fstrest  (016) : packed array [1..18] of char;
      street   (034) : packed array [1..25] of char;
      houseno  (059) : packed array [1..10] of char;
      zip      (069) : packed array [1..05] of char;
      city     (074) : packed array [1..27] of char;
      phone    (101) : packed array [1..18] of char;
   end;


   (* structure of the message area *)

   id_data = packed array [1..225] of char;

   id_nb = record
      case redefines of
         v1 : (tac        (000) : packed array [1..8] of char);
         v2 : (data       (008) : id_data);
         v3 : (form       (000) : t_forma);
         v4 : (address    (034) : id_addressrecord);
         v5 : (message    (153) : array [1..80] of char);
         v6 : (chars      (000) : array [1..233] of char);
         else: ();
   end;


   (* structure of the communication area *)

   id_kdckb = record
      kbheader : kckb;                   (* KB header *)
      kbprg    : id_nb;                  (* program area *)
   end;
   (* structure of the standard primary working area *)
   id_spab = record
      kcpal : kcpal;                     (* parameter field *)
      nb    : id_nb;                     (* message area *)
   end;
   (* definition of the (global) file "addrfile" *)

   type
      id_addrfile  = file of id_addressrecord;

   var
      addrfile : id_addrfile;

      (* Note
            Specifications for file organization (ISAM, position and
            length of the key, record format) are not component parts
            of Pascal-XT; they must thus be defined by means of a
            /FILE command and/or by means of the standard procedure
            assignfile *)



   (* program unit DISPLAY *)

   entry procedure DISPLAY (var kb: id_kdckb; var spab: id_spab);


   (* program unit MODIFY *)

   entry procedure MODIFY (var kb: id_kdckb; var spab: id_spab);


   (* program unit FIMAPAPA *)

   entry procedure FIMAPAPA (var kb: id_kdckb; var spab: id_spab);


   (* program unit BADTACS *)

   entry procedure BADTACS (var kb: id_kdckb; var spab: id_spab);

end (*EXAMPLE_1*).
```

Package body for the address management example

In the following example, the compilation listing (only slightly modified) is reproduced in order to demonstrate the package concepts in Pascal-XT.

```
*** SOURCE LISTING ***  BS2000   PASCAL-XT COMPILER V2.2B         DATE: ...


GLOBAL OPTIONS FOR THIS COMPILATION

CHECK      =    OFF          BY COMMAND
DEBUG      =    OFF          BY COMMAND
GENERATE   =    ON           BY COMMAND
INITIALIZE =    OFF          BY COMMAND
LIST       =    ON           BY COMMAND
MAP        =    ON           BY COMMAND
OPTIMIZE   =    ON           BY COMMAND
STANDARD   =    OFF          BY COMMAND
XREF       =    OFF          BY COMMAND


LIST OF RECOMPILED PACKAGE SPECIFICATIONS (SOURCE FILES)

               ($xy.TUTM.SRCLIB, KCKBLS(*STD,S))

               ($xy.TUTM.SRCLIB, FORMA-SPEC(*STD,S))

               ($PASCAL-XT-SPECS, DMSIO(*STD,S))

               ($PASCAL-XT-SPECS, ERRORS(*STD,S))

               ($PASCAL-XT-SPECS, BS2000CALLS(*STD,S))

               ($xy.TUTM.SRCLIB, EXAMPLE-S(*STD,S))


CURRENT COMPILATION UNIT (SOURCE FILE)

               ($xy.TUTM.SRCLIB, EXAMPLE-B(*STD,S))

     1      with kckbl;
     2      from kckbl use kckb, kcpal, redefines, pic_xx, pic_xxx, pic_x_4;
     3      with forma;
     4      from forma use t_forma;
     5      with dmsio;
     6      from dmsio use getkey, nokey, elim, replace, close;
     7      with errors;
     8      from errors use system_code;
     9      with bs2000calls;
```

```
from bs2000calls use cmd;
package body EXAMPLE_1 (addrfile);

type
   range = 1..sizeof (id_nb);
   id_spaces = array [range] of char;
var
   space_area: id_spaces;
const
   space_constant = id_spaces(' ':sizeof (id_spaces));

   no_addr = '***      NO ADDRESS WITH THIS NAME EXISTS    ***';
   dup_addr = '***  ADDRESS WITH THIS NAME ALREADY EXISTS   ***';


inline procedure spaces (var c: packed array[lo..hi:range] of char);
begin
   pack (space_area, 1, c);
end (* spaces *);


inline function hexa (h: integer): string;
type
   t_hexdig = array [0..15] of char;
const
   c_hexdig = t_hexdig ('0','1','2','3','4','5','6','7',
                        '8','9','A','B','C','D','E','F');
var
   i: 0..7;
   s: string[8];
   n: integer;
begin
   n := h + #10000;
   s := '';
   for i := 0 to 7
   do begin
      n := (n * #10) mod #100000;
      insert (c_hexdig[(n div #10000)], s, length (s) + 1);
   end;
   hexa := s;
end (* hexa *);


inline function dvserr (errcode: integer): string;
begin
   dvserr := concat ('*** FILE ERROR #', hexa (system_code), ' ***');
end (* dvserr *);



```

```

function READ (var address: id_addressrecord): string;
begin
   (* enter value for ISAM key *)
   addrfile^.lastname := address.lastname;
   addrfile^.fst := address.fst;
   (* delete other fields *)
   with address
   do begin
      spaces (street);
      spaces (no);
      spaces (zip);
      spaces (city);
      spaces (phone);
   end;
   getkey (addrfile);
   if nokey (addrfile)
   then read := no_addr
   else begin
      address := addrfile^;
      read := '';
   end;
exception
   if error_number = file_error
   then read := dvserr (system_code)
   else raise (error_number);
end (* READ *);


function WRITE (var address: id_addressrecord): string;
begin
   (* check whether entry already exists *)
   addrfile^.lastname := address.lastname;
   addrfile^.fst := address.fst;
   getkey (addrfile);
   if nokey (addrfile)
   then begin (* record does not yet exist *)
      addrfile^ := address;
      put (addrfile);
      write := '';
   end
   else write := dup_addr;
exception
   if error_number = file_error
   then write := dvserr (system_code)
   else raise (error_number);
end (* WRITE *);


function OVERWRITE (var address: id_addressrecord): string;
```

```
begin
   addrfile^.lastname := address.lastname;
   addrfile^.fst:= address.fst;
   getkey (addrfile);
   if nokey (addrfile)
   then overwrite := no_addr
   else begin
      addrfile^ := address;
      put (addrfile);
      overwrite := '';
   end;
exception
   if error_number = file_error
   then overwrite := dvserr (system_code)
   else raise (error_number);
end (* OVERWRITE *);


function DELETE (var address: id_addressrecord): string;
begin
   addrfile^.lastname := address.lastname;
   addrfile^.fst := address.fst;
   getkey (addrfile);
   if nokey (addrfile)
   then delete := no_addr
   else begin
      elim (addrfile);
      delete := '';
   end;
exception
   if error_number = file_error
   then delete := dvserr (system_code)
   else raise (error_number);
end (* DELETE *);


(* program unit FIMAPAPA *)

procedure FIMAPAPA (var kb: id_kdckb; var spab: id_spab);
var
   filecmd: string;
   not_ok: boolean;
begin
   if kb.kbheader.kctacvg = 'STARTUP '
   then begin
      writestring (filecmd, '/FILE ADDRESSES,LINK=ADDRFILE,'
                           ,'FCBTYPE=ISAM,RECFORM=V,BLKSIZE=STD,'
                           ,'KEYPOS=5,KEYLEN=16,DUPEKY=NO');
      cmd (filecmd, not_ok);
      if not not_ok
      then replace (addrfile)
      else raise (999); (* abort application if command error *)
   end;
   if kb.kbheader.kctacvg = 'SHUTDOWN'
   then close (addrfile);
end (* FIMAPAPA *);

(* common procedures for the following program units *)

procedure INIT (var pf: kcpal; var nb: id_nb);
   procedure kdcs (var pf: kcpal); COBOL;
begin
   with pf
   do begin
      kcop := 'INIT';
      kclkbprg := 0;
      kclpab := sizeof (id_spab);
   end;
   with nb
   do begin
      spaces (tac);
      spaces (data);
   end;
   kdcs (pf);
end (*INIT*);


procedure MGET (var pf: kcpal; var nb: id_nb);
   procedure kdcs (var pf: kcpal; var data: id_data); COBOL;
begin
   with pf
   do begin
      kcop := 'MGET';
      kcla := sizeof (nb);
      kcmf := '*FORMA  ';
   end;
   kdcs (pf, nb.data);
end (*MGET*);

procedure MPUT (var pf: kcpal; var nb: id_nb; len: integer);
   procedure kdcs (var pf: kcpal; var nb: id_nb); COBOL;
begin
   with pf
   do begin
      kcop := 'MPUT';
      kcom := 'NE';
      kclm := len;
      if len = sizeof (t_forma)
      then kcmf := '*FORMA  '
      else spaces (kcmf);
      spaces (kcrn);
   end;
   kdcs (pf, nb);
end (*MPUT*);

procedure PEND (var pf: kcpal; opmod: pic_xx);
   procedure kdcs (var pf: kcpal); COBOL;
begin
   with pf
   do begin
      kcop := 'PEND';
      kcom := opmod;
   end;
   kdcs (pf);
end (*PEND*);

procedure ERROR (tp: string; f_op: pic_x_4; f_rc: pic_xxx; spab: id_spab);
var
   temp: string;
begin
   writestring (temp, '*** E R R O R ***  PROGRAM UNIT: ', tp:8,
                      ' OPERATION CODE: ', f_op,
                      ' RETURNCODE: ', f_rc);
   unpack (temp, spab.nb.chars, 1);
   spab.kcpal.kcdf := 0; (* no screen functions *)
   mput (spab.kcpal, spab.nb, length (temp));
   pend (spab.kcpal, 'ER');
end (*ERROR*);



(* program unit DISPLAY *)

procedure DISPLAY (var kb: id_kdckb; var spab: id_spab);
var
   dvrc : string; (* DVS error message *)
begin
   with spab, kb.kbheader
   do begin
      init (kcpal, nb);
      if kcrccc <> '000'
      then error ('DISPLAY', kcpal.kcop, kcrccc, spab);
      mget (kcpal, nb);
      if kcrccc <> '000'
      then error ('DISPLAY', kcpal.kcop, kcrccc, spab);
      dvrc := read (nb.address);
      if dvrc <> ''
      then unpack (dvrc, nb.message, 1);
      mput (kcpal, nb, sizeof (id_nb));
      if kcrccc <> '000'
      then error ('DISPLAY', kcpal.kcop, kcrccc, spab);
      pend (kcpal, 'FI');
   end;
end (*DISPLAY*);



(* program unit MODIFY *)

procedure MODIFY (var kb: id_kdckb; var spab: id_spab);
label
   9; (* error exit *)
var
   dvrc : string; (* DVS error message *)
   bintac: integer;
begin
   with spab, kb.kbheader
   do begin
      init (kcpal, nb);
      if kcrccc <> '000' then goto 9;
      mget (kcpal, nb);
      if kcrccc <> '000' then goto 9;
      begin
         readstring (kctacvg, bintac);
      exception
         bintac := -1; (* error if not only digits and blanks *)
      end;
      case bintac of
         2 : dvrc := write (nb.address);
         3 : dvrc := overwrite (nb.address);
         4 : dvrc := delete (nb.address);
         else: dvrc := concat ('*** INVALID TAC (', kctacvg, ') ***');
      end;
      if dvrc <> ''
      then unpack (dvrc, nb.message, 1);
      mput (kcpal, nb, sizeof (id nb));
      if kcrccc <> '000' then goto 9;
      pend (kcpal, 'FI');

   9: (* error ; no return from ERROR *)
      error ('MODIFY', kcpal.kcop, kcrccc, spab);
   end;
end (*MODIFY*);



(* program unit BADTACS *)

procedure BADTACS (var kb: id_kdckb; var spab: id_spab);
```

```
label
   9; (* KDCS error *)
var
   temp: string;
begin
   with spab, kb.kbheader,nb
   do begin
      init (kcpal, nb);
      if kcrccc <> '000' then goto 9;
      mget (kcpal, nb);
      if kcrccc = '05Z'
      then spaces (nb.data)
      else
      if kcrccc <> '000' then goto 9;
      spaces (form.messagetext);
      unpack (concat ('******INVALID TAC - PLEASE REPEAT
                                    INPUT.******'),
              message, 1);
      spaces (form.tac);
      mput (kcpal, nb, sizeof (id nb));
      if kcrccc <> '000' then goto 9;
      pend (kcpal, 'FI');

   9: (* KDCS error *)
      error ('BADTACS', kcpal.kcop, kcrccc, spab);

   end;
end (*BADTACS*);


begin (* package body is empty *)
end.
```

```
*** MAP LISTING ***     BS2000 SIEMENS PASCAL-XT COMPILER             DATE: ...



                    PROCEDURE ENTRY VECTOR

       PEV-ADDRESS      MODULE-OFFSET        PROCEDURE / FUNCTION
      24 (00000018)      96 (00000060)   INITIAL PROCEDURE
      28 (0000001C)       0 (00000000)   DISPLAY
      32 (00000020)       0 (00000000)   MODIFY
      36 (00000024)       0 (00000000)   FIMAPAPA
      40 (00000028)       0 (00000000)   BADTACS
      44 (0000002C)       0 (00000000)   READ
      48 (00000030)       0 (00000000)   WRITE
      52 (00000034)       0 (00000000)   OVERWRITE
      56 (00000038)       0 (00000000)   DELETE
      60 (0000003C)      -1 (FFFFFFFF)   kdcs
      64 (00000040)       0 (00000000)   INIT
      68 (00000044)      -1 (FFFFFFFF)   kdcs
      72 (00000048)       0 (00000000)   MGET
      76 (0000004C)      -1 (FFFFFFFF)   kdcs
      80 (00000050)       0 (00000000)   MPUT
      84 (00000054)      -1 (FFFFFFFF)   kdcs
      88 (00000058)       0 (00000000)   PEND
      92 (0000005C)       0 (00000000)   ERROR


                    GLOBAL CONSTANTS OF THE UNIT

     MODULE-OFFSET     TYPE       NAME          VALUE
      96 (00000060)   STRUCT   space_constant
     324 (00000144)   STRING   no_addr         '***NO ADDRESS WITH THIS NAME EXISTS***'
     372 (00000174)   STRING   dup_addr        '***ADDRESS WITH THIS NAME ALREADY
EXISTS***'
     420 (000001A4)   STRUCT   c_hexdig
      -1 (FFFFFFFF)   STRING                   ''
     436 (000001B4)   STRING                   '*** FILE ERROR #'
     453 (000001C5)   STRING                   ' ***'
      -1 (FFFFFFFF)   STRING                   ''
      -1 (FFFFFFFF)   STRING                   ''
      -1 (FFFFFFFF)   STRING                   ''
      -1 (FFFFFFFF)   STRING                   ''
     457 (000001C9)   STRING                   'STARTUP '
     465 (000001D1)   STRING                   '/FILE ADDRESSES,LINK=ADDRFILE,'
     494 (000001EE)   STRING                   'FCBTYPE=ISAM,RECFORM=V,BLKSIZE=STD,
     529 (00000211)   STRING                   'KEYPOS=5,KEYLEN=16,DUPEKY=NO'
     557 (0000022D)   STRING                   'SHUTDOWN'
     565 (00000235)   STRING                   'INIT'
     569 (00000239)   STRING                   'MGET'
     573 (0000023D)   STRING                   '*FORMA  '
     581 (00000245)   STRING                   'MPUT'
     585 (00000249)   STRING                   'NE'
     587 (0000024B)   STRING                  '*FORMA  '
     595 (00000253)   STRING                  'PEND'
     599 (00000257)   STRING                  '***E R R O R*** PROGRAM UNIT: '
     634 (0000027A)   STRING                  ' OPERATION CODE: '
     651 (0000028B)   STRING                  ' RETURNCODE: '
     664 (00000298)   STRING                  'ER'
     666 (0000029A)   STRING                  '000'
     669 (0000029D)   STRING                  'DISPLAY'
     676 (000002A4)   STRING                  '000'
     679 (000002A7)   STRING                  'DISPLAY'
      -1 (FFFFFFFF)   STRING                  ''
     686 (000002AE)   STRING                  '000'
     689 (000002B1)   STRING                  'DISPLAY'
     696 (000002B8)   STRING                  'FI'
     698 (000002BA)   STRING                  '000'
     701 (000002BD)   STRING                  '000'
     704 (000002C0)   STRING                  '*** INVALID TAC ('
     724 (000002D4)   STRING                  ') ***'
      -1 (FFFFFFFF)   STRING                  ''
     729 (000002D9)   STRING                  '000'
     732 (000002DC)   STRING                  'FI'
     734 (000002DE)   STRING                  'MODIFY'
     741 (000002E5)   STRING                  '000'
     744 (000002E8)   STRING                  '05Z'
     747 (000002EB)   STRING                  '000'
     750 (000002EE)   STRING                  '******INVALID TAC - PLEASE
                                                     REPEAT INPUT.******'
     804 (00000324)   STRING                  '******INVALID TAC - PLEASE
                                                     REPEAT INPUT.******'
     859 (0000035B)   STRING                  '000'
     862 (0000035E)   STRING                  'FI'
     864 (00000360)   STRING                  'BADTACS'

                    GLOBAL VARIABLES OF THE UNIT

      32 (00000020)   addrfile
     506 (000001FA)   space_area


***********************************************************
*                   COMPILATION SUMMARY                   *
***********************************************************
*  ERRORS DETECTED        :        0                      *
*  WARNINGS               :        0                      *
*  SIZE OF CODE MODULE    :     6640 BYTES                *
*  SIZE OF DATA MODULE    :     1076 BYTES                *
*  COMPILATION TIME       :    9.904 SEC                  *
***********************************************************
```

KDCDEF statements

```
REM *************************************************************
REM ***             D E F     S T A T E M E N T S             ***
REM ***                                                       ***
REM ***                   KDCFILE = APPLI                     ***
REM *************************************************************
MAX APPLINAME=A
MAX KDCFILE=(KDCFILE.APPLI,S),TASKS=2,ASYNTASKS=1
MAX CONRTIME=5,LOGACKWAIT=60
ROOT ADRROOT
OPTION GEN=ALL
REM *************************************************************
REM ************          PROGRAM STATEMENTS         ************
REM *************************************************************
PROGRAM KDCADM,COMP=ILCS
PROGRAM DISPLAY,COMP=ILCS
PROGRAM MODIFY,COMP=ILCS
PROGRAM FIMAPAPA,COMP=ILCS
PROGRAM BADTACS,COMP=ILCS
REM *************************************************************
REM ************           EXIT STATEMENTS           ************
REM *************************************************************
EXIT PROGRAM=FIMAPAPA,USAGE=START
EXIT PROGRAM=FIMAPAPA,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 KDCSENDA
TAC KDCAPPLA
TAC KDCDIAGA
TAC KDCLOGA
TAC KDCINFA
TAC KDCHELPA
TAC KDCSHUTA
TAC KDCTCLA
DEFAULT TAC TYPE=D,PROGRAM=(STD)
TAC KDCBADTC,PROGRAM=BADTACS
TAC 1,LOCK=1,PROGRAM=DISPLAY
TAC 2,LOCK=2,PROGRAM=MODIFY
TAC 3,LOCK=2,PROGRAM=MODIFY
TAC 4,LOCK=2,PROGRAM=MODIFY
REM ************************************************************
REM ************          USER STATEMENTS           ************
REM ************************************************************
USER  SUSIE,PASS=C'UTM4EVER',KSET=BUNCH1,PERMIT=ADMIN,FORMAT=*FORMA
USER  GERTRUDE,PASS=C'UTMNEVER',KSET=BUNCH2,STATUS=ON,FORMAT=*FORMA
USER  BARBARA,KSET=BUNCH3,STATUS=ON,FORMAT=*FORMA
REM ************************************************************
REM ************       PTERM/LTERM STATEMENTS       ************
REM ************************************************************
DEFAULT PTERM PRONAM=DSR01,PTYPE=T9750
PTERM DDT01,LTERM=UTMDT1
PTERM DDT02,LTERM=UTMDT2
PTERM DDT03,LTERM=UTMDT3
DEFAULT PTERM PRONAM=DSR01,PTYPE=T9022,USAGE=O
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=O
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
```

4 Data structures for Pascal-XT

For each Pascal-XT data structure there are 2 files per package; these differ in the last letter of the file name:

S identifies the package specification, which contains the description of the data structure.

B identifies the package body, whose contents are of no significance. It is only required to comply with Pascal-XT conventions.

Package FIELD_ATTRIBUTE_PACKAGE

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY FIELD_ATTRIBUTE_PACKAGE;
   { leer }
begin
   { leer }
END.  {FIELD_ATTRIBUTE_PACKAGE}

{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
(****************************************************************)
(*      KDCS ATTRIBUTE FUNCTIONS                               *)
(*      FOR  PASCAL-XT                                         *)
(****************************************************************)
PACKAGE FIELD_ATTRIBUTE_PACKAGE;
(*********************************************************************)
(*                                                                 *)
(*              TYPE DEFINITIONS FOR FIELD ATTRIBUTES              *)
(*                                                                 *)
(*********************************************************************)
TYPE T_FIELD_ATTRIBUTE =
                          (*----------------------------------------*)
                          (* FHS NAME     EXPLANATION              *)
                          (*----------------------------------------*)
 (   INITIAL_CURSOR,      (* IC:        CURSOR IS MOVED TO SPECIFIED  *)
                          (*            FIELD AFTER OUTPUTS           *)
     UNPROTECTED,         (* UNPROT:    NOT PROTECTED AGAINST INPUT   *)
     PROTECTED_RETURNING, (* PROTRET:   PROTECTED BUT ALWAYS SENT     *)
                          (*            BACK TO THE USER PROGRAM      *)
     PRINTABLE,           (* PRINT:     PRINTABLE VIA HARDCOPY        *)
     DETECTABLE,          (* DET:       DETECTABLE (MARKABLE)         *)
     UNPROTECTED_RETURNING, (* FSET:    NOT PROTECTED AND ALWAYS SENT *)
                          (*            BACK TO THE USER PROGRAM      *)
     NUMERIC,             (* NUM:       ONLY NUMERICAL INPUT ('0',....,*)
                          (*            '9', '+', '-', '*', '/', '.', *)
                          (*            ',') FROM TERMINAL IS ALLOWED *)
     PROTECTED,           (* PROT:      PROTECTED AGAINST INPUT       *)
```

```
        RESERVED_1,            (*          DON'T USE!                    *)
        RESERVED_2,            (*          DON'T USE!                    *)
        BRIGHT,                (* BRT:     INTENSITY OF DISPLAY: HIGH    *)
        RESERVED_3,            (*          DON'T USE!                    *)
        NORMAL,                (* NORM:    INTENSITY OF DISPLAY: NORMAL  *)
        DARK,                  (* DRK:     INTENSITY OF DISPLAY: DARK    *)
        ITALIC,                (* ITAL:    ITALIC/ UNDERLINED DISPLAY    *)
        FLASHING     );        (* SIGN:    FLASHING DISPLAY              *)
                               (*---------------------------------------*)
    TYPE T_FIELD_ATTRIBUTE_SET = SET OF T_FIELD_ATTRIBUTE;
    (*$PAGE *)
    (*******************************************************************)
    (*                                                                 *)
    (*                   STANDARD ATTRIBUTE COMBINATIONS               *)
    (*                                                                 *)
    (*******************************************************************)
    CONST KCALPH  =  (. UNPROTECTED, PRINTABLE, BRIGHT .);
          KCNUME  =  (. UNPROTECTED, PRINTABLE, NUMERIC, BRIGHT .);
          KCPROT  =  (. PRINTABLE, PROTECTED, NORMAL .);
          KCUNPR  =  (. UNPROTECTED, PRINTABLE, BRIGHT .);
          KCNINT  =  (. UNPROTECTED, PRINTABLE, NORMAL .);
          KCDINT  =  (. UNPROTECTED, PRINTABLE, DARK .);
          KCHINT  =  (. UNPROTECTED, PRINTABLE, BRIGHT .);
          KCITAL  =  (. UNPROTECTED, PRINTABLE, BRIGHT, ITALIC .);
          KCSIGN  =  (. UNPROTECTED, PRINTABLE, BRIGHT, FLASHING .);
          KCDETE  =  (. PRINTABLE, DETECTABLE, PROTECTED, BRIGHT .);
          KCAUN   =  (. UNPROTECTED, PRINTABLE, NORMAL .);
          KCNUN   =  (. UNPROTECTED, PRINTABLE, NUMERIC, NORMAL .);
          KCAPN   =  (. PRINTABLE, PROTECTED, NORMAL .);
          KCNPN   =  (. PRINTABLE, PROTECTED, NORMAL .);
          KCAUD   =  (. UNPROTECTED, PRINTABLE, DARK .);
          KCNUD   =  (. UNPROTECTED, PRINTABLE, NUMERIC, DARK .);
          KCAPD   =  (. PRINTABLE, PROTECTED, DARK .);
          KCNPD   =  (. PRINTABLE, PROTECTED, DARK .);
          KCAUH   =  (. UNPROTECTED, PRINTABLE, BRIGHT .);
          KCNUH   =  (. UNPROTECTED, PRINTABLE, NUMERIC, BRIGHT .);
          KCAPH   =  (. PRINTABLE, PROTECTED, BRIGHT .);
          KCNPH   =  (. PRINTABLE, PROTECTED, BRIGHT .);
          KCAUI   =  (. UNPROTECTED, PRINTABLE, BRIGHT, ITALIC .);
          KCNUI   =  (. UNPROTECTED, PRINTABLE, NUMERIC, BRIGHT, ITALIC .);
          KCAPI   =  (. PRINTABLE, PROTECTED, NORMAL, ITALIC .);
          KCNPI   =  (. PRINTABLE, PROTECTED, NORMAL, ITALIC .);
          KCAUS   =  (. UNPROTECTED, PRINTABLE, BRIGHT, FLASHING .);
          KCNUS   =  (. UNPROTECTED, PRINTABLE, NUMERIC, BRIGHT, FLASHING .);
          KCAPS   =  (. PRINTABLE, PROTECTED, NORMAL, FLASHING .);
          KCNPS   =  (. PRINTABLE, PROTECTED, NORMAL, FLASHING .);
          KCPREM  =  (. PRINTABLE, UNPROTECTED_RETURNING, BRIGHT .);
          KCAUNP  =  (. PRINTABLE, UNPROTECTED_RETURNING, NORMAL .);
          KCNUNP  =  (. PRINTABLE, UNPROTECTED_RETURNING, NUMERIC, NORMAL .);
          KCAPNP  =  (. PROTECTED_RETURNING, PRINTABLE, NORMAL .);
          KCNPNP  =  (. PROTECTED_RETURNING, PRINTABLE, NORMAL .);
          KCAUHP  =  (. PRINTABLE, UNPROTECTED_RETURNING, BRIGHT .);
          KCNUHP  =  (. PRINTABLE, UNPROTECTED_RETURNING,NUMERIC, BRIGHT .);
          KCAPHP  =  (. PROTECTED_RETURNING, PRINTABLE, BRIGHT .);
          KCNPHP  =  (. PROTECTED_RETURNING, PRINTABLE, BRIGHT .);
          KCAUND  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, NORMAL .);
          KCNUND  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, NORMAL .);
          KCAPND  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL .);
          KCNPND  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL .);
          KCAUHD  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT .);
          KCNUHD  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT .);
          KCAPHD  =  (. PRINTABLE, DETECTABLE, PROTECTED, BRIGHT .);
          KCNPHD  =  (. PRINTABLE, DETECTABLE, PROTECTED, BRIGHT .);
          KCAUID  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT, ITALIC .);
          KCNUID  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT, ITALIC .);
          KCAPID  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL, ITALIC .);
          KCNPID  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL, ITALIC .);
          KCAUSD  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT .);
          KCNUSD  =  (. UNPROTECTED, PRINTABLE, DETECTABLE, BRIGHT .);
          KCAPSD  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL .);
          KCNPSD  =  (. PRINTABLE, DETECTABLE, PROTECTED, NORMAL .);
       END.    (*  OF FIELD_ATTRIBUTE_PACKAGE  *)
```

Package KCAPROL

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +**}
{*                     ALL RIGHTS RESERVED                        +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY Kcapro;
   { leer }
begin
   { leer }
END.  {Kcapro}
```

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +**}
{*                     ALL RIGHTS RESERVED                        +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
{*******************************************************************}
{* input            information of                               *}
{* APRO    call for  PASCAL-XT                KCAPROLS            *}
{*******************************************************************}
PACKAGE Kcapro;
type
   pic_X           = char;
   pic_XX          = packed array [1..2] of pic_X;
   pic_XXX         = packed array [1..3] of pic_X;
   pic_X_4         = packed array [1..4] of pic_X;
   pic_X_6         = packed array [1..6] of pic_X;
   pic_X_8         = packed array [1..8] of pic_X;
   pic_X_10        = packed array [1..10] of pic_X;
   pic_X_12        = packed array [1..12] of pic_X;
   pic_X_16        = packed array [1..16] of pic_X;
   pic_X_34        = packed array [1..34] of pic_X;
   pic_9           = '0'..'9';
   pic_99          = packed array [1..2] of pic_9;
   pic_999         = packed array [1..3] of pic_9;
   pic_9999        = packed array [1..4] of pic_9;
   pic_9_4_comp    = short_integer;
   record_2        = packed array [1..2] of char;
   record_4        = packed array [1..4] of char;
   record_6        = packed array [1..6] of char;
   record_7        = packed array [1..7] of char;
   record_8        = packed array [1..8] of char;
   record_9        = packed array [1..9] of char;
   record_11       = packed array [1..11] of char;
   record_12       = packed array [1..12] of char;
   record_14       = packed array [1..14] of char;
   record_15       = packed array [1..15] of char;
   record_16       = packed array [1..16] of char;
   record_18       = packed array [1..18] of char;
   record_22       = packed array [1..22] of char;
   record_24       = packed array [1..24] of char;
   record_26       = packed array [1..26] of char;
   record_32       = packed array [1..32] of char;
   record_48       = packed array [1..48] of char;
   record_50       = packed array [1..50] of char;
   record_116      = packed array [1..116] of char;
   record_146      = packed array [1..146] of char;
   REDEFINES  =                    { simulates COBOL redefinitions }
                (     v1, v2, v3, v4, v5, v6, v7, v8, v9
                ,v10, v11,v12,v13,v14,v15,v16,v17,v18,v19
                ,v20, v21,v22,v23,v24,v25,v26,v27,v28,v29
                ,v30, v31,v32,v33,v34,v35,v36,v37,v38,v39
                ,v40, v41,v42,v43,v44,v45,v46,v47,v48,v49
                ,v50, v51,v52,v53,v54,v55,v56,v57,v58,v59
                ,v60, v61,v62,v63,v64,v65,v66,v67,v68,v69
                ,v70, v71,v72,v73,v74,v75,v76,v77,v78,v79
                );
TYPE
       {03}           KCAPRO      = record case REDEFINES of
{*******************************************************************}
{*             input  information for   APRO                      *}
{*******************************************************************}
         {07}   v2 : (KCVERS  (00): pic_9_4_comp);
                                       { interface version    (1)   }
         {07}   v3 : (KCFUPOL (02): pic_X);
                                       { polarized / shared  (Y/N)  }
         {07}   v4 : (KCFUHSH (03): pic_X);
                                       { handshake           (Y/N)  }
         {07}   v5 : (KCFUCOM (04): pic_X);
                                       { commit        info  (Y/N)  }
         {07}   v6 : (KCFUCHN (05): pic_X);
                                       { chained / unchained (Y/N)  }
         {07}   v7 : (KCSECTYP(06): pic_X);
                                       { security type     (N/S/P)  }
         {07}   v8 : (KCUIDTYP(07): pic_X);
                                       { string type       (P/T/O)  }
         {07}   v9 : (KCUIDLTH(08): pic_X);
                                       { lth of userid              }
         {07}   v10: (KCUSERID(09): pic_X_16);
                                       { userid                     }
         {07}   v11: (KCPWDTYP(25): pic_X);
                                       { string type       (P/T/O)  }
```

```
          {07}   v12: (KCPWDLTH(26): pic_X);
                                            { lth of passowrd              }
          {07}   v13: (KCPSWORD(27): pic_X_16);
                                            { password                     }
          else: ();   end; {kcapro}
      end.  {kcapro}
```

Package KCCFL

```
{******************************************************************+**}
{*                                                                 +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +**}
{*                     ALL RIGHTS RESERVED                         +**}
{*                                                                 +**}
{******************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
PACKAGE BODY Kccfl;
   { leer }
begin
   { leer }
END.  {Kccfildl}
```

```
{******************************************************************+**}
{*                                                                 +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +**}
{*                     ALL RIGHTS RESERVED                         +**}
{*                                                                 +**}
{******************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
package Kccfl;
{----------------------------------------------------------------------}
{       CONTROL FIELDS FOR INPUT-EXIT                                  }
{                                                                      }
{                                            COPY:  KCCFLS             }
{----------------------------------------------------------------------}
type
   pic_X           = char;
   pic_X_8         = packed array [1..8]   of pic_X;
   pic_X_132       = packed array [1..132] of pic_X;
   pic_9_9_comp    = integer;
   REDEFINES  =                    { simulates COBOL redefinitions }
               (     v1, v2, v3, v4, v5, v6, v7, v8, v9 );
type
{01}            kccfs_type        = record case REDEFINES of
   {03}    v6 :(KCCFFNAM  (00) : pic_X_8);     { format name          }
   {03}    v7 :(KCCFREM   (08) : pic_X_8);     { remark from IFG      }
   {03}    v8 :(KCCFLOFL  (16) : pic_9_9_comp);{ length of            }
                                               { control field        }
   {03}    v9 :(KCCFFLD   (20) : pic_X_132);   { control field        }
   else: ();
             end; {kccfs_type}
type
{01}            KCCFILDL          = record case REDEFINES of
   {03}    v1 :(KCCFCREM  (00): pic_X_8);      { remark as defined    }
                                               { by IFG               }
   {03}    v2 :(KCCFCFLD  (08) : pic_X_132);   { control field        }
```

```
{03}    v3 :(KCCFNOCF  (140): pic_9_9_comp);{ number of            }
                                            { control fields       }
{03}    v5 :(KCCFS     (144): array [1..50] of kccfs_type);
                                            { array of control     }
                                            { field information    }
else: ();
          end; {kccfildl}
end.  {kccfl}
```

Package KCDADL

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                    ALL RIGHTS RESERVED                         +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
PACKAGE BODY Kcdadl;
   { leer }
begin
   { leer }
END.  {Kcdadl}
```

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                    ALL RIGHTS RESERVED                         +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
{*******************************************************************}
{*                                                                 *}
{*      Structures for resultinformation                           *}
{*      of dadm function  KCSDADM                                  *}
{*      for  PASCAL-XT                        KCdadl               *}
{*******************************************************************}
PACKAGE Kcdadl;
type
   pic_X            = char;
   pic_XX           = packed array [1..2] of pic_X;
   pic_X_3          = packed array [1..3] of pic_X;
   pic_X_6          = packed array [1..6] of pic_X;
   pic_X_8          = packed array [1..8] of pic_X;
   pic_X_16         = packed array [1..16] of pic_X;
   pic_9            = '0'..'9';
   pic_99           = packed array [1..2] of pic_9;
   pic_999          = packed array [1..3] of pic_9;
   record_9         = packed array [1..9] of char;
   record_44        = packed array [1..44] of char;
   REDEFINES  =                   { simulates COBOL redefinitions }
                (     v1, v2, v3, v4, v5, v6, v7, v8, v9,
                 v10, v11,v12,v13,v14,v15,v16,v17,v18,v19,
                 v20, v21,v22,v23,v24,v25                );
TYPE
       {03}           KCDADM1      = record case REDEFINES of
         {05}   v1 : (KCDAGUS (00): pic_x_8);
                                              { USER ID              }
         {05}   v2 : (KCDADPID(08): pic_x_8);
                                          { DPUT ID                }
         {05}   v3 : (KCDAGTIM(16): record_9);
                                          { generation time        }
           {07} v4 : (KCDAGDOY(16): pic_x_3);
                                          { day of year            }
           {07} v5 : (KCDAGHR (19): pic_xx);
                                          { hour                   }
           {07} v6 : (KCDAGMIN(21): pic_xx);
                                          { minute                 }
           {07} v7 : (KCDAGSEC(23): pic_xx);
                                          { Second                 }
         {05}   v8 : (KCDASTIM(25): record_9);
                                          { desired start time     }
           {07} v9 : (KCDASDOY(25): pic_x_3);
                                          { day of year            }
           {07} v10: (KCDASHR (28): pic_xx);
                                          { hour                   }
           {07} v11: (KCDASMIN(30): pic_xx);
                                          { minute                 }
           {07} v12: (KCDASSEC(32): pic_xx);
                                          { second                 }
         {05}   v13: (KCDAPMSG(34): pic_x);
                                          { positive               }
                                          { acknowl. job           }
         {05}   v14: (KCDANMSG(35): pic_x);
                                          { negative               }
                                          { acknowl. job           }
         else: ();  end;  {kcdadml}
end.  {kcdadl}
```

Package KCDFL

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992 +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY KCDFL;
   { leer }
begin
   { leer }
END.  {KCDFL}
```

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992 +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
 {*************************************************************}
 {*    KDCS SREEN OUTPUT FUNCTIONS                             *}
 {*    FUER PASCAL-XT           PACKAGE:   KCDFL               *}
 {*                                                            *}
 {*************************************************************}
 PACKAGE   KCDFL;                           { KDCS_DEVICE_FEATURE }
 TYPE        KCDFL_FIELD =  SHORT_INTEGER;
 CONST       KCREPL      =    001;          {('0000000000000001'B)}
                                            { CLEAR SCREEN AND    }
                                            { DISPLAY FORMAT      }
             KCRESTRT    =  #0  001;        {('0000000000000001'B)}
                                            { SCREEN RESTART      }
                                            { WITH PEND RS        }
             KCERAS      =  #0  002;        {('0000000000000010'B)}
                                            { ERASE UNPROTECTED   }
                                            { FIELDS              }
             KCALARM     =  #0  004;        {('0000000000000100'B)}
                                            { BEL-FUNCTION        }
                                            {                     }
             KCREPR      =  #0  008;        {('0000000000001000'B)}
                                            { OUTPUT ON LOCAL     }
                                            { PRINTER             }
             KCEXTEND    =    000;          {('0010000000000000'B)}
                                            { EXTENDED LINE MODE  }
                                            {                     }
             KCCARD      =    000;          {('0100000000000000'B)}
                                            { NEXT INPUT FROM     }
                                            { CARD READER         }
 END.  { KCDFL }
```

Package KCINIL

```
{*************************************************************+**}
{*                                                             +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1994  +**}
{*                    ALL RIGHTS RESERVED                      +**}
{*                                                             +**}
{*************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY Kcinil;
   { leer }
begin
   { leer }
END.  {Kcinil}
```

```
{*************************************************************+**}
{*                                                             +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1993  +**}
{*                    ALL RIGHTS RESERVED                      +**}
{*                                                             +**}
{*************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
{***************************************************************}
{* input and output information of                            *}
{* INIT PU call for  PASCAL-XT                KCINIL            *}
{***************************************************************}
PACKAGE Kcinil;
type
   pic_X           = char;
   pic_XX          = packed array [1..2] of pic_X;
   pic_XXX         = packed array [1..3] of pic_X;
   pic_X_4         = packed array [1..4] of pic_X;
   pic_X_6         = packed array [1..6] of pic_X;
   pic_X_8         = packed array [1..8] of pic_X;
   pic_X_10        = packed array [1..10] of pic_X;
   pic_X_12        = packed array [1..12] of pic_X;
   pic_X_34        = packed array [1..34] of pic_X;
   pic_9           = '0'..'9';
   pic_99          = packed array [1..2] of pic_9;
   pic_999         = packed array [1..3] of pic_9;
   pic_9999        = packed array [1..4] of pic_9;
   pic_9_4_comp    = short_integer;
   record_2        = packed array [1..2] of char;
   record_4        = packed array [1..4] of char;
   record_6        = packed array [1..6] of char;
   record_7        = packed array [1..7] of char;
   record_8        = packed array [1..8] of char;
   record_9        = packed array [1..9] of char;
   record_11       = packed array [1..11] of char;
   record_12       = packed array [1..12] of char;
   record_14       = packed array [1..14] of char;
   record_15       = packed array [1..15] of char;
   record_16       = packed array [1..16] of char;
   record_18       = packed array [1..18] of char;
   record_22       = packed array [1..22] of char;
   record_24       = packed array [1..24] of char;
   record_26       = packed array [1..26] of char;
   record_32       = packed array [1..32] of char;
   record_48       = packed array [1..48] of char;
   record_50       = packed array [1..50] of char;
   record_116      = packed array [1..116] of char;
   record_146      = packed array [1..146] of char;
   REDEFINES  =                   { simulates COBOL redefinitions }
                (     v1, v2, v3, v4, v5, v6, v7, v8, v9
                ,v10, v11,v12,v13,v14,v15,v16,v17,v18,v19
                ,v20, v21,v22,v23,v24,v25,v26,v27,v28,v29
                ,v30, v31,v32,v33,v34,v35,v36,v37,v38,v39
                ,v40, v41,v42,v43,v44,v45,v46,v47,v48,v49
                ,v50, v51,v52,v53,v54,v55,v56,v57,v58,v59
                ,v60, v61,v62,v63,v64,v65,v66,v67,v68,v69
                ,v70, v71,v72,v73,v74,v75,v76,v77,v78,v79
                );
TYPE
       {03}           KCINI1      = record case REDEFINES of
{*******************************************************************}
{*              input  information for   KCOM = PU                *}
{*******************************************************************}
        {05}    v1 : (KCINPUT(00): record_16);
                                        { input information         }
         {07}   v2 : (KCINIVER(00): pic_9_4_comp);
                                        { interface version    (1)   }
         {07}   v3 : (KCDATE  (02): pic_X);
                                        { date and time info  (Y/N)  }
         {07}   v4 : (KCAPPL  (03): pic_X);
                                        { application   info  (Y/N)  }
         {07}   v5 : (KCLOCALE(04): pic_X);
                                        { locale        info  (Y/N)  }
         {07}   v6 : (KCOSITP (05): pic_X);
                                        { OSI TP        info  (Y/N)  }
         {07}   v7 : (KCFILLIN(06): pic_X_10);
                                        { not used                   }
{*******************************************************************}
{*              output information for   KCOM = PU                *}
{*******************************************************************}
        {05}    v8 : (KCOUTPUT(16): record_146);
{*******************************************************************}
{*              general information      generated maximal length   *}
{*******************************************************************}
```

```
          {07}   v9 : (KCGPAB  (16): pic_9_4_comp);
                                             { of spab              }
          {07}   v10: (KCGNB   (18): pic_9_4_comp);
                                             { of nb                }
{*******************************************************************}
{*              date and time information                          *}
{*******************************************************************}
         {05}    v11: (KCDTTM  (20): record_48);
          {07}   v12: (KCADTTM (20): record_18);
                                             { date/time of         }
                                             { application start    }
           {09}  v13: (KCADATE (20): record_11);
                                             { date:                }
            {11} v14: (KCADAY  (20): pic_99);
                                             { day                  }
            {11} v15: (KCAMONTH(22): pic_99);
                                             { month                }
            {11} v16: (KCAYEAR (24): pic_9999);
                                             { year                 }
            {11} v17: (KCADOY  (28): pic_999);
                                             { day of year          }
           {09}  v18: (KCATIME (31): record_6);
                                             { time:                }
            {11} v19: (KCAHOUR (31): pic_99);
                                             { hour                 }
            {11} v20: (KCAMIN  (33): pic_99);
                                             { minute               }
            {11} v21: (KCASEC  (35): pic_99);
                                             { second               }
           {09}  v22: (KCASEAS (37): pic_X);
                                             { season      (w/s)    }
          {07}   v23: (KCPDTTM (38): record_18);
                                             { date/time of         }
                                             { program start        }
           {09}  v24: (KCPDATE (38): record_11);
                                             { date:                }
            {11} v25: (KCPDAY  (38): pic_99);
                                             { day                  }
            {11} v26: (KCPMONTH(40): pic_99);
                                             { month                }
            {11} v27: (KCPYEAR (42): pic_9999);
                                             { year                 }
            {11} v28: (KCPDOY  (46): pic_999);
                                             { day of year          }
           {09}  v29: (KCPTIME (49): record_6);
                                             { time:                }
            {11} v30: (KCPHOUR (49): pic_99);
                                             { hour                 }
            {11} v31: (KCPMIN  (51): pic_99);
```

```
                                                    { minute                }
          {11} v32: (KCPSEC  (53): pic_99);
                                                    { second                }
         {09}  v33: (KCPSEAS (55): pic_X);
                                                    { season      (w/s)     }
        {07}   v34: (KCTMZONE(56): pic_X_12);
                                                    { time zone             }
{**********************************************************************}
{******     application information                                   *}
{**********************************************************************}
       {05}    v35: (KCAPINF (68): record_50);
                                                    {                       }
        {07}   v36: (KCAPPLNM(68): pic_X_8);
                                                    { application name      }
        {07}   v37: (KCHOSTNM(76): pic_X_8);
                                                    { HOST name             }
        {07}   v38: (KCPTRMNM(84): pic_X_8);
                                                    { PTRM name             }
        {07}   v39: (KCPRONM (92): pic_X_8);
                                                    { processor name        }
        {07}   v40: (KCBCAPNM(100): pic_X_8);
                                                    { BCAM applname         }
        {07}   v41: (KCVERS  (108): pic_X_6);
                                                     { UTM-Version          }
        {07}   v42: (KCIVER  (114): pic_9_4_comp);
                                                     { Interface-version    }
        {07}   v43: (KCIVAR  (116): pic_X);
                                                     { bs2 or sinix         }
        {07}   v44: (KCFILL1 (117): pic_X);
                                                     { not used             }
{**********************************************************************}
{******         locale information                                    *}
{**********************************************************************}
       {05}    v45: (KCLOCINF(118): record_22);
                                                    { locale information    }
        {07}   v46: (KCUSLOC (118): record_12);
                                                    { locale of user        }
         {09}  v47: (KCUSLANG(118): pic_XX);
                                                    {    language id        }
         {09}  v48: (KCUSTERR(120): pic_XX);
                                                    {    territory id       }
         {09}  v49: (KCUSCCSN(122): pic_X_8);
                                                    { coded char set name   }
         {09}  v50: (KCFILL2 (130): pic_X_8);
                                                    { not used              }
        {07}   v51: (KCCSINFO(138): record_7);
                                                    { info for XHCS support }
         {09}  v52: (KCCURCCS(138): pic_X_8);
                                                    { ccsname of current msg}
          {09}  v53: (KCDEVCAP(146): pic_X);
                                         { '7'/'8': 7-/8-bit-dev }
         {07}   v54: (KCFILL3 (147): pic_X);
                                          { not used            }
{*******************************************************************}
{******         OSI TP information                                *}
{*******************************************************************}
        {05}    v55: (KCOSIINF(148): record_8);
                                         { OSI TP information    }
         {07}   v56: (KCFUPOL (148): pic_X);
                                         { polarized fu          }
         {07}   v57: (KCFUHSH (149): pic_X);
                                         { handshake fu          }
         {07}   v58: (KCFUCM  (150): pic_X);
                                         { commit fu             }
         {07}   v59: (KCFUCHND(151): pic_X);
                                         { chained fu            }
         {07}   v60: (KCENDTA (152): pic_X);
                                         { end transaction ind   }
         {07}   v61: (KCSEND  (153): pic_X);
                                         { send to client        }
         {07}   v62: (KCFILL4 (154): pic_XX);
                                          { not used            }
         else: ();   end; {kcinil}
end.  {kcinil}
```

Package KCINL

```
{*****************************************************************+***}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+***}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY Kcinl;
   { leer }
begin
   { leer }
END.  {Kcinl}
```

```
{*****************************************************************+***}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+***}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
{*******************************************************************}
{*      return information of INFO call                          *}
{*      for  PASCAL-XT                        KCINL              *}
{*******************************************************************}
PACKAGE Kcinl;
type
   pic_X           = char;
   pic_XX          = packed array [1..2] of pic_X;
   pic_XXX         = packed array [1..3] of pic_X;
   pic_X_4         = packed array [1..4] of pic_X;
   pic_X_6         = packed array [1..6] of pic_X;
   pic_X_8         = packed array [1..8] of pic_X;
   pic_X_10        = packed array [1..10] of pic_X;
   pic_X_34        = packed array [1..34] of pic_X;
   pic_9           = '0'..'9';
   pic_99          = packed array [1..2] of pic_9;
   pic_999         = packed array [1..3] of pic_9;
   pic_9_4_comp    = short_integer;
   record_2        = packed array [1..2] of char;
   record_4        = packed array [1..4] of char;
   record_6        = packed array [1..6] of char;
   record_8        = packed array [1..8] of char;
   record_9        = packed array [1..9] of char;
   record_12       = packed array [1..12] of char;
   record_14       = packed array [1..14] of char;
   record_15       = packed array [1..15] of char;
   record_16       = packed array [1..16] of char;
   record_24       = packed array [1..24] of char;
   record_26       = packed array [1..26] of char;
   record_32       = packed array [1..32] of char;
   record_50       = packed array [1..50] of char;
   record_65       = packed array [1..65] of char;
   record_116      = packed array [1..116] of char;
   REDEFINES  =                    { simulates COBOL redefinitions }
                (     v1, v2, v3, v4, v5, v6, v7, v8, v9
                ,v10, v11,v12,v13,v14,v15,v16,v17,v18,v19
                ,v20, v21,v22,v23,v24,v25,v26,v27,v28,v29
                ,v30, v31,v32,v33,v34,v35,v36,v37,v38,v39
                ,v40, v41,v42,v43,v44,v45,v46,v47,v48,v49
                ,v50, v51,v52,v53,v54,v55,v56,v57,v58,v59
                ,v60, v61,v62,v63,v64,v65,v66,v67,v68,v69
                ,v70, v71,v72,v73,v74,v75,v76,v77,v78,v79
                );
TYPE
       {03}           KCINFl      = record case REDEFINES of
        {05}    v1 : (KCRETINF(00): record_65);
                                        { maximum size of return info}
{**********************************************************************}
{*              return information for   KCOM = DT                   *}
{**********************************************************************}
        {05}    v2 : (KCDATTIM(00): record_65);
         {07}   v3 : (KCDTAS  (00): record_15);
                                             { date/time of           }
                                             { application start      }
          {09}  v4 : (KCDATAS (00): record_9);
                                             { date:                  }
           {11} v5 : (KCTAGAS (00): pic_99);
                                             { day                    }
           {11} v6 : (KCMONAS (02): pic_99);
                                             { month                  }
           {11} v7 : (KCJHRAS (04): pic_99);
                                             { year                   }
           {11} v8 : (KCTJHAS (06): pic_999);
                                             { day of year            }
          {09}  v9 : (KCUHRAS (09): record_6);
                                             { time:                  }
           {11} v10: (KCSTDAS (09): pic_99);
                                             { hour                   }
           {11} v11: (KCMINAS (11): pic_99);
                                             { minute                 }
           {11} v12: (KCSEKAS (13): pic_99);
                                             { second                 }
{* * * *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  * *}
         {07}   v13: (KCDTAK  (15): record_15);
                                             { date/time of           }
                                             { program start          }
          {09}  v14: (KCDATAK (15): record_9);
                                         { date:                 }
           {11} v15: (KCTAGAK (15): pic_99);
                                         { day                   }
           {11} v16: (KCMONAK (17): pic_99);
                                         { month                 }
           {11} v17: (KCJHRAK (19): pic_99);
                                         { year                  }
           {11} v18: (KCTJHAK (21): pic_999);
                                         { day of year           }
          {09}  v19: (KCUHRAK (24): record_6);
                                         { time:                 }
           {11} v20: (KCSTDAK (24): pic_99);
                                         { hour                  }
           {11} v21: (KCMINAK (26): pic_99);
                                         { minute                }
           {11} v22: (KCSEKAK (28): pic_99);
                                         { second                }
{* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *}
         {07}       {FILLER  (27): pic_x_20 }
                                         { not used              }
{*******************************************************************}
{******        return information for   KCOM = SI               *}
{*******************************************************************}
        {05}    v23: (KCSYSINF(00): record_65);
                                         { system information    }
         {07}   v24: (KCAPPLNM(00): pic_x_8);
                                         { application name      }
         {07}   v25: (KCHOSTNM(08): pic_x_8);
                                         { HOST name             }
         {07}   v26: (KCPTRMNM(16): pic_x_8);
                                         { PTRM name             }
         {07}   v27: (KCPRONM (24): pic_x_8);
                                         { processor name        }
         {07}   v28: (KCBCAPNM(32): pic_x_8);
                                         { BCAM applname         }
         {07}   v29: (KCVERS  (40): pic_x_6); { UTM-Version           }
         {07        (KCIVER  (46): pic_x-2); { Interface-version     }
         {07        (KCIVAR  (48): pic_x-1); { bs2 or sinix          }
         {07        (FILLER  (49): pic_x-1); { not used              }
{*******************************************************************}
{******        return information for   KCOM = PC               *}
{*******************************************************************}
        {05}    v30: (KCPREINF(00): record_65);
                                         { information of prede- }
                                         { cessor conversation   }
         {07}   v31: (KCPFN   (00): pic_x_8);
                                         { format name           }
         {07}   v32: (KCPNXTAC(08): pic_x_8);
```

```
                                              { next TAC               }
         {07}   v33: (KCPCVTAC(16): pic_x_8);
                                              { conversation TAC       }
         {07}   v34: (KCPLDATE(24): record_9);
                                              { date of last           }
                                              { program run:           }
          {09}  v35: (KCPLDAY (24): pic_99);
                                              { day                    }
          {09}  v36: (KCPLMON (26): pic_99);
                                              { month                  }
          {09}  v37: (KCPLYEAR(28): pic_99);
                                              { year                   }
          {09}  v38: (KCPLDOY (30): pic_999);
                                              { day of year            }
         {07}   v40: (KCPLTIME(33): record_9);
                                              { time of last           }
                                              { program run:           }
          {09}  v41: (KCPLHOUR(33): pic_99);
                                              { hour                   }
          {09}  v42: (KCPLMIN (35): pic_99);
                                              { minute                 }
          {09}  v43: (KCPLSEC (37): pic_99);
                                              { second                 }
          {09   v44: (FILLER  (39): pic_X_26);                         }
                                              { not used               }
{*****************************************************************}
{******         return information for   KCOM = LO               *}
{*****************************************************************}
        {05}    v45: (KCLOCINF(00): record_65);
                                              { locale information     }
         {07}   v46: (KCLTLOC(00):  record_12);
                                              { locale of spec. lterm }
          {09}  v47: (KCLTLANG(00): pic_xx);
                                              {    language id         }
          {09}  v48: (KCLTTERR(02): pic_xx);
                                              {    territory id        }
          {09}  v49: (KCLTCCSN(04): pic_x_8);
                                              { coded char set name    }
         {07    v50: (FILLER  (12): pic_X_8);                          }
                                              { not used               }
         {07}   v51: (KCAPLOC(20):  record_12);
                                              { locale of application }
          {09}  v52: (KCAPLANG(20): pic_xx);
                                              {    language id         }
          {09}  v53: (KCAPTERR(22): pic_xx);
                                              {    territory id        }
          {09}  v54: (KCAPCCSN(24): pic_x_8);
                                              { coded char set name    }
         {07    v55: (FILLER  (32): pic_X_8);                          }
```

```
                                               { not used             }
          {07}   v56: (KCCSINFO(40): record_26);
                                               { info for XHCS support }
           {09}  v57: (KCDEFCCS(40): pic_x_8);
                                               { default ccs          }
           {09}  v58: (KCCCSNO(48):  pic_x);
                                               { no of supported ccs   }
           {09}  v59: (KCCCSTAB(49): record_16);
                                               { table of supported ccs}
            {11} v60: (KCVAR1(49):   pic_x);
                                               { iso var no 1. supp ccs}
            {11} v61: (KCVAR2(50):   pic_x);
                                               { iso var no 2. supp ccs}
            {11} v62: (KCVAR3(51):   pic_x);
                                               { iso var no 3. supp ccs}
            {11} v63: (KCVAR4(52):   pic_x);
                                               { iso var no 4. supp ccs}
            {11} v64: (KCVAR5(53):   pic_x);
                                               { iso var no 5. supp ccs}
            {11} v65: (KCVAR6(54):   pic_x);
                                               { iso var no 6. supp ccs}
            {11} v66: (KCVAR7(55):   pic_x);
                                               { iso var no 7. supp ccs}
            {11} v67: (KCVAR8(56):   pic_x);
                                               { iso var no 8. supp ccs}
            {11} v68: (KCVAR9(57):   pic_x);
                                               { iso var no 9. supp ccs}
            {11} v69: (KCVAR10(58):  pic_x);
                                               { iso var no 10 supp ccs}
            {11} v70: (KCVAR11(59):  pic_x);
                                               { iso var no 11 supp ccs}
            {11} v71: (KCVAR12(60):  pic_x);
                                               { iso var no 12 supp ccs}
            {11} v72: (KCVAR13(61):  pic_x);
                                               { iso var no 13 supp ccs}
            {11} v73: (KCVAR14(62):  pic_x);
                                               { iso var no 14 supp ccs}
            {11} v74: (KCVAR15(63):  pic_x);
                                               { iso var no 15 supp ccs}
            {11} v75: (KCVAR16(64):  pic_x);
                                               { iso var no 16 supp ccs}
          else: ();   end; {kcinfl}
  end.  {kcinl}
```

Package KCINPL

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992 +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
PACKAGE BODY Kcinpl;
   { leer }
begin
   { leer }
END.  {Kcinputl}
```

```
{*****************************************************************+**}
{*                                                                +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992 +**}
{*                   ALL RIGHTS RESERVED                          +**}
{*                                                                +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0       +**}
package Kcinpl;
{--------------------------------------------------------------------}
{       PARAMETER AREA FOR INPUT-EXIT                                }
{                                                                    }
{                                            COPY:  KCINPC           }
{--------------------------------------------------------------------}
type
   pic_X          = char;
   pic_XX         = packed array [1..2] of pic_X;
   pic_XXX        = packed array [1..3] of pic_X;
   pic_X_4        = packed array [1..4] of pic_X;
   pic_X_6        = packed array [1..6] of pic_X;
   pic_X_8        = packed array [1..8] of pic_X;
   pic_9          = '0'..'9';
   pic_99         = packed array [1..2] of pic_9;
   pic_999        = packed array [1..3] of pic_9;
   pic_9_4_comp   = short_integer;
   record_2       = packed array [1..2] of char;
   record_3       = packed array [1..3] of char;
   record_4       = packed array [1..4] of char;
   record_6       = packed array [1..6] of char;
   record_8       = packed array [1..8] of char;
   record_9       = packed array [1..9] of char;
   record_14      = packed array [1..14] of char;
   record_16      = packed array [1..16] of char;
   record_24      = packed array [1..24] of char;
   record_32      = packed array [1..32] of char;
   record_116      = packed array [1..116] of char;
   REDEFINES  =                   { simulates COBOL redefinitions }
               (     v1, v2, v3, v4, v5, v6, v7, v8, v9,
                v10, v11,v12,v13,v14,v15,v16,v17,v18,v19,
                v20, v21,v22,v23,v24,v25,v26,v27,v28,v29,
                v30, v31,v32,v33,v34,v35,v36,v37,v38,v39,
                v40, v41,v42,v43,v44,v45,v46,v47,v48,v49,
                v50, v51,v52,v53,v54,v55,v56,v57,v58,v59,
                v60);
type


{01}            KCINPUTL          = record case REDEFINES of
   {03}    v1 :(KCIFCH    (00): pic_X_8);      { first 8 characters   }
                                               { if input message     }
   {03}    v2 :(KCIFN     (08): pic_X_8);      { format name          }
   {03}    v3 :(KCICVTAC  (16): pic_x_8);      { conversation TAC     }
   {03}    v4 :(KCICVST   (24): pic_XX);       { conversation state   }
   {03}    v5 :(KCIFKEY   (26): pic_9_4_comp); { f_key                }
   {03}    v6 :(KCIKKEY   (28): pic_9_4_comp); { k_key                }
   {03}    v8 :(KCICFINF  (30): pic_XX);       { control field        }
                                               { information          }
   {03}    v9 :(KCILTERM  (32): pic_x_8);      { current LTERM        }
   {03}    v10:(KCIUSER   (40): pic_x_8);      { current USER         }
   {03     v11:(FILLER    (48): pic_X_32);       reserved             }
   {03}    v12:(KCINTAC   (80): pic_X_8);      { next TAC             }
   {03}    v13:(KCINCMD   (80): pic_X_8);      { next command         }
   {03}    v14:(KCICCD    (88): pic_XX);       { continuation code    }
   {03}    v15:(KCICUT    (90): pic_X);        { cut TAC  (Y/N)       }
   {       v16:(FILLER    (91): pic_X);          reserved             }
   {03}    v17:(KCIERRCD  (92): pic_X_4);      { error code           }
   {       v18:(FILLER    (96): pic_X_44);       reserved             }
   else: ();
             end; {kcinputl}
   end.  {kcinpl}
```

Package KCKBL

```
{*****************************************************************+**}
{*                                                                 +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                           +**}
{*                                                                 +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
PACKAGE BODY Kckbl;
   { leer }
begin
   { leer }
END.  {Kckbl}
```

```
{*****************************************************************+**}
{*                                                                 +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                   ALL RIGHTS RESERVED                           +**}
{*                                                                 +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
(***********************************************************)
(*                                                         *)
(*    KDCS definitions  for PASCAL-XT                      *)
(*                                          KCKBL          *)
(*    this package includes the description of:            *)
(*         KDCS-communication area,                        *)
(*         KDCS-param area,                                *)
(*         KDCS-operation codes                            *)
(***********************************************************)
package Kckbl;
{----------------------------------------------------------}
{----------------      general  remarks        ----------------}
{----------------------------------------------------------}
{                                                          }
{  This modul is made of constants and type definitions. It is  }
{  used for the user specific UTM adapter (body is empty).      }
{                                                          }
{  This modul is the same in each UTM application; it must not  }
{  be changed by UTM user!                                 }
{                                                          }
{  All structures are based on COBOL copy members.         }
{                                                          }
{                                                          }
{----------------------------------------------------------}
{----------------------------------------------------------}
{----------   type definitions analogous to COBOL  -------------}
{----------------------------------------------------------}
type
   pic_X          = char;
   pic_XX         = packed array [1..2] of pic_X;
   pic_XXX        = packed array [1..3] of pic_X;
   pic_X_4        = packed array [1..4] of pic_X;
   pic_X_5         = packed array [1..5] of pic_X;
   pic_X_6         = packed array [1..6] of pic_X;
   pic_X_8         = packed array [1..8] of pic_X;
   pic_9           = '0'..'9';
   pic_99          = packed array [1..2] of pic_9;
   pic_999         = packed array [1..3] of pic_9;
   pic_9_4_comp    = short_integer;
   record_2        = packed array [1..2] of char;
   record_3        = packed array [1..3] of char;
   record_4        = packed array [1..4] of char;
   record_6        = packed array [1..6] of char;
   record_8        = packed array [1..8] of char;
   record_9        = packed array [1..9] of char;
   record_14       = packed array [1..14] of char;
   record_16       = packed array [1..16] of char;
   record_24       = packed array [1..24] of char;
   record_32       = packed array [1..32] of char;
   record_116      = packed array [1..116] of char;
   REDEFINES  =                   { simulates COBOL redefinitions }
              (     v1, v2, v3, v4, v5, v6, v7, v8, v9,
               v10, v11,v12,v13,v14,v15,v16,v17,v18,v19,
               v20, v21,v22,v23,v24,v25,v26,v27,v28,v29,
               v30, v31,v32,v33,v34,v35,v36,v37,v38,v39,
               v40, v41,v42,v43,v44,v45,v46,v47,v48,v49,
               v50, v51,v52,v53,v54,v55,v56,v57,v58,v59,
               v60, v61,v62,v63,v64,v65,v66,v67,v68,v69,
               v70);
{----------------------------------------------------------------}
{                                                                }
{  By this kind of definition and redefinition with              }
{  representation specification it is possible to address        }
{  structure, substructure, field without full qualification.    }
{                                                                }
{  i.e.:         KCVGST may be addressed:                        }
{                                                                }
{                     <var>.KCVGST                               }
{                                                                }
{                                                                }
{                full qualification should be:                   }
{                                                                }
{                     <var>.KCRFELD.KCRST.KCVGST                 }
{                                                                }
{                                                                }
{----------------------------------------------------------------}
{$PAGE}
{----------------------------------------------------------------}
{-------------    KDCS  operation  codes          --------------}
{----------------------------------------------------------------}
type    kc_opcode = pic_X_4;                       { KDCS operation codes }
const   INIT  = kc_opcode ('I','N','I','T');       { initialize program run }
        PEND  = kc_opcode ('P','E','N','D');       { program run end }
        RSET  = kc_opcode ('R','S','E','T');       { reset transaction }
        MGET  = kc_opcode ('M','G','E','T');       { read dialog message (part) }
        MPUT  = kc_opcode ('M','P','U','T');       { write dialogmessage (part) }
        FGET  = kc_opcode ('F','G','E','T');     { read asynchronous }
                                                 { message (part) }
        FPUT  = kc_opcode ('F','P','U','T');     { write asynchronous }
                                                 { message (part) }
        SGET  = kc_opcode ('S','G','E','T');     { read secondary storage }
        SPUT  = kc_opcode ('S','P','U','T');     { write secondary storage }
        SREL  = kc_opcode ('S','R','E','L');     { release secondary storage }
        GTDA  = kc_opcode ('G','T','D','A');     { read terminal specific }
                                                 { secondary storage }
        PTDA  = kc_opcode ('P','T','D','A');     { write terminal specific }
                                                 { secondary storage }
        UNLK  = kc_opcode ('U','N','L','K');     { unlock global }
                                                 { secondary storage }
        LPUT  = kc_opcode ('L','P','U','T');     { write record to }
                                                 { user log file }
        INFO  = kc_opcode ('I','N','F','O');     { call info-services }
        DPUT  = kc_opcode ('D','P','U','T');     { write time-driven }
                                                 { asynchr. message (part) }
        APRO  = kc_opcode ('A','P','R','O');     { adressing a job receiving }
                                                 { conversation }
        MCOM  = kc_opcode ('M','C','O','M');     { define message complex }
        SIGN  = kc_opcode ('S','I','G','N');     { use sign-on functions }
        DADM  = kc_opcode ('D','A','D','M');     { administration of     }
                                                 { asynchronous message }
        PADM  = kc_opcode ('P','A','D','M');     { administration of     }
                                                 { printer              }
        PGWT  = kc_opcode ('P','G','W','T');     { program wait         }
{$PAGE}
{----------------------------------------------------------------}
{  KDCS communication area   (KB)       UTM V04.0                }
{----------------------------------------------------------------}
type
{01}           KCKB        = record case REDEFINES of
 {03}    v1 :(KCKBKOPF(00): record_116);          { header of KDCS communication area }
  {05}   v2 :(KCBENID (00): pic_X_8);             { user identification }
  {05}   v3 :(KCVORG  (08): record_24);           { conversation-specific }
                                                  { data fields:     }
   {07}  v4 :(KCTACVG (08): pic_X_8);             { transcation code }
   {07}  v5 :(KCDATVG (16): record_9);            { date:            }
    {09} v6 :(KCTAGVG (16): pic_XX);              {    day           }
    {09} v7 :(KCMONVG (18): pic_XX);              {    Month         }
    {09} v8 :(KCJHRVG (20): pic_XX);              {    year          }
    {09} v9 :(KCTJHVG (22): pic_XXX);             {    day of year }
   {07}  v10:(KCUHRVG (25): record_6);            { time:            }
    {09} v11:(KCSTDVG (25): pic_XX);              {    hour          }
    {09} v12:(KCMINVG (27): pic_XX);              {    minute        }
    {09} v13:(KCSEKVG (29): pic_XX);              {    Sekond        }
   {07}  v14:(KCKNZVG (31): pic_X);               { conversation id }
  {05}   v15:(KCAKTUEL(32): record_16);           { data specific to current }
                                                  { program run:     }
   {07}  v16:(KCTACAL (32): pic_X_8);             { transaction code }
   {07}  v17:(KCUHRAL (40): record_6);            { time:            }
    {09} v18:(KCSTDAL (40): pic_XX);              {    hour          }
    {09} v19:(KCMINAL (42): pic_XX);              {    minute        }
    {09} v20:(KCSEKAL (44): pic_XX);              {    second      }
   {07}  v21:(KCAUSWEIS(46):pic_X);               { A = card in reader }
   {07}  v22:(KCTAIND (47): pic_X );              { transaction indicator }
  {05}   v23:(KCLOGTER(48): pic_X_8);             { name of UTM terminal (= LTERM) }
  {05}   v24:(KCTERMN (56): pic_XX);              { device type of physical terminal }
  {05}   v25:(KCLKBPB (58): pic_9_4_comp);        { maximum length of KB program area }
  {05}   v26:(KCSTA   (60): record_3);            { stack information: }
   {07}  v27:(KCHSTA  (60): pic_9_4_comp);        { current stack level   }
   {07}  v28:(KCDSTA  (62): pic_X);               { change in stack level }
   {07}      {FILLER  (63): pic_X }
  {05}   v29:(KCPRIND (64): pic_X);               { program indicator }
  {05}   v30:(KCOF1   (65): pic_X);               { OSI-TP function1 }
  {05}   v31:(KCOF2   (66): pic_X);               { OSI-TP function2}
  {05}   v32:(KCTARB  (67): pic_X);               { ta is marked rb }
  {05}   v33:(KCYEARVG(68): pic_X_4);             { year start conversation }
  {05}       {FILLER  (72): pic_X_12 }
    {---------------------------------------------}
    {----   KDCS  return area   ------------------}
    {---------------------------------------------}
                                          { contains returninfo from UTM  }
  {05}   v41:(KCRI    (84): pic_XX);              { return identification }
  {05}   v42:(KCRDF   (84): pic_9_4_comp);        { return device feature }
  {05}   v43:(KCRLM   (86): pic_9_4_comp);        { return length }
  {05}   v44:(KCRINFCC(88): pic_XXX);             { info call return code }
  {05}   v45:(KCRSTAT (88): record_4);
   {07}  v46:(KCRSTATE(88): pic_XX);              { conversation and transaction status }
   {07}  v47:(KCRST   (88): record_2);
    {09} v48:(KCVGST  (88): pic_X);               { conversation status }
    {09} v49:(KCTAST  (89): pic_X);               { transaction status  }
   {07}      {FILLER  (90): pic_X }
  {05}   v50:(KCRSIGN (88): record_3);            { status of sign-on: }
   {07}  v51:(KCRSIGN1(88): pic_X);               { primary code       }
   {07}  v52:(KCRSIGN2(89): pic_XX);              { secondary code     }
  {05}   v53:(KCRMGT  (91): pic_X);               {return info mget }
  {05}   v55:(KCRC    (92): record_8);            { return error codes:  }
   {07}  v56:(KCRCCC  (92): pic_XXX);             { KDCS error code }
   {07}  v57:(KCRCKZ  (95): pic_X);               { indicator:  P=Produktion , T=UTM-T }
   {07}  v58:(KCRCDC  (96): pic_X_4);             { additional error code }
                                                { from UTM (not compatible) }
  {05}   v59:(KCRMF  (100): pic_X_8);             { return message format }
  {05}   v60:(KCRPI  (108): pic_X_8);             { return conversation id }
  {05}   v61:(KCRUS  (108): pic_X_8);             { return user (sign st)  }
 {03     v70:(KCKBPRG(116): KDCS KB program area, to declare      }
 {                          by user, including KCKBL              }
         else: ();          end; { KCKB }
{$PAGE}
{----------------------------------------------------------------}
{  KDCS param area (PA)                 UTM V04.0                }
{----------------------------------------------------------------}
type
{01       KCSPAB       : standard primary working area; to       }
{                        declare by user, including KCPAL        }
  {03}        KCPAL       = record case REDEFINES of
   {05}  v1 :(KCOP    (00): kc_opcode);           { operation code  }
```

```
   {05}  v2 :(KCOM    (04): pic_XX);             { operation modification }
   {05}  v3 :(KCLA    (06): pic_9_4_comp);       { length of area  }
   {05}  v4 :(KCLKBPRG(06): pic_9_4_comp);       { length of KB program area }
   {05}  v5 :(KCLM    (08): pic_9_4_comp);       { length of message }
   {05}  v6 :(KCLPAB  (08): pic_9_4_comp);       { length of SPAB    }
   {05}  v7 :(KCRN    (10): pic_X_8);            { reference name:   }
                                                 { TAC/LTERM/storage area }
   {05}  v8 :(KCMF    (18): pic_X_8);            { message format    }
   {05}  v9 :(KCLT    (18): pic_X_8);            { name of UTM terminal (= LTERM) }
   {05}  v10:(KCUS    (18): pic_X_8);            { user id  }
   {05}  v11:(KCPA    (18): pic_X_8);            { name of partner application }
   {05}  v12:(KCDF    (26): pic_9_4_comp);       { screen function }
   {05}  v13:(KCLI    (26): pic_9_4_comp);       { length of init area  }
   {05}  v20:(EXTENT  (28): record_14);          { extent part      }
   {05}  v24:(KCDPUT  (28): record_14);          { data for DPUT call:  }
    {07} v25:(KCMOD   (28): pic_X);              { A=absolute, R=relative, Space= no time}
    {07} v26:(KCTAG   (29): pic_999);            { day     }
    {07} v27:(KCSTD   (32): pic_99);             { hour    }
    {07} v28:(KCMIN   (34): pic_99);             { minute  }
    {07} v29:(KCSEK   (36): pic_99);             { second  }
    {07       FILLER  (38): pic_X_4 }
   {05}  v30:(KCAPRO  (28): record_14);          { data for APRO call:  }
    {07} v31:(KCPI    (28): pic_X_8);            { conversation id }
    {07} v32:(KCOF    (36): pic_X);              { OSI-TP functions     }
    {07       FILLER  (37): pic_X_5  }
   {05}  v33:(KCPADM  (28): record_14);          { data for PADM call:  }
    {07} v34:(KCACT   (28): pic_XXX);            { KCOM=CS: action      }
    {07} v35:(KCADRLT (31): pic_X_8);            { KCOM=CA: LTERM name  }
    {07       FILLER  (39): pic_XXX }
   {05}  v36:(KCSGCL  (28): record_14);          { data for SIGN CL call:   }
    {07} v37:(KCLANGID(28): pic_XX);             { language id              }
    {07} v38:(KCTERRID(30): pic_XX);             { territory id             }
    {07} v39:(KCCSNAME(32): pic_X_8);            { coded character set name }
    {07       FILLER  (40): pic_XX }
   {05}  v40:(KCMCOM  (18): record_24);          { data for MCOM call:  }
    {07} v41:(KCPOS   (18): pic_X_8);            { destination in positiv case }
    {07} v42:(KCNEG   (26): pic_X_8);            { destination in negativ case }
    {07} v43:(KCCOMID (34): pic_X_8);            { complex id  }
         else: ();
                          end; { KCPAL }
 end. { Kckbl}
```

Package KCMSL

```
{*****************************************************************+**}
{*                                                                 +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                    ALL RIGHTS RESERVED                          +**}
{*                                                                 +**}
{*****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0        +**}
PACKAGE BODY Kcmsl;
   { leer }
begin
   { leer }
END.  {Kcmsl}
```

```
{*****************************************************************}
{**                                                             **}
{**  COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992   **}
{**                 ALL RIGHTS RESERVED                         **}
{**                                                             **}
{*****************************************************************}
{**  SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM ....        **}
{*****************************************************************}
{**                                                             **}
{**   Layout of UTM-messages            UTM (BS2000)  V04.0      **}
{**                                     KCMSL         16.07.1996 **}
{*****************************************************************}
PACKAGE KCMSL;
TYPE
   CHAR_001    = PACKED ARRAY [1..001] OF CHAR;
   CHAR_002    = PACKED ARRAY [1..002] OF CHAR;
   CHAR_003    = PACKED ARRAY [1..003] OF CHAR;
   CHAR_004    = PACKED ARRAY [1..004] OF CHAR;
   CHAR_005    = PACKED ARRAY [1..005] OF CHAR;
   CHAR_006    = PACKED ARRAY [1..006] OF CHAR;
   CHAR_007    = PACKED ARRAY [1..007] OF CHAR;
   CHAR_008    = PACKED ARRAY [1..008] OF CHAR;
   CHAR_009    = PACKED ARRAY [1..009] OF CHAR;
   CHAR_010    = PACKED ARRAY [1..010] OF CHAR;
   CHAR_011    = PACKED ARRAY [1..011] OF CHAR;
   CHAR_012    = PACKED ARRAY [1..012] OF CHAR;
   CHAR_013    = PACKED ARRAY [1..013] OF CHAR;
   CHAR_014    = PACKED ARRAY [1..014] OF CHAR;
   CHAR_015    = PACKED ARRAY [1..015] OF CHAR;
   CHAR_016    = PACKED ARRAY [1..016] OF CHAR;
   CHAR_017    = PACKED ARRAY [1..017] OF CHAR;
   CHAR_018    = PACKED ARRAY [1..018] OF CHAR;
   CHAR_019    = PACKED ARRAY [1..019] OF CHAR;
   CHAR_020    = PACKED ARRAY [1..020] OF CHAR;
```

```
CHAR_021     = PACKED ARRAY [1..021] OF CHAR;
CHAR_022     = PACKED ARRAY [1..022] OF CHAR;
CHAR_023     = PACKED ARRAY [1..023] OF CHAR;
CHAR_024     = PACKED ARRAY [1..024] OF CHAR;
CHAR_025     = PACKED ARRAY [1..025] OF CHAR;
CHAR_026     = PACKED ARRAY [1..026] OF CHAR;
CHAR_027     = PACKED ARRAY [1..027] OF CHAR;
CHAR_028     = PACKED ARRAY [1..028] OF CHAR;
CHAR_029     = PACKED ARRAY [1..029] OF CHAR;
CHAR_030     = PACKED ARRAY [1..030] OF CHAR;
CHAR_031     = PACKED ARRAY [1..031] OF CHAR;
CHAR_032     = PACKED ARRAY [1..032] OF CHAR;
CHAR_033     = PACKED ARRAY [1..033] OF CHAR;
CHAR_034     = PACKED ARRAY [1..034] OF CHAR;
CHAR_035     = PACKED ARRAY [1..035] OF CHAR;
CHAR_036     = PACKED ARRAY [1..036] OF CHAR;
CHAR_037     = PACKED ARRAY [1..037] OF CHAR;
CHAR_038     = PACKED ARRAY [1..038] OF CHAR;
CHAR_039     = PACKED ARRAY [1..039] OF CHAR;
CHAR_040     = PACKED ARRAY [1..040] OF CHAR;
CHAR_041     = PACKED ARRAY [1..041] OF CHAR;
CHAR_042     = PACKED ARRAY [1..042] OF CHAR;
CHAR_043     = PACKED ARRAY [1..043] OF CHAR;
CHAR_044     = PACKED ARRAY [1..044] OF CHAR;
CHAR_045     = PACKED ARRAY [1..045] OF CHAR;
CHAR_046     = PACKED ARRAY [1..046] OF CHAR;
CHAR_047     = PACKED ARRAY [1..047] OF CHAR;
CHAR_048     = PACKED ARRAY [1..048] OF CHAR;
CHAR_049     = PACKED ARRAY [1..049] OF CHAR;
CHAR_050     = PACKED ARRAY [1..050] OF CHAR;
CHAR_051     = PACKED ARRAY [1..051] OF CHAR;
CHAR_052     = PACKED ARRAY [1..052] OF CHAR;
CHAR_053     = PACKED ARRAY [1..053] OF CHAR;
CHAR_054     = PACKED ARRAY [1..054] OF CHAR;
CHAR_055     = PACKED ARRAY [1..055] OF CHAR;
CHAR_056     = PACKED ARRAY [1..056] OF CHAR;
CHAR_057     = PACKED ARRAY [1..057] OF CHAR;
CHAR_058     = PACKED ARRAY [1..058] OF CHAR;
CHAR_059     = PACKED ARRAY [1..059] OF CHAR;
CHAR_060     = PACKED ARRAY [1..060] OF CHAR;
CHAR_061     = PACKED ARRAY [1..061] OF CHAR;
CHAR_062     = PACKED ARRAY [1..062] OF CHAR;
CHAR_063     = PACKED ARRAY [1..063] OF CHAR;
CHAR_064     = PACKED ARRAY [1..064] OF CHAR;
CHAR_065     = PACKED ARRAY [1..065] OF CHAR;
CHAR_066     = PACKED ARRAY [1..066] OF CHAR;
CHAR_067     = PACKED ARRAY [1..067] OF CHAR;
CHAR_068     = PACKED ARRAY [1..068] OF CHAR;
CHAR_069     = PACKED ARRAY [1..069] OF CHAR;
```

```
CHAR_070    = PACKED ARRAY [1..070] OF CHAR;
CHAR_071    = PACKED ARRAY [1..071] OF CHAR;
CHAR_072    = PACKED ARRAY [1..072] OF CHAR;
CHAR_073    = PACKED ARRAY [1..073] OF CHAR;
CHAR_074    = PACKED ARRAY [1..074] OF CHAR;
CHAR_075    = PACKED ARRAY [1..075] OF CHAR;
CHAR_076    = PACKED ARRAY [1..076] OF CHAR;
CHAR_077    = PACKED ARRAY [1..077] OF CHAR;
CHAR_078    = PACKED ARRAY [1..078] OF CHAR;
CHAR_079    = PACKED ARRAY [1..079] OF CHAR;
CHAR_080    = PACKED ARRAY [1..080] OF CHAR;
CHAR_081    = PACKED ARRAY [1..081] OF CHAR;
CHAR_082    = PACKED ARRAY [1..082] OF CHAR;
CHAR_083    = PACKED ARRAY [1..083] OF CHAR;
CHAR_084    = PACKED ARRAY [1..084] OF CHAR;
CHAR_085    = PACKED ARRAY [1..085] OF CHAR;
CHAR_086    = PACKED ARRAY [1..086] OF CHAR;
CHAR_087    = PACKED ARRAY [1..087] OF CHAR;
CHAR_088    = PACKED ARRAY [1..088] OF CHAR;
CHAR_089    = PACKED ARRAY [1..089] OF CHAR;
CHAR_090    = PACKED ARRAY [1..090] OF CHAR;
CHAR_091    = PACKED ARRAY [1..091] OF CHAR;
CHAR_092    = PACKED ARRAY [1..092] OF CHAR;
CHAR_093    = PACKED ARRAY [1..093] OF CHAR;
CHAR_094    = PACKED ARRAY [1..094] OF CHAR;
CHAR_095    = PACKED ARRAY [1..095] OF CHAR;
CHAR_096    = PACKED ARRAY [1..096] OF CHAR;
CHAR_097    = PACKED ARRAY [1..097] OF CHAR;
CHAR_098    = PACKED ARRAY [1..098] OF CHAR;
CHAR_099    = PACKED ARRAY [1..099] OF CHAR;
CHAR_100    = PACKED ARRAY [1..100] OF CHAR;
CHAR_101    = PACKED ARRAY [1..101] OF CHAR;
CHAR_102    = PACKED ARRAY [1..102] OF CHAR;
CHAR_103    = PACKED ARRAY [1..103] OF CHAR;
CHAR_104    = PACKED ARRAY [1..104] OF CHAR;
CHAR_105    = PACKED ARRAY [1..105] OF CHAR;
CHAR_106    = PACKED ARRAY [1..106] OF CHAR;
CHAR_107    = PACKED ARRAY [1..107] OF CHAR;
CHAR_108    = PACKED ARRAY [1..108] OF CHAR;
CHAR_109    = PACKED ARRAY [1..109] OF CHAR;
CHAR_110    = PACKED ARRAY [1..110] OF CHAR;
CHAR_111    = PACKED ARRAY [1..111] OF CHAR;
CHAR_112    = PACKED ARRAY [1..112] OF CHAR;
CHAR_113    = PACKED ARRAY [1..113] OF CHAR;
CHAR_114    = PACKED ARRAY [1..114] OF CHAR;
CHAR_115    = PACKED ARRAY [1..115] OF CHAR;
CHAR_116    = PACKED ARRAY [1..116] OF CHAR;
CHAR_117    = PACKED ARRAY [1..117] OF CHAR;
CHAR_118    = PACKED ARRAY [1..118] OF CHAR;
```

```
CHAR_119    = PACKED ARRAY [1..119] OF CHAR;
CHAR_120    = PACKED ARRAY [1..120] OF CHAR;
CHAR_121    = PACKED ARRAY [1..121] OF CHAR;
CHAR_122    = PACKED ARRAY [1..122] OF CHAR;
CHAR_123    = PACKED ARRAY [1..123] OF CHAR;
CHAR_124    = PACKED ARRAY [1..124] OF CHAR;
CHAR_125    = PACKED ARRAY [1..125] OF CHAR;
CHAR_126    = PACKED ARRAY [1..126] OF CHAR;
CHAR_127    = PACKED ARRAY [1..127] OF CHAR;
CHAR_128    = PACKED ARRAY [1..128] OF CHAR;
CHAR_129    = PACKED ARRAY [1..129] OF CHAR;
CHAR_130    = PACKED ARRAY [1..130] OF CHAR;
CHAR_131    = PACKED ARRAY [1..131] OF CHAR;
CHAR_132    = PACKED ARRAY [1..132] OF CHAR;
CHAR_133    = PACKED ARRAY [1..133] OF CHAR;
CHAR_134    = PACKED ARRAY [1..134] OF CHAR;
CHAR_135    = PACKED ARRAY [1..135] OF CHAR;
CHAR_136    = PACKED ARRAY [1..136] OF CHAR;
CHAR_137    = PACKED ARRAY [1..137] OF CHAR;
CHAR_138    = PACKED ARRAY [1..138] OF CHAR;
CHAR_139    = PACKED ARRAY [1..139] OF CHAR;
CHAR_140    = PACKED ARRAY [1..140] OF CHAR;
CHAR_141    = PACKED ARRAY [1..141] OF CHAR;
CHAR_142    = PACKED ARRAY [1..142] OF CHAR;
CHAR_143    = PACKED ARRAY [1..143] OF CHAR;
CHAR_144    = PACKED ARRAY [1..144] OF CHAR;
CHAR_145    = PACKED ARRAY [1..145] OF CHAR;
CHAR_146    = PACKED ARRAY [1..146] OF CHAR;
CHAR_147    = PACKED ARRAY [1..147] OF CHAR;
CHAR_148    = PACKED ARRAY [1..148] OF CHAR;
CHAR_149    = PACKED ARRAY [1..149] OF CHAR;
CHAR_150    = PACKED ARRAY [1..150] OF CHAR;
CHAR_151    = PACKED ARRAY [1..151] OF CHAR;
CHAR_152    = PACKED ARRAY [1..152] OF CHAR;
REDEFINED   = { SIMULIERT COBOL REDEFINITIONEN                   }
              ( L1, L2, L3, L4, L5, L6, LKXXX,
                LK001,LK002,LK003,LK004,LK005,
                LK006,LK007,LK008,LK009,LK010,
                LK011,LK012,LK013,LK014,LK015,
                LK016,LK017,LK018,LK019,LK020,
                LK021,LK022,LK023,LK024,LK025,
                LK026,LK027,LK028,LK029,LK030,
                LK031,LK032,LK033,LK034,LK035,
                LK036,LK037,LK038,LK039,LK040,
                LK041,LK042,LK043,LK044,LK045,
                LK046,LK047,LK048,LK049,LK050,
                LK051,LK052,LK053,LK054,LK055,
                LK056,LK057,LK058,LK059,LK060,
                LK061,LK062,LK063,LK064,LK065,
                LK066,LK067,LK068,LK069,LK070,
                LK071,LK072,LK073,LK074,LK075,
                LK076,LK077,LK078,LK079,LK080,
                LK081,LK082,LK083,LK084,LK085,
                LK086,LK087,LK088,LK089,LK090,
                LK091,LK092,LK093,LK094,LK095,
                LK096,LK097,LK098,LK099,LK100,
                LK101,LK102,LK103,LK104,LK105,
                LK106,LK107,LK108,LK109,LK110,
                LK111,LK112,LK113,LK114,LK115,
                LK116,LK117,LK118,LK119,LK120,
                LK121,LK122,LK123,LK124,LK125,
                LK126,LK127,LK128,LK129,LK130,
                LK131,LK132,LK133,LK134,LK135,
                LK136,LK137,LK138,LK139,LK140,
                LK141,LK142,LK143,LK144,LK145,
                LK146,LK147,LK148,LK149,LK150,
                LK151,LK152,LK153,LK154,LK155,
                LK156,LK157,LK158,LK159,LK160,
                LK161,LK162,LK163,LK164,LK165,
                LK166,LK167,LK168,LK169,LK170,
                LK171,LK172,LK173,LK174,LK175,
                LK176,LK177,LK178,LK179,LK180,
                LK181,LK182,LK183,LK184,LK185,
                LK186,LK187,LK188,LK189,LK190,
                LK191,LK192,LK193,LK194,LK195,
                LK196,LK197,LK198,LK199,LK200,
                LK201,LK202,LK203,LK204,LK205,
                LK206,LK207,LK208,LK209,LK210,
                LK211,LK212,LK213,LK214,LK215,
                LK216,LK217,LK218,LK219,LK220,
                LK221,LK222,LK223,LK224,LK225,
                LK226,LK227,LK228,LK229,LK230,
                LK231,LK232,LK233,LK234,LK235,
                LK236,LK237,LK238,LK239,LK240,
                LK241,LK242,LK243,LK244,LK245,
                LK246,LK247,LK248,LK249,LK250,
                LK251,LK252,LK253,LK254,LK255,
                LP001,LP002,LP003,LP004,LP005,
                LP006,LP007,LP008,LP009,LP010,
                LP011,LP012,LP013,LP014,LP015,
                LP016,LP017,LP018,LP019);
                TYPE
MK001       = RECORD
            PTRM  :  CHAR_008;        {*PTERM NAME                *}
            PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
            BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
            LTRM  :  CHAR_008;        {*LTERM NAME                *}
            APPL  :  CHAR_008;        {*APPLICATION NAME          *}
```

```
                  MTXT  :  CHAR_112;
                              END;
      MK002       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  APPL  :  CHAR_008;        {*APPLICATION NAME           *}
                  MTXT  :  CHAR_112;
                              END;
      MK003       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  CMD   :  CHAR_008;        {*COMMAND NAME               *}
                  MTXT  :  CHAR_112;
                              END;
      MK004       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                  MTXT  :  CHAR_112;
                              END;
      MK005       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                  MTXT  :  CHAR_112;
                              END;
      MK006       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                  MTXT  :  CHAR_112;
                              END;
      MK007       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                  MTXT  :  CHAR_112;
```

```
                       END;
MK008       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
              MTXT  :  CHAR_112;
                       END;
MK009       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
              TAC   :  CHAR_008;        {*TRANSACTION CODE            *}
              MTXT  :  CHAR_104;
                       END;
MK010       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
              TAC   :  CHAR_008;        {*TRANSACTION CODE            *}
              MTXT  :  CHAR_104;
                       END;
MK011       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
              ATAC  :  CHAR_008;        {*ASYNCHRONOUS TAC            *}
              MTXT  :  CHAR_104;
                       END;
MK013       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
              CMD   :  CHAR_008;        {*COMMAND NAME                *}
              MTXT  :  CHAR_112;
                       END;
MK014       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                  *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
              LTRM  :  CHAR_008;        {*LTERM NAME                  *}
```

```
                     USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                     MTXT  :  CHAR_112;
                                 END;
          MK015        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                     TAC   :  CHAR_008;        {*TRANSACTION CODE          *}
                     FORM  :  CHAR_008;        {*FORMAT NAME (FOR K015     *}
                                               {*ONLY)                     *}
                     RCDC  :  CHAR_004;        {*KCRCDC                    *}
                     RCF2  :  CHAR_004;        {*SECONDARY FHS/VTSU RET    *}
                                               {*CODE                      *}
                     MTXT  :  CHAR_088;
                                 END;
          MK016        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                     MTXT  :  CHAR_112;
                                 END;
          MK017        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                     TCVG  :  CHAR_008;        {*CONVERSATION TAC          *}
                     RCCC  :  CHAR_003;        {*KCRCCC                    *}
                     RCDC  :  CHAR_004;        {*KCRCDC                    *}
                     RCF2  :  CHAR_004;        {*SECONDARY FHS/VTSU RET    *}
                                               {*CODE                      *}
                     TAC   :  CHAR_008;        {*TRANSACTION CODE          *}
                     MTXT  :  CHAR_085;
                                 END;
          MK018        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     APPL  :  CHAR_008;        {*APPLICATION NAME          *}
                     MTXT  :  CHAR_112;
                                 END;
          MK019        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     APPL  :  CHAR_008;          {*APPLICATION NAME            *}
                     MTXT  :  CHAR_112;
                                END;
        MK020        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     USER  :  CHAR_008;          {*USER/LSES/OSI-ASS NAME      *}
                     MTXT  :  CHAR_112;
                                END;
        MK021        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     MTXT  :  CHAR_120;
                                END;
        MK022        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     MTXT  :  CHAR_120;
                                END;
        MK023        = RECORD
                     OMSG  :  CHAR_074;          {*BROADCAST MESSAGE           *}
                     MTXT  :  CHAR_078;
                                END;
        MK024        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     USER  :  CHAR_008;          {*USER/LSES/OSI-ASS NAME      *}
                     MTXT  :  CHAR_112;
                                END;
        MK025        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
                     PRNM  :  CHAR_008;          {*PROCESSOR NAME              *}
                     BCAP  :  CHAR_008;          {*BCAM APPLICATION NAME       *}
                     LTRM  :  CHAR_008;          {*LTERM NAME                  *}
                     MTXT  :  CHAR_120;
                                END;
        MK026        = RECORD
                     PTRM  :  CHAR_008;          {*PTERM NAME                  *}
```

```
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LTRM  :  CHAR_008;        {*LTERM NAME                  *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
                MTXT  :  CHAR_112;
                            END;
MK027           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                  *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LTRM  :  CHAR_008;        {*LTERM NAME                  *}
                MTXT  :  CHAR_120;
                            END;
MK029           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                  *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LTRM  :  CHAR_008;        {*LTERM NAME                  *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
                MTXT  :  CHAR_112;
                            END;
MK030           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                  *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LTRM  :  CHAR_008;        {*LTERM NAME                  *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
                MTXT  :  CHAR_112;
                            END;
MK031           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                  *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LTRM  :  CHAR_008;        {*LTERM NAME                  *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
                MTXT  :  CHAR_112;
                            END;
MK032           = RECORD
                CON   :  CHAR_008;        {*CONNECTION NAME             *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
                LPAP  :  CHAR_008;        {*LPAP NAME                   *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME      *}
                RCF1  :  CHAR_003;        {*RETURN CODE 1               *}
                RCF2  :  CHAR_004;        {*RETURN CODE 2               *}
                MTXT  :  CHAR_105;
                            END;
MK033           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                  *}
```

```
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                     REST  :  CHAR_001;        {*RESTART INDICATOR OF      *}
                                               {*LTERM                     *}
                     MTXT  :  CHAR_111;
                                 END;
        MK036        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     RSLT  :  CHAR_001;        {*RESULT                    *}
                     REAS  :  CHAR_001;        {*REASON                    *}
                     MTXT  :  CHAR_118;
                                 END;
        MK040        = RECORD
                     WLEV  :  CHAR_001;        {*WARN LEVEL OF PAGE POOL   *}
                     MTXT  :  CHAR_151;
                                 END;
        MK041        = RECORD
                     WLEV  :  CHAR_001;        {*WARN LEVEL OF PAGE POOL   *}
                     MTXT  :  CHAR_151;
                                 END;
        MK043        = RECORD
                     DMSE  :  CHAR_004;        {*DMS ERROR CODE            *}
                     FNAM  :  CHAR_054;        {*FILE NAME                 *}
                     MTXT  :  CHAR_094;
                                 END;
        MK045        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     PALT  :  CHAR_008;        {*LTERM NAME PRINT ADMIN    *}
                                               {*STATION                   *}
                     CID   :  CHAR_008;        {*PRINTER CONTROL ID        *}
                     MTXT  :  CHAR_104;
                                 END;
        MK046        = RECORD
                     PTRM  :  CHAR_008;        {*PTERM NAME                *}
                     PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                     BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                     LTRM  :  CHAR_008;        {*LTERM NAME                *}
                     PALT  :  CHAR_008;        {*LTERM NAME PRINT ADMIN    *}
                                               {*STATION                   *}
                     CID   :  CHAR_008;        {*PRINTER CONTROL ID        *}
                     DPID  :  CHAR_008;        {*ASYNCHRONOUS MESSAGE ID   *}
```

```
                ERPR  :  CHAR_001;        {*PRINT ERROR CODE           *}
                IMSG  :  CHAR_032;        {*FIRST PART OF INPUT        *}
                                          {*MESSAGE                    *}
                MTXT  :  CHAR_063;
                            END;
   MK049        = RECORD
                RCCC  :  CHAR_004;        {*STARTUP ERROR CODE         *}
                MTXT  :  CHAR_148;
                            END;
   MK050        = RECORD
                APPL  :  CHAR_008;        {*APPLICATION NAME           *}
                VERS  :  CHAR_008;        {*UTM VERSION                *}
                MTXT  :  CHAR_136;
                            END;
   MK051        = RECORD
                APPL  :  CHAR_008;        {*APPLICATION NAME           *}
                VERS  :  CHAR_008;        {*UTM VERSION                *}
                MTXT  :  CHAR_136;
                            END;
   MK052        = RECORD
                TASK  :  CHAR_004;        {*TSN OF UTM TASK            *}
                APPL  :  CHAR_008;        {*APPLICATION NAME           *}
                PRGV  :  CHAR_004;        {*PROGRAM VERSION IN CASE    *}
                                          {*OF PROGRAM EXCHANGE        *}
                MTXT  :  CHAR_136;
                            END;
   MK053        = RECORD
                CNTR  :  CHAR_006;        {*NUMBER OF LPUT RECORDS     *}
                MTXT  :  CHAR_146;
                            END;
   MK055        = RECORD
                ATAC  :  CHAR_008;        {*ASYNCHRONOUS TAC           *}
                RCCC  :  CHAR_003;        {*KCRCCC                     *}
                RCDC  :  CHAR_004;        {*KCRCDC                     *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                MTXT  :  CHAR_121;
                            END;
   MK056        = RECORD
                TASK  :  CHAR_004;        {*TSN OF UTM TASK            *}
                MTXT  :  CHAR_148;
                            END;
   MK058        = RECORD
                TASK  :  CHAR_004;        {*TSN OF UTM TASK            *}
                MTXT  :  CHAR_148;
                            END;
   MK060        = RECORD
                TRMA  :  CHAR_006;        {*TERM APPLICATION REASON    *}
                MTXT  :  CHAR_146;
```

```
                        END;
MK061        = RECORD
               FNAM  :  CHAR_054;        {*FILE NAME                   *}
               MTXT  :  CHAR_098;
                        END;
MK063        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                  *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
               LTRM  :  CHAR_008;        {*LTERM NAME                  *}
               FMTN  :  CHAR_008;        {*FORMAT NAME                 *}
               RCF1  :  CHAR_004;        {*KCRCDC                      *}
               RCF2  :  CHAR_004;        {*SECONDARY FHS/VTSU RET      *}
                                         {*CODE                        *}
               MTXT  :  CHAR_104;
                        END;
MK064        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                  *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
               LTRM  :  CHAR_008;        {*LTERM NAME                  *}
               DEVC  :  CHAR_001;        {*DEVICE TYPE                 *}
               FIL1  :  CHAR_001;        {*APPLICATION STATE           *}
               FIL2  :  CHAR_001;        {*LTERM STATE                 *}
               FIL3  :  CHAR_002;        {*PTERM STATE                 *}
               VTRC  :  CHAR_004;        {*VTSU OR ASECO RETURN CODE *}
               IMSG  :  CHAR_032;        {*FIRST PART OF INPUT         *}
                                         {*MESSAGE                     *}
               REAS  :  CHAR_001;        {*REASON                      *}
               CBRC  :  CHAR_004;        {*VTSUCB RETURN CODE          *}
               MTXT  :  CHAR_074;
                        END;
MK065        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                  *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
               LTRM  :  CHAR_008;        {*LTERM NAME                  *}
               FIL1  :  CHAR_001;        {*BCAM REQUEST OR ANNO TYPE *}
                                         {*/ UTM ANNO TYPE             *}
               FIL2  :  CHAR_004;        {*BCAM INFOWORD               *}
               MTXT  :  CHAR_115;
                        END;
MK069        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                  *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME              *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME       *}
               LTRM  :  CHAR_008;        {*LTERM NAME                  *}
               COTM  :  CHAR_004;        {*ELAPSED CONNECTION TIME     *}
                                         {*IN SECONDS                  *}
```

```
                  REAS  :  CHAR_001;        {*DIAGNOSTIC INFORMATION    *}
                                            {*(DISCONNECT REASON)       *}
                  REA6  :  CHAR_001;        {*DIAGNOSTIC INFORMATION    *}
                                            {*(DISCONNECT USER REASON)  *}
                  MTXT  :  CHAR_114;
                              END;
      MK070       = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                  COTM  :  CHAR_004;        {*ELAPSED CONNECTION TIME   *}
                                            {*IN SECONDS                *}
                  CPTM  :  CHAR_004;        {*CPU TIME SINCE SIGN-ON IN *}
                                            {*MILLISECONDS              *}
                  MTXT  :  CHAR_104;
                              END;
      MK072       = RECORD
                  STMT  :  CHAR_011;        {*STATEMENT OF KDCDEF       *}
                  MTXT  :  CHAR_141;
                              END;
      MK073       = RECORD
                  ATTR  :  CHAR_011;        {*ATTRIBUT OF               *}
                                            {*LOAD-MODULE/PROGRAM       *}
                  STMT  :  CHAR_011;        {*STATEMENT OF KDCDEF       *}
                  PROG  :  CHAR_032;        {*PROGRAM OR LOAD MODULE    *}
                                            {*NAME                      *}
                  MTXT  :  CHAR_098;
                              END;
      MK074       = RECORD
                  CTYP  :  CHAR_004;        {*TYPE OF PROGRAM EXCHANGE  *}
                  PROG  :  CHAR_032;        {*PROGRAM OR LOAD MODULE    *}
                                            {*NAME                      *}
                  PVER  :  CHAR_024;        {*PROGRAM VERSION           *}
                  MTXT  :  CHAR_092;
                              END;
      MK075       = RECORD
                  CTYP  :  CHAR_004;        {*TYPE OF PROGRAM EXCHANGE  *}
                  PROG  :  CHAR_032;        {*PROGRAM OR LOAD MODULE    *}
                                            {*NAME                      *}
                  PVER  :  CHAR_024;        {*PROGRAM VERSION           *}
                  MTXT  :  CHAR_092;
                              END;
      MK076       = RECORD
                  RCCC  :  CHAR_003;        {*KCRCCC                    *}
                  RCDC  :  CHAR_004;        {*KCRCDC                    *}
                  ADTC  :  CHAR_008;        {*ADMINISTRATION TAC        *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
```

```
            LTRM  :  CHAR_008;        {*LTERM NAME                 *}
            MTXT  :  CHAR_121;
                        END;
MK079       = RECORD
            REAS  :  CHAR_002;        {*REASON                     *}
            MTXT  :  CHAR_150;
                        END;
MK081       = RECORD
            IMSG  :  CHAR_005;        {*NUMBER OF TERMINAL INPUT  *}
                                      {*MESSAGES                   *}
            OMSG  :  CHAR_005;        {*NUMBER OF TERMINAL OUTPUT *}
                                      {*MESSAGES                   *}
            CONU  :  CHAR_005;        {*NUMBER OF CONNECTED USERS *}
            ATAC  :  CHAR_005;        {*NUMBER OF UNPROCESSED     *}
                                      {*ASYNCHRONOUS TACS          *}
            LWRT  :  CHAR_005;        {*NUMBER OF USLOG FILE      *}
                                      {*WRITES                     *}
            HITR  :  CHAR_003;        {*CACHE HIT RATE             *}
            WTBF  :  CHAR_003;        {*CACHE WAITS FOR BUFFER    *}
            MTXT  :  CHAR_121;
                        END;
MK086       = RECORD
            PTRM  :  CHAR_008;        {*PTERM NAME                 *}
            PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
            BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
            LTRM  :  CHAR_008;        {*LTERM NAME                 *}
            USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
            SYSD  :  CHAR_002;        {*SYSTEM SENSE DATA          *}
            USSD  :  CHAR_002;        {*USER SENSE DATA            *}
            FMH7  :  CHAR_080;        {*ERROR RECOVERY PROCEDURE  *}
                                      {*MESSAGE                    *}
            AGUS  :  CHAR_008;        {*JOB-SUBMITTING USER       *}
            MTXT  :  CHAR_020;
                        END;
MK088       = RECORD
            LSES  :  CHAR_008;        {*LSES NAME                  *}
            RSES  :  CHAR_008;        {*RSES NAME                  *}
            LPAP  :  CHAR_008;        {*LPAP NAME                  *}
            SRFG  :  CHAR_004;        {*SAVED SESSION STATE       *}
            PSQN  :  CHAR_004;        {*SAVED PET SEQUENCE NUMBER *}
            ESQS  :  CHAR_004;        {*SAVED SEQUENCE NUMBER     *}
            EBSS  :  CHAR_004;        {*SAVED BRACKET STATE       *}
            ESQR  :  CHAR_005;        {*ACTUAL REQUEST SEQUENCE   *}
                                      {*NUMBER                     *}
            ESRR  :  CHAR_005;        {*ACTUAL RESPONSE SEQUENCE  *}
                                      {*NUMBER                     *}
            EBSR  :  CHAR_004;        {*ACTUAL BRACKET STATE      *}
            MTXT  :  CHAR_098;
                        END;
```

```
MK089        = RECORD
               GNDA  :  CHAR_003;        {*GENERATION DATE            *}
                                         {*ASYNCHRONOUS MESSAGE       *}
               GNTI  :  CHAR_008;        {*GENERATION TIME            *}
                                         {*ASYNCHRONOUS MESSAGE       *}
               DEST  :  CHAR_008;        {*DESTINATION OF             *}
                                         {*ASYNCHRONOUS MSG           *}
               GNUS  :  CHAR_008;        {*USER NAME OF ASYNCHRON.    *}
                                         {*MESSAGE GENERATION         *}
               USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
               DLDA  :  CHAR_003;        {*DAY OF KDCS CALL PADM      *}
                                         {*DL/DA                      *}
               DLTI  :  CHAR_008;        {*TIME OF KDCS CALL PADM     *}
                                         {*DL/DA                      *}
               CHAI  :  CHAR_003;        {*CHAINED MESSAGE            *}
                                         {*INFORMATION                *}
               MTXT  :  CHAR_103;
                           END;
MK090        = RECORD
               DEST  :  CHAR_008;        {*DESTINATION OF             *}
                                         {*ASYNCHRONOUS MSG           *}
               USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
               DLDA  :  CHAR_003;        {*DAY OF KDCS CALL PADM      *}
                                         {*DL/DA                      *}
               DLTI  :  CHAR_008;        {*TIME OF KDCS CALL PADM     *}
                                         {*DL/DA                      *}
               MTXT  :  CHAR_125;
                           END;
MK091        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                 *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
               LTRM  :  CHAR_008;        {*LTERM NAME                 *}
               USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
               ASRC  :  CHAR_004;        {*ASECO RETURN CODE (CHIP    *}
                                         {*CARD MODULE)               *}
               MTXT  :  CHAR_108;
                           END;
MK092        = RECORD
               PTRM  :  CHAR_008;        {*PTERM NAME                 *}
               PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
               BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
               LTRM  :  CHAR_008;        {*LTERM NAME                 *}
               USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
               PAS1  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
               PAS2  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
               PAS3  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
               MTXT  :  CHAR_052;
                           END;
```

```
MK093       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                   *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME               *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME        *}
              LTRM  :  CHAR_008;        {*LTERM NAME                   *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME       *}
              HSTA  :  CHAR_002;        {*HEIGHT OF STACK              *}
              MSTA  :  CHAR_002;        {*MAXIMUM STACK HEIGHT         *}
              MTXT  :  CHAR_108;
                          END;
MK094       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                   *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME               *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME        *}
              LTRM  :  CHAR_008;        {*LTERM NAME                   *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME       *}
              RCF1  :  CHAR_003;        {*RETURN CODE 1                *}
              MTXT  :  CHAR_109;
                          END;
MK097       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                   *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME               *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME        *}
              LTRM  :  CHAR_008;        {*LTERM NAME                   *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME       *}
              MTXT  :  CHAR_112;
                          END;
MK098       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                   *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME               *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME        *}
              LTRM  :  CHAR_008;        {*LTERM NAME                   *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME       *}
              RCF1  :  CHAR_004;        {*RETURN CODE 1                *}
              RCF2  :  CHAR_004;        {*RETURN CODE 2                *}
              MTXT  :  CHAR_104;
                          END;
MK101       = RECORD
              PTRM  :  CHAR_008;        {*PTERM NAME                   *}
              PRNM  :  CHAR_008;        {*PROCESSOR NAME               *}
              BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME        *}
              LTRM  :  CHAR_008;        {*LTERM NAME                   *}
              USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME       *}
              MTXT  :  CHAR_112;
                          END;
MK104       = RECORD
              UTMD  :  CHAR_007;        {*UTM-D EVENT                  *}
              LSES  :  CHAR_008;        {*LSES NAME                    *}
              LPAP  :  CHAR_008;        {*LPAP NAME                    *}
```

```
                AGUS  :  CHAR_008;        {*JOB-SUBMITTING USER       *}
                OCVS  :  CHAR_001;        {*OLD CONVERSATION STATE    *}
                OTAS  :  CHAR_001;        {*OLD TRANSACTION STATE     *}
                ACTI  :  CHAR_006;        {*SYSTEM ACTION             *}
                NCVS  :  CHAR_001;        {*NEW CONVERSATION STATE    *}
                NTAS  :  CHAR_001;        {*NEW TRANSACTION STATE     *}
                MTXT  :  CHAR_111;
                            END;
MK105           = RECORD
                LSES  :  CHAR_008;        {*LSES NAME                 *}
                LPAP  :  CHAR_008;        {*LPAP NAME                 *}
                AGUS  :  CHAR_008;        {*JOB-SUBMITTING USER       *}
                SYST  :  CHAR_004;        {*SYSTEM                    *}
                MTXT  :  CHAR_124;
                            END;
MK106           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                LTRM  :  CHAR_008;        {*LTERM NAME                *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                DEVC  :  CHAR_001;        {*DEVICE TYPE               *}
                FIL1  :  CHAR_001;        {*APPLICATION STATE         *}
                FIL2  :  CHAR_001;        {*LTERM STATE               *}
                FIL3  :  CHAR_002;        {*PTERM STATE               *}
                VTRC  :  CHAR_004;        {*VTSU OR ASECO RETURN CODE *}
                CBRC  :  CHAR_004;        {*VTSUCB RETURN CODE        *}
                OMSG  :  CHAR_032;        {*FIRST PART OF OUTPUT      *}
                                          {*MESSAGE                   *}
                FMTN  :  CHAR_008;        {*FORMAT NAME               *}
                CCSN  :  CHAR_008;        {*CCSNAME                   *}
                MTXT  :  CHAR_051;
                            END;
MK107           = RECORD
                TTYP  :  CHAR_008;        {*TERMINAL TYPE             *}
                MTXT  :  CHAR_144;
                            END;
MK108           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                *}
                PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                LTRM  :  CHAR_008;        {*LTERM NAME                *}
                USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                ASRC  :  CHAR_004;        {*ASECO RETURN CODE (CHIP   *}
                                          {*CARD MODULE)              *}
                MTXT  :  CHAR_108;
                            END;
MK109           = RECORD
                PTRM  :  CHAR_008;        {*PTERM NAME                *}
```

```
                      PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                      BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                      LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                      USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                      ASRC  :  CHAR_004;        {*ASECO RETURN CODE (CHIP    *}
                                                {*CARD MODULE)               *}
                      ADFN  :  CHAR_016;        {*ADF NAME                   *}
                      MTXT  :  CHAR_092;
                                  END;
         MK115        = RECORD
                      PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                      PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                      BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                      LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                      SNPT  :  CHAR_008;        {*MUX SESSION PTERM NAME     *}
                      SNPR  :  CHAR_008;        {*MUX SESSION PROCESSOR      *}
                                                {*NAME                       *}
                      SNLT  :  CHAR_008;        {*MUX SESSION LTERM NAME     *}
                      CCC   :  CHAR_001;        {*CONTXT MACRO: CONDITION    *}
                                                {*CODE IN PCR FORMAT         *}
                      REAS  :  CHAR_001;        {*REASON                     *}
                      ANNO  :  CHAR_032;        {*ANNO RECEIVED              *}
                      MTXT  :  CHAR_062;
                                  END;
         MK116        = RECORD
                      PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                      PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                      BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                      LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                      SNPT  :  CHAR_008;        {*MUX SESSION PTERM NAME     *}
                      SNPR  :  CHAR_008;        {*MUX SESSION PROCESSOR      *}
                                                {*NAME                       *}
                      SNLT  :  CHAR_008;        {*MUX SESSION LTERM NAME     *}
                      USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                      REAS  :  CHAR_001;        {*REASON                     *}
                      MTXT  :  CHAR_087;
                                  END;
         MK117        = RECORD
                      PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                      PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                      BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                      LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                      SNPT  :  CHAR_008;        {*MUX SESSION PTERM NAME     *}
                      SNPR  :  CHAR_008;        {*MUX SESSION PROCESSOR      *}
                                                {*NAME                       *}
                      SNLT  :  CHAR_008;        {*MUX SESSION LTERM NAME     *}
                      USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
                      REAS  :  CHAR_001;        {*REASON                     *}
                      MTXT  :  CHAR_087;
```

```
                            END;
MK119        = RECORD
             OSLP  :  CHAR_008;        {*OSI-LPAP NAME              *}
             USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
             TAC   :  CHAR_008;        {*TRANSACTION CODE           *}
             DIA1  :  CHAR_004;        {*DIAGNOSTIC INFORMATION     *}
             DIA2  :  CHAR_004;        {*DIAGNOSTIC INFORMATION     *}
             DIA3  :  CHAR_004;        {*DIAGNOSTIC INFORMATION     *}
             MTXT  :  CHAR_116;
                            END;
MK120        = RECORD
             PTRM  :  CHAR_008;        {*PTERM NAME                 *}
             PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
             BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
             LTRM  :  CHAR_008;        {*LTERM NAME                 *}
             USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
             MTXT  :  CHAR_112;
                            END;
MK121        = RECORD
             PTRM  :  CHAR_008;        {*PTERM NAME                 *}
             PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
             BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
             LTRM  :  CHAR_008;        {*LTERM NAME                 *}
             USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
             PAS1  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
             PAS2  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
             PAS3  :  CHAR_020;        {*SPACE FOR PASSWORD         *}
             NUMD  :  CHAR_002;        {*NUMBER DAYS PASSWORD       *}
                                       {*VALID                      *}
             MTXT  :  CHAR_050;
                            END;
MK123        = RECORD
             LTRM  :  CHAR_008;        {*LTERM NAME                 *}
             TAC   :  CHAR_008;        {*TRANSACTION CODE           *}
             USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME     *}
             MTXT  :  CHAR_128;
                            END;
MK124        = RECORD
             RCXA  :  CHAR_004;        {*RETURNCODE XAP-TP          *}
                                       {*STARTFUNCTIONS             *}
             PHAX  :  CHAR_014;        {*INIT or START/RESTART of   *}
                                       {*XAP-TP                     *}
             MTXT  :  CHAR_134;
                            END;
MK125        = RECORD
             PTRM  :  CHAR_008;        {*PTERM NAME                 *}
             PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
             BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
             LTRM  :  CHAR_008;        {*LTERM NAME                 *}
```

```
                        USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                        MTXT  :  CHAR_112;
                                    END;
            MK126       = RECORD
                        SATR  :  CHAR_004;        {*SAT RETURNCODE            *}
                        MTXT  :  CHAR_148;
                                    END;
            MK128       = RECORD
                        CON   :  CHAR_008;        {*CONNECTION NAME           *}
                        PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                        BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                        LPAP  :  CHAR_008;        {*LPAP NAME                 *}
                        LSES  :  CHAR_008;        {*LSES NAME                 *}
                        REAS  :  CHAR_001;        {*REASON                    *}
                        RCDC  :  CHAR_004;        {*KCRCDC                    *}
                        TAC   :  CHAR_008;        {*TRANSACTION CODE          *}
                        MTXT  :  CHAR_099;
                                    END;
            MK130       = RECORD
                        TPRI  :  CHAR_001;        {*EXTERNAL TASK-PRIORITY    *}
                        TASK  :  CHAR_004;        {*TSN OF UTM TASK           *}
                        MTXT  :  CHAR_147;
                                    END;
            MK135       = RECORD
                        PTRM  :  CHAR_008;        {*PTERM NAME                *}
                        PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                        BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                        LTRM  :  CHAR_008;        {*LTERM NAME                *}
                        UPCR  :  CHAR_001;        {*UPIC ERROR REASON         *}
                        UPCS  :  CHAR_002;        {*USRTNSR UPIC STATE        *}
                        UPCP  :  CHAR_004;        {*UPIC PROTOCOLL            *}
                        MTXT  :  CHAR_113;
                                    END;
            MK137       = RECORD
                        FNAM  :  CHAR_054;        {*FILE NAME                 *}
                        MTXT  :  CHAR_098;
                                    END;
            MK138       = RECORD
                        FNAM  :  CHAR_054;        {*FILE NAME                 *}
                        MTXT  :  CHAR_098;
                                    END;
            MK139       = RECORD
                        FNAM  :  CHAR_054;        {*FILE NAME                 *}
                        MTXT  :  CHAR_098;
                                    END;
            MK140       = RECORD
                        PTRM  :  CHAR_008;        {*PTERM NAME                *}
                        PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                        BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
```

```
                    LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                    MXP1  :  CHAR_004;        {*MUX PROTOCOLLVERSION       *}
                                              {*(LOWER BOUNDARY)           *}
                    MXP2  :  CHAR_004;        {*MUX PROTOCOLLVERSION       *}
                                              {*(UPPER BOUNDARY)           *}
                    MTXT  :  CHAR_112;
                                END;
        MK141       = RECORD
                    PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                    PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                    BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                    LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                    MXP1  :  CHAR_004;        {*MUX PROTOCOLLVERSION       *}
                                              {*(LOWER BOUNDARY)           *}
                    MTXT  :  CHAR_116;
                                END;
        MK142       = RECORD
                    PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                    PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                    BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                    LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                    MXPT  :  CHAR_008;        {*MUX PTERM                  *}
                    MXPR  :  CHAR_008;        {*MUX PROCESSOR              *}
                    MXLT  :  CHAR_008;        {*MUX LTERM                  *}
                    MTXT  :  CHAR_096;
                                END;
        MK143       = RECORD
                    PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                    PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                    BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                    LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                    STS1  :  CHAR_002;        {*STSN-REQ SEQUENCE NUMBER   *}
                                              {*RCV-CNT                    *}
                    STS2  :  CHAR_002;        {*STSN-REQ SEQUENCE NUMBER   *}
                                              {*SEND-CNT                   *}
                    STS3  :  CHAR_002;        {*STSN-RSP SEQUENCE NUMBER   *}
                                              {*SLU-PLU                    *}
                    STS4  :  CHAR_002;        {*STSN-RSP SEQUENCE NUMBER   *}
                                              {*PLU-SLU                    *}
                    MTXT  :  CHAR_112;
                                END;
        MK144       = RECORD
                    PTRM  :  CHAR_008;        {*PTERM NAME                 *}
                    PRNM  :  CHAR_008;        {*PROCESSOR NAME             *}
                    BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME      *}
                    LTRM  :  CHAR_008;        {*LTERM NAME                 *}
                    DEVC  :  CHAR_001;        {*DEVICE TYPE                *}
                    FIL1  :  CHAR_001;        {*APPLICATION STATE          *}
                    FIL2  :  CHAR_001;        {*LTERM STATE                *}
```

```
                  FIL3  :  CHAR_002;        {*PTERM STATE               *}
                  VTRC  :  CHAR_004;        {*VTSU OR ASECO RETURN CODE *}
                  CBRC  :  CHAR_004;        {*VTSUCB RETURN CODE        *}
                  OMSG  :  CHAR_032;        {*FIRST PART OF OUTPUT      *}
                                            {*MESSAGE                   *}
                  FMTN  :  CHAR_008;        {*FORMAT NAME               *}
                  CCSN  :  CHAR_008;        {*CCSNAME                   *}
                  MTXT  :  CHAR_059;
                              END;
      MK145         = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                  MTXT  :  CHAR_112;
                              END;
      MK146         = RECORD
                  BCMO  :  CHAR_004;        {*BCMM-OPCODE               *}
                  BCMR  :  CHAR_004;        {*BCMM-RETURNCODE           *}
                  STDH  :  CHAR_008;        {*BS2000 STANDARDHEADER     *}
                  TASK  :  CHAR_004;        {*TSN OF UTM TASK           *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                  MTXT  :  CHAR_124;
                              END;
      MK147         = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                *}
                  USER  :  CHAR_008;        {*USER/LSES/OSI-ASS NAME    *}
                  MTXT  :  CHAR_112;
                              END;
      MK150         = RECORD
                  PTRM  :  CHAR_008;        {*PTERM NAME                *}
                  PRNM  :  CHAR_008;        {*PROCESSOR NAME            *}
                  BCAP  :  CHAR_008;        {*BCAM APPLICATION NAME     *}
                  LTRM  :  CHAR_008;        {*LTERM NAME                *}
                  RSOA  :  CHAR_032;        {*RSO ANNO                  *}
                  RSOO  :  CHAR_001;        {*RSO ACTION                *}
                  RSOM  :  CHAR_007;        {*RSO ERROR MESSAGE         *}
                  RSOR  :  CHAR_004;        {*RSO RETURNCODE            *}
                  RSO2  :  CHAR_004;        {*RSO ASYN RETURNCODE       *}
                  MTXT  :  CHAR_072;
                              END;
      MK151         = RECORD
                  IDEF  :  CHAR_008;        {*RETURNCODE OF INVERSE     *}
                                            {*KDCDEF                    *}
                  DMSE  :  CHAR_004;        {*DMS ERROR CODE            *}
```

```
                FNAM  :  CHAR_054;         {*FILE NAME                *}
                MTXT  :  CHAR_086;
                            END;
MK152           = RECORD
                COND  :  CHAR_003;         {*CONDITION                *}
                MTYP  :  CHAR_004;         {*MESSAGE TYPE             *}
                OSLP  :  CHAR_008;         {*OSI-LPAP NAME            *}
                USER  :  CHAR_008;         {*USER/LSES/OSI-ASS NAME   *}
                LTAC  :  CHAR_008;         {*TAC OR LTAC              *}
                AAIS  :  CHAR_004;         {*ATOMIC ACTION IDENTIFIER *}
                                           {*SIZE                     *}
                AAID  :  CHAR_064;         {*ATOMIC ACTION IDENTIFIER *}
                MTXT  :  CHAR_053;
                            END;
MP001           = RECORD
                XPFU  :  CHAR_020;         {*CALLED OSI-TP FUNCTION   *}
                XPRE  :  CHAR_004;         {*OSI-TP RETURN CODE       *}
                XPER  :  CHAR_004;         {*OSI-TP ERROR CODE        *}
                XP1I  :  CHAR_004;         {*OSI-TP ADDITIONAL        *}
                                           {*INFORMATION 1            *}
                XP2I  :  CHAR_004;         {*OSI-TP ADDITIONAL        *}
                                           {*INFORMATION 2            *}
                XPCO  :  CHAR_004;         {*MESSAGE CORRELATOR NUMBER *}
                MTXT  :  CHAR_112;
                            END;
MP002           = RECORD
                XPFU  :  CHAR_020;         {*CALLED OSI-TP FUNCTION   *}
                ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                OSLP  :  CHAR_008;         {*OSI-LPAP NAME            *}
                XPRE  :  CHAR_004;         {*OSI-TP RETURN CODE       *}
                XPER  :  CHAR_004;         {*OSI-TP ERROR CODE        *}
                XP1I  :  CHAR_004;         {*OSI-TP ADDITIONAL        *}
                                           {*INFORMATION 1            *}
                XP2I  :  CHAR_004;         {*OSI-TP ADDITIONAL        *}
                                           {*INFORMATION 2            *}
                XPCO  :  CHAR_004;         {*MESSAGE CORRELATOR NUMBER *}
                MTXT  :  CHAR_096;
                            END;
MP003           = RECORD
                ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                XPRJ  :  CHAR_004;         {*OSI-TP ASSOCIATION REASON *}
                                           {*FOR REJECT               *}
                XPLT  :  CHAR_004;         {*OSI-TP INVALID LENGTH    *}
                MTXT  :  CHAR_136;
                            END;
MP004           = RECORD
                ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                OSLP  :  CHAR_008;         {*OSI-LPAP NAME            *}
                XPRJ  :  CHAR_004;         {*OSI-TP ASSOCIATION REASON *}
```

```
                                             {*FOR REJECT               *}
                  MTXT  :  CHAR_132;
                              END;
      MP005       = RECORD
                  ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                  XPNS  :  CHAR_008;         {*OSI-TP N-SEL OF PARTNER  *}
                  XPTS  :  CHAR_008;         {*OSI-TP T-SEL OF PARTNER  *}
                  XPLS  :  CHAR_004;         {*OSI-TP LENGTH S-SEL OF   *}
                                             {*PARTNER                  *}
                  XPCS  :  CHAR_016;         {*OSI-TP S-SEL OF PARTNER  *}
                                             {*(CHAR)                   *}
                  XPHS  :  CHAR_016;         {*OSI-TP S-SEL OF PARTNER  *}
                                             {*(HEX)                    *}
                  XPLP  :  CHAR_004;         {*OSI-TP LENGTH P-SEL OF   *}
                                             {*PARTNER                  *}
                  XPCP  :  CHAR_016;         {*OSI-TP P-SEL OF PARTNER  *}
                                             {*(CHAR)                   *}
                  XPHP  :  CHAR_016;         {*OSI-TP P-SEL OF PARTNER  *}
                                             {*(HEX)                    *}
                  MTXT  :  CHAR_056;
                              END;
      MP006       = RECORD
                  ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                  OSLP  :  CHAR_008;         {*OSI-LPAP NAME            *}
                  XP00  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*0                        *}
                  XP10  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*1                        *}
                  XP20  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*2                        *}
                  XP30  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*3                        *}
                  XP40  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*4                        *}
                  XP50  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*5                        *}
                  XP60  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*6                        *}
                  XP70  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*7                        *}
                  XP80  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*8                        *}
                  XP90  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER *}
                                             {*9                        *}
                  MTXT  :  CHAR_096;
                              END;
      MP007       = RECORD
                  ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME        *}
                  OSLP  :  CHAR_008;         {*OSI-LPAP NAME            *}
```

```
                        XPRE  :  CHAR_004;        {*OSI-TP RETURN CODE        *}
                        XPER  :  CHAR_004;        {*OSI-TP ERROR CODE         *}
                        XP1I  :  CHAR_004;        {*OSI-TP ADDITIONAL         *}
                                                  {*INFORMATION 1             *}
                        XP2I  :  CHAR_004;        {*OSI-TP ADDITIONAL         *}
                                                  {*INFORMATION 2             *}
                        XPCO  :  CHAR_004;        {*MESSAGE CORRELATOR NUMBER *}
                        MTXT  :  CHAR_116;
                                    END;
            MP008       = RECORD
                        ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME         *}
                        OSLP  :  CHAR_008;        {*OSI-LPAP NAME             *}
                        XPOS  :  CHAR_004;        {*OSI-TP ASSOCIATION        *}
                                                  {*REFERENCE                 *}
                        MTXT  :  CHAR_132;
                                    END;
            MP009       = RECORD
                        ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME         *}
                        OSLP  :  CHAR_008;        {*OSI-LPAP NAME             *}
                        XPRJ  :  CHAR_004;        {*OSI-TP ASSOCIATION REASON *}
                                                  {*FOR REJECT                *}
                        XPLT  :  CHAR_004;        {*OSI-TP INVALID LENGTH     *}
                        XPOS  :  CHAR_004;        {*OSI-TP ASSOCIATION        *}
                                                  {*REFERENCE                 *}
                        MTXT  :  CHAR_124;
                                    END;
            MP010       = RECORD
                        ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME         *}
                        OSLP  :  CHAR_008;        {*OSI-LPAP NAME             *}
                        XPNS  :  CHAR_008;        {*OSI-TP N-SEL OF PARTNER   *}
                        XPTS  :  CHAR_008;        {*OSI-TP T-SEL OF PARTNER   *}
                        XPLS  :  CHAR_004;        {*OSI-TP LENGTH S-SEL OF    *}
                                                  {*PARTNER                   *}
                        XPCS  :  CHAR_016;        {*OSI-TP S-SEL OF PARTNER   *}
                                                  {*(CHAR)                    *}
                        XPHS  :  CHAR_016;        {*OSI-TP S-SEL OF PARTNER   *}
                                                  {*(HEX)                     *}
                        XPLP  :  CHAR_004;        {*OSI-TP LENGTH P-SEL OF    *}
                                                  {*PARTNER                   *}
                        XPCP  :  CHAR_016;        {*OSI-TP P-SEL OF PARTNER   *}
                                                  {*(CHAR)                    *}
                        XPHP  :  CHAR_016;        {*OSI-TP P-SEL OF PARTNER   *}
                                                  {*(HEX)                     *}
                        XPOS  :  CHAR_004;        {*OSI-TP ASSOCIATION        *}
                                                  {*REFERENCE                 *}
                        MTXT  :  CHAR_044;
                                    END;
            MP011       = RECORD
                        ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME         *}
```

```
                 OSLP  :  CHAR_008;         {*OSI-LPAP NAME              *}
                 XP00  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*0                          *}
                 XP10  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*1                          *}
                 XP20  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*2                          *}
                 XP30  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*3                          *}
                 XP40  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*4                          *}
                 XP50  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*5                          *}
                 XP60  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*6                          *}
                 XP70  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*7                          *}
                 XP80  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*8                          *}
                 XP90  :  CHAR_004;         {*OSI-TP OBJECT IDENTIFIER  *}
                                            {*9                          *}
                 XPOS  :  CHAR_004;         {*OSI-TP ASSOCIATION         *}
                                            {*REFERENCE                  *}
                 MTXT  :  CHAR_092;
                             END;
         MP012       = RECORD
                 XPCT  :  CHAR_004;         {*CMX ERROR TYPE             *}
                 XPCC  :  CHAR_004;         {*CMX ERROR CLASS            *}
                 XPCV  :  CHAR_004;         {*CMX ERROR VALUE            *}
                 XPBC  :  CHAR_004;         {*BCAM INFOWORD              *}
                 XPCO  :  CHAR_004;         {*MESSAGE CORRELATOR NUMBER *}
                 MTXT  :  CHAR_132;
                             END;
         MP013       = RECORD
                 ACPN  :  CHAR_008;         {*ACCESS-POINT-NAME          *}
                 OSLP  :  CHAR_008;         {*OSI-LPAP NAME              *}
                 XPCR  :  CHAR_004;         {*OSI-TP NEGATIVE            *}
                                            {*CONFIRMATION RESULT        *}
                 XPSR  :  CHAR_004;         {*OSI-TP RESULT SOURCE FROM *}
                                            {*PARTNER                    *}
                 XPND  :  CHAR_004;         {*OSI-TP NEGATIVE            *}
                                            {*DIAGNOSTICS                *}
                 XP1B  :  CHAR_005;         {*OSI-TP CCR V2 NOT          *}
                                            {*AVAILABLE                  *}
                 XP2B  :  CHAR_005;         {*OSI-TP PROTOCOL VERSION   *}
                                            {*INCOMPATIBILITY            *}
                 XP3B  :  CHAR_005;         {*OSI-TP CONTENTION WINNER  *}
                                            {*ASSIGNMENT REJECTED        *}
                 XP4B  :  CHAR_005;         {*OSI-TP BID MANDATORY       *}
```

```
                                                {*REJECTED                    *}
                XP5B  :  CHAR_005;              {*OSI-TP NO REASON GIVEN      *}
                XPOS  :  CHAR_004;              {*OSI-TP ASSOCIATION          *}
                                                {*REFERENCE                   *}
                MTXT  :  CHAR_095;
                            END;
MP014           = RECORD
                XPFU  :  CHAR_020;              {*CALLED OSI-TP FUNCTION      *}
                ACPN  :  CHAR_008;              {*ACCESS-POINT-NAME           *}
                OSLP  :  CHAR_008;              {*OSI-LPAP NAME               *}
                XPRE  :  CHAR_004;              {*OSI-TP RETURN CODE          *}
                XPER  :  CHAR_004;              {*OSI-TP ERROR CODE           *}
                XP1I  :  CHAR_004;              {*OSI-TP ADDITIONAL           *}
                                                {*INFORMATION 1               *}
                XP2I  :  CHAR_004;              {*OSI-TP ADDITIONAL           *}
                                                {*INFORMATION 2               *}
                XPOS  :  CHAR_004;              {*OSI-TP ASSOCIATION          *}
                                                {*REFERENCE                   *}
                XPCO  :  CHAR_004;              {*MESSAGE CORRELATOR NUMBER *}
                MTXT  :  CHAR_092;
                            END;
MP015           = RECORD
                XPFU  :  CHAR_020;              {*CALLED OSI-TP FUNCTION      *}
                ACPN  :  CHAR_008;              {*ACCESS-POINT-NAME           *}
                OSLP  :  CHAR_008;              {*OSI-LPAP NAME               *}
                XPLN  :  CHAR_004;              {*OSI-TP LINK                 *}
                XPSR  :  CHAR_004;              {*OSI-TP RESULT SOURCE FROM *}
                                                {*PARTNER                     *}
                XPND  :  CHAR_004;              {*OSI-TP NEGATIVE             *}
                                                {*DIAGNOSTICS                 *}
                XPIN  :  CHAR_004;              {*OSI-TP INITIATOR            *}
                XP1I  :  CHAR_004;              {*OSI-TP ADDITIONAL           *}
                                                {*INFORMATION 1               *}
                XP2I  :  CHAR_004;              {*OSI-TP ADDITIONAL           *}
                                                {*INFORMATION 2               *}
                XPOS  :  CHAR_004;              {*OSI-TP ASSOCIATION          *}
                                                {*REFERENCE                   *}
                XPCO  :  CHAR_004;              {*MESSAGE CORRELATOR NUMBER *}
                MTXT  :  CHAR_084;
                            END;
MP016           = RECORD
                ACPN  :  CHAR_008;              {*ACCESS-POINT-NAME           *}
                OSLP  :  CHAR_008;              {*OSI-LPAP NAME               *}
                XPLN  :  CHAR_004;              {*OSI-TP LINK                 *}
                XPND  :  CHAR_004;              {*OSI-TP NEGATIVE             *}
                                                {*DIAGNOSTICS                 *}
                XPOS  :  CHAR_004;              {*OSI-TP ASSOCIATION          *}
                                                {*REFERENCE                   *}
                MTXT  :  CHAR_124;
```

```
                           END;
     MP017       = RECORD
                 XPPD  :  CHAR_004;        {*OSI-TP PDU TYPE            *}
                 XP1D  :  CHAR_004;        {*OSI-TP DIAGNOSTIC          *}
                                           {*INFORMATION 1              *}
                 XP2D  :  CHAR_004;        {*OSI-TP DIAGNOSTIC          *}
                                           {*INFORMATION 2              *}
                 XP3D  :  CHAR_004;        {*OSI-TP DIAGNOSTIC          *}
                                           {*INFORMATION 3              *}
                 MTXT  :  CHAR_136;
                           END;
     MP018       = RECORD
                 ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME          *}
                 OSLP  :  CHAR_008;        {*OSI-LPAP NAME              *}
                 XPPT  :  CHAR_004;        {*OSI-TP PRIITIVE TYPE       *}
                 XPFS  :  CHAR_010;        {*OSI-TP FSM NAME            *}
                 MTXT  :  CHAR_122;
                           END;
     MP019       = RECORD
                 ACPN  :  CHAR_008;        {*ACCESS-POINT-NAME          *}
                 OSLP  :  CHAR_008;        {*OSI-LPAP NAME              *}
                 XPAP  :  CHAR_020;        {*OSI-TP APDU TYPE           *}
                 XP3I  :  CHAR_040;        {*OSI-TP ADDITIONAL          *}
                                           {*INFORMATION 3              *}
                 MTXT  :  CHAR_076;
                           END;
        {****************************************************************}
        {*       MESSAGE HEADER                                         *}
        {****************************************************************}
        TYPE  {03} KCMSGL    = RECORD CASE REDEFINED OF
     {05}     L1 :(MSGKOPF  (00): CHAR_024);
                                           {  MESSAGE HEADER            }
       {07}      { FILLER_1   PIC X }
                                           {  FILLER                    }
       {07}   L2 :(MSGNR    (01): CHAR_004);
                                           {  MESSAGE NUMBER            }
       {07}      { FILLER_2   PIC X }
                                           {  FILLER                    }
       {07}   L3 :(MSGDATE  (06): CHAR_011);
                                           {  DATE OF ORIGIN            }
                                           {  MM/DD/YYJJJ               }
       {07}      { FILLER_3   PIC X }
                                           {  FILLER                    }
       {07}   L4 :(MSGTIME  (18): CHAR_006);
                                           {  DATE OF ORIGIN            }
                                           {  (HHMMSS)                  }
       {07}   L5 :(MSGYEAR  (06): CHAR_004);
                                           {  YEAR OF ORIGIN (YYYY)     }
        {***********************************************************}
        {*      INSERTS OF MESSAGES                                 *}
        {***********************************************************}
     {05}  LKXXX     :(KXXX   (24): CHAR_152);
     {05}  LK001     :(K001   (24): MK001);
     {05}  LK002     :(K002   (24): MK002);
     {05}  LK003     :(K003   (24): MK003);
     {05}  LK004     :(K004   (24): MK004);
     {05}  LK005     :(K005   (24): MK005);
     {05}  LK006     :(K006   (24): MK006);
     {05}  LK007     :(K007   (24): MK007);
     {05}  LK008     :(K008   (24): MK008);
     {05}  LK009     :(K009   (24): MK009);
     {05}  LK010     :(K010   (24): MK010);
     {05}  LK011     :(K011   (24): MK011);
     {05}  LK013     :(K013   (24): MK013);
     {05}  LK014     :(K014   (24): MK014);
     {05}  LK015     :(K015   (24): MK015);
     {05}  LK016     :(K016   (24): MK016);
     {05}  LK017     :(K017   (24): MK017);
     {05}  LK018     :(K018   (24): MK018);
     {05}  LK019     :(K019   (24): MK019);
     {05}  LK020     :(K020   (24): MK020);
     {05}  LK021     :(K021   (24): MK021);
     {05}  LK022     :(K022   (24): MK022);
     {05}  LK023     :(K023   (24): MK023);
     {05}  LK024     :(K024   (24): MK024);
     {05}  LK025     :(K025   (24): MK025);
     {05}  LK026     :(K026   (24): MK026);
     {05}  LK027     :(K027   (24): MK027);
     {05}  LK029     :(K029   (24): MK029);
     {05}  LK030     :(K030   (24): MK030);
     {05}  LK031     :(K031   (24): MK031);
     {05}  LK032     :(K032   (24): MK032);
     {05}  LK033     :(K033   (24): MK033);
     {05}  LK036     :(K036   (24): MK036);
     {05}  LK040     :(K040   (24): MK040);
     {05}  LK041     :(K041   (24): MK041);
     {05}  LK043     :(K043   (24): MK043);
     {05}  LK045     :(K045   (24): MK045);
     {05}  LK046     :(K046   (24): MK046);
     {05}  LK049     :(K049   (24): MK049);
     {05}  LK050     :(K050   (24): MK050);
     {05}  LK051     :(K051   (24): MK051);
     {05}  LK052     :(K052   (24): MK052);
     {05}  LK053     :(K053   (24): MK053);
     {05}  LK055     :(K055   (24): MK055);
     {05}  LK056     :(K056   (24): MK056);
     {05}  LK058     :(K058   (24): MK058);
```

```
{05}  LK060     :(K060   (24): MK060);
{05}  LK061     :(K061   (24): MK061);
{05}  LK063     :(K063   (24): MK063);
{05}  LK064     :(K064   (24): MK064);
{05}  LK065     :(K065   (24): MK065);
{05}  LK069     :(K069   (24): MK069);
{05}  LK070     :(K070   (24): MK070);
{05}  LK072     :(K072   (24): MK072);
{05}  LK073     :(K073   (24): MK073);
{05}  LK074     :(K074   (24): MK074);
{05}  LK075     :(K075   (24): MK075);
{05}  LK076     :(K076   (24): MK076);
{05}  LK079     :(K079   (24): MK079);
{05}  LK081     :(K081   (24): MK081);
{05}  LK086     :(K086   (24): MK086);
{05}  LK088     :(K088   (24): MK088);
{05}  LK089     :(K089   (24): MK089);
{05}  LK090     :(K090   (24): MK090);
{05}  LK091     :(K091   (24): MK091);
{05}  LK092     :(K092   (24): MK092);
{05}  LK093     :(K093   (24): MK093);
{05}  LK094     :(K094   (24): MK094);
{05}  LK097     :(K097   (24): MK097);
{05}  LK098     :(K098   (24): MK098);
{05}  LK101     :(K101   (24): MK101);
{05}  LK104     :(K104   (24): MK104);
{05}  LK105     :(K105   (24): MK105);
{05}  LK106     :(K106   (24): MK106);
{05}  LK107     :(K107   (24): MK107);
{05}  LK108     :(K108   (24): MK108);
{05}  LK109     :(K109   (24): MK109);
{05}  LK115     :(K115   (24): MK115);
{05}  LK116     :(K116   (24): MK116);
{05}  LK117     :(K117   (24): MK117);
{05}  LK119     :(K119   (24): MK119);
{05}  LK120     :(K120   (24): MK120);
{05}  LK121     :(K121   (24): MK121);
{05}  LK123     :(K123   (24): MK123);
{05}  LK124     :(K124   (24): MK124);
{05}  LK125     :(K125   (24): MK125);
{05}  LK126     :(K126   (24): MK126);
{05}  LK128     :(K128   (24): MK128);
{05}  LK130     :(K130   (24): MK130);
{05}  LK135     :(K135   (24): MK135);
{05}  LK137     :(K137   (24): MK137);
{05}  LK138     :(K138   (24): MK138);
{05}  LK139     :(K139   (24): MK139);
{05}  LK140     :(K140   (24): MK140);
{05}  LK141     :(K141   (24): MK141);
```

```
   {05}  LK142     :(K142   (24): MK142);
   {05}  LK143     :(K143   (24): MK143);
   {05}  LK144     :(K144   (24): MK144);
   {05}  LK145     :(K145   (24): MK145);
   {05}  LK146     :(K146   (24): MK146);
   {05}  LK147     :(K147   (24): MK147);
   {05}  LK150     :(K150   (24): MK150);
   {05}  LK151     :(K151   (24): MK151);
   {05}  LK152     :(K152   (24): MK152);
   {05}  LP001     :(P001   (24): MP001);
   {05}  LP002     :(P002   (24): MP002);
   {05}  LP003     :(P003   (24): MP003);
   {05}  LP004     :(P004   (24): MP004);
   {05}  LP005     :(P005   (24): MP005);
   {05}  LP006     :(P006   (24): MP006);
   {05}  LP007     :(P007   (24): MP007);
   {05}  LP008     :(P008   (24): MP008);
   {05}  LP009     :(P009   (24): MP009);
   {05}  LP010     :(P010   (24): MP010);
   {05}  LP011     :(P011   (24): MP011);
   {05}  LP012     :(P012   (24): MP012);
   {05}  LP013     :(P013   (24): MP013);
   {05}  LP014     :(P014   (24): MP014);
   {05}  LP015     :(P015   (24): MP015);
   {05}  LP016     :(P016   (24): MP016);
   {05}  LP017     :(P017   (24): MP017);
   {05}  LP018     :(P018   (24): MP018);
   {05}  LP019     :(P019   (24): MP019);
                  ELSE: ();
                  END;  {KCMSGL}
END.  {KCMSL}
```

Package KCPADL

```
{****************************************************************+**}
{*                                                               +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                 ALL RIGHTS RESERVED                           +**}
{*                                                               +**}
{****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0      +**}
PACKAGE BODY Kcpadl;
   { leer }
begin
   { leer }
END.  {Kcpadl}
```

```
{****************************************************************+**}
{*                                                               +**}
{*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1992  +**}
{*                 ALL RIGHTS RESERVED                           +**}
{*                                                               +**}
{****************************************************************+**}
{*      SIEMENS NIXDORF INFORMATIONSSYSTEME AG openUTM  4.0      +**}
{*******************************************************************}
{*                                                                 *}
{*      Structures for resultinformation                           *}
{*      of padm function  KCSPADM                                  *}
{*      for  PASCAL-XT                        KCpadl               *}
{*******************************************************************}
PACKAGE Kcpadl;
type
   pic_X          = char;
   pic_XX         = packed array [1..2] of pic_X;
   pic_X_3        = packed array [1..3] of pic_X;
   pic_X_6        = packed array [1..6] of pic_X;
   pic_X_8        = packed array [1..8] of pic_X;
   pic_X_10       = packed array [1..10] of pic_X;
   pic_9          = '0'..'9';
   pic_99         = packed array [1..2] of pic_9;
   pic_999        = packed array [1..3] of pic_9;
   record_9       = packed array [1..9] of char;
   record_44      = packed array [1..44] of char;
   REDEFINES  =                   { simulates COBOL redefinitions }
              (     v1, v2, v3, v4, v5, v6, v7, v8, v9,
               v10, v11,v12,v13,v14,v15,v16,v17,v18,v19,
               v20, v21,v22,v23,v24,v25,v26            );
TYPE
       {03}         KCPADM1      = record case REDEFINES of
        {05}    v1 : (KCRETPAD(00): record_44);
                                         { max. length of information }
```

```
{*****************************************************************}
{*          structure for modification KCOM = AI                 *}
{*****************************************************************}
        {05}    v2 : (KCACKINF(00): record_44);
         {07}   v3 : (KCACKCID(00): pic_x_8);
                                        { Printer Control ID    }
         {07}   v4 : (KCGENUI (08): pic_x_8);
                                        { USER ID               }
         {07}   v5 : (KCDPUTID(16): pic_x_8);
                                        { DPUT ID               }
         {07}   v6 : (KCGENTIM(24): record_9);
                                        { generation time       }
           {09} v7 : (KCGENDOY(24): pic_x_3);
                                        { day of year           }
           {09} v8 : (KCGENHR (27): pic_xx);
                                        { hour                  }
           {09} v9 : (KCGENMIN(29): pic_xx);
                                        { minute                }
           {09} v10: (KCGENSEC(31): pic_xx);
                                        { Second                }
         {07}   v11: (KCSTTIM (33): record_9);
                                        { desired start time    }
           {09} v12: (KCSTDOY (33): pic_x_3);
                                        { day of year           }
           {09} v13: (KCSTHR  (36): pic_xx);
                                        { hour                  }
           {09} v14: (KCSTMIN (38): pic_xx);
                                        { minute                }
           {09} v15: (KCSTSEC (40): pic_xx);
                                        { second                }
         {07}   v16: (KCPOSMSG(42): pic_x);
                                        { positive              }
                                        { acknowl. job          }
         {07}   v17: (KCNEGMSG(43): pic_x);
                                        { negative              }
                                        { acknowl. job          }
{*****************************************************************}
{******         structure for modification KCOM=PI               *}
{*****************************************************************}
        {05}    v18: (KCPRTINF(00): record_44);
                                        { printer information   }
         {07}   v19: (KCPRTCID(00): pic_x_8);
                                        { printer ID            }
         {07}   v20: (KCSTATE (08): pic_x_3);
                                        { PTRM state            }
         {07}   v21: (KCCON   (11): pic_x);
                                        { Y: PTRM connected     }
                                        { N: PTRM disconnected  }
         {07}   v22: (KCPRTMOD(12): pic_xx);
```

```
                                               { print mode            }
          {07}   v23: (KCLTRMNM(14): pic_x_8);
                                               { LTERM name            }
          {07}   v24: (KCFPMSGS(22): pic_x_6);
                                               { no output messages    }
          {07}   v25: (KCDPMSGS(28): pic_x_6);
                                               { no delayed messages   }
          {07}        {FILLER  (34): pic_x_10 }
                                               { not used              }
          else: ();  end;  {kcpadml}
      end.  {kcpadl}
```

Package TIAMCTRL (example)

This package is not delivered with UTM!

```
package TIAMCTRL;

(*************************************************************)
(*                                                           *)
(*      TIAMCTRL   V801                                      *)
(*      line mode control characters                         *)
(*                                                           *)
(*************************************************************)

(* LOGICAL RECORD DELIMITERS *)
const
   CC_NEW_LINE          = #'15';
   CC_NEW_PAGE          = #'0C';
   CC_CONT_SAME_LINE    = #'0D';
   CC_CONT_LINE_N       = #'29';
   CC_SHEET_FEED_N      = #'21';
   CC_CONT_ACT_POS      = #'20';

(* LOGICAL UNIT DELIMITERS *)
   CC_EMPH_LAYOUT1      = #'1D';
   CC_EMPH_LAYOUT2      = #'1F';
   CC_EMPH_LAYOUT3      = #'13';
   CC_EMPH_LAYOUT4      = #'14';
   CC_NORMAL_LAYOUT     = #'1E';
   CC_DARK_LAYOUT       = #'12';
   CC_PART_LINE_UP      = #'2C';
   CC_PART_LINE_DOWN    = #'2B';

   CC_SECOND_CHAR_SET   = #'0E';
   CC_NORMAL_CHAR_SET   = #'0F';

   CC_START_PROT_AREA   = #'36';
   CC_END_PROT_AREA     = #'08';
   CC_START_NUM_DATA    = #'11';

   CC_VERT_MOVE_IND     = #'24';
   CC_HORIZ_MOVE_IND    = #'23';
   CC_LEFT_MARGIN       = #'38';
   CC_START_PROP_TYPE   = #'1A';
   CC_END_PROP_TYPE     = #'1B';
   CC_MAX_LINE_LEN      = #'33';
   CC_MAX_LINE_NUM      = #'35';

(* SPECIAL FUNCTIONS *)
   CC_DELETE_CHAR       = #'07';
   CC_BACKSPACE         = #'16';
   CC_SUBSTITUTE        = #'3F';

(* PHYSICAL UNIT DELIMITERS *)
   CC_PHYS_ESC          = #'27';
   CC_PHYS_DC4          = #'3C';
   CC_PHYS_HT           = #'05';
   CC_PHYS_VT           = #'0B';
end.

PACKAGE BODY TIAMCTRL;
begin
end.
```

Index

Contents

*open*UTM V4.0

Supplement for Pascal-XT (BS2000/OSD)

User Guide

Target group
Programmers of UTM Pascal-XT applications
Contents
– Translation of the KDCS program interface into the language Pascal-XT
– All the information required by programmers of UTM Pascal-XT applications
Applications
BS2000 transaction processing

Edition: February 1997

File: utm_pas.pdf