1 Einleitung

Der universelle Transaktionsmonitor *open*UTM bietet als umfassende Middleware-Plattform alle Möglichkeiten, die Sie für das Design und den Einsatz transaktionsorientierter OLTP-Anwendungen benötigen. Darüber hinaus ist in *open*UTM die Funktionalität eines kompletten Message Queuing Systems integriert.

Durch optimale Performance, ausgereifte Security-Funktionen und höchste Verfügbarkeit ist *open*UTM auch für Anwendungsszenarien geeignet, bei denen herkömmliche OLTP-Systeme längst an ihre Grenzen stoßen.

*open*UTM bildet ein sicheres und effizientes Framework für moderne multi-tier Client/Server-Architekturen: *open*UTM sorgt für die Steuerung globaler Transaktionen, optimiert den Einsatz von System-Ressourcen (Arbeitsspeicher, CPU etc.), übernimmt das Management von parallelen Zugriffen, kümmert sich um Zugangs- und Zugriffskontrollen, den Aufbau von Netzverbindungen und vieles mehr.

Der Name „*open*UTM“ weist bereits auf wichtige Leistungsmerkmale hin:

open ... weil *open*UTM dem von X/Open definierten Referenzmodell für „Distributed Transaction Processing (DTP)“ entspricht und die von X/Open standardisierten offenen Schnittstellen unterstützt.

Universal ... weil *open*UTM unterschiedliche Welten verbindet und für die unterschiedlichsten Einsatzszenarien konzipiert ist: *open*UTM integriert heterogene Netze, Plattformen, Resource Manager und Anwendungen.

Transaction ... weil *open*UTM anwendungsübergreifend volle Transaktionssicherheit gewährleistet, entsprechend den klassischen ACID-Eigenschaften Atomacity, Consistency, Isolation und Durability.

Monitor ... weil *open*UTM nicht nur „bloße“ Transaktionsverarbeitung bietet, sondern das Management von verteilten, unternehmensweiten IT-Lösungen ermöglicht.

1.1 Konzept und Zielgruppen dieses Handbuchs

Dieses Handbuch soll Pascal-XT-Programmierer von *open*UTM-Anwendungen in ihrer Arbeit unterstützen. Es ergänzt das Basishandbuch „Anwendungen programmieren mit KDCS für COBOL, C und C++“.

Zu seinem Verständnis sind Grundkenntnisse des Betriebssystems und von *open*UTM sowie das Basishandbuch „Anwendungen programmieren mit KDCS für COBOL, C und C++“ notwendig. Zur Vertiefung der gebotenen Informationen können die *open*UTM-Handbücher „Anwendungen generieren und betreiben“, „Meldungen, Test und Diagnose“, sowie „Konzepte und Funktionen“ herangezogen werden.

Dieses Handbuch beschreibt die sprachspezifischen Besonderheiten bei der Erstellung von Pascal-XT-Teilprogrammen.

Es enthält in Pascal-XT erstellte Beispiele zu einzelnen KDCS-Aufrufen, zum Event-Service MSGTAC sowie ein Beispiel für eine komplette *open*UTM-Anwendung.

Im Kapitel „Datenstrukturen für Pascal-XT“ auf Seite 67 finden Sie eine Aufstellung aller Pascal-XT-Datenstrukturen.

Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen SYSRME.*produkt.version.sprache*. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei können Sie mit dem Kommando /SHOW-FILE oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

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

bei SPOOL -Versionen kleiner 3.0A:

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

2 Programmaufbau bei Pascal-XT-Teilprogrammen

In diesem Kapitel erfahren Sie:

- Wie ein Pascal-XT-Teilprogramm als Unterprogramm zu erstellen ist.
- Wie die UTM-Pascal-XT-Pakete aussehen und wie ein KDCS-Aufruf in Pascal-XT programmiert werden muß.
- Welche Besonderheiten und Einschränkungen es bei Pascal-XT-Teilprogrammen gibt.

Grundlage der Beschreibung ist der Pascal-XT-Compiler der Version 2.2B.

2.1 Pascal-XT-Teilprogramm als Unterprogramm

Dieser Abschnitt behandelt folgende Themen:

- Spezifikation und Implemtierung von UTM-Pascal-XT-Teilprogrammen
- Paket-Spezifikation
- Konstanten und Datenstrukturen für UTM-Pascal-XT-Programme
- Compiler, Laufzeitsysteme und Generierungsoptionen

2.1.1 Spezifikation und Implementierung von UTM-Pascal-XT-Teilprogrammen

UTM-Teilprogramme einschließlich Event-Exits sind Unterprogramme der UTM-Main Routine. Daraus ergeben sich für die Struktur dieser Programme einige Konsequenzen:

– UTM-Pascal-XT-Teilprogramme müssen als ENTRY-Prozeduren geschrieben werden. Sie müssen daher in Paketen implementiert werden.

– Die Namen der ENTRY-Prozeduren (Prozedurbezeichner) müssen in den ersten 8 Zeichen mit den in der KDCDEF-Anweisung PROGRAM angegebenen Teilprogrammnamen übereinstimmen.

– Die Parameterliste der ENTRY-Prozeduren muß mindenstens zwei formale Parameter enthalten (für den Kommunikationsbereich und den Standard-Primären Arbeitsbereich SPAB). Bei Verwendung zusätzlicher Speicherbereiche, die mit der KDCDEF-Anweisung AREA definiert wurden, muß die Parameterliste die entsprechenden formalen Parameter zusätzlich enthalten. Diese Parameter in den Entry-Prozedur-Deklarationen müssen als Variablenparameter „VAR“... definiert sein.

Sie können mehrere ENTRY-Prozeduren (Teilprogramme) in einem Paket implementieren.

Um kompatibel zu sein und mit fehlerfreien Einträgen zu arbeiten, wird empfohlen, die durch UTM festgelegten Datenstrukturen und Konstanten aus den Paket-Spezifikationen der Bibliothek SYSLIB.UTM.040.PASC zu importieren.

Die Datenstrukturen und Konstanten werden im folgenden Abschnitt ausführlich beschrieben, ihre Verwendung bei den einzelnen Aufrufen ist im Basishandbuch „Anwendungen programmieren mit KDCS für COBOL, C und C++“ beschrieben.

2.1.2 Paket-Spezifikation

In der Paket-Spezifikation deklarieren Sie die Konstanten, Datenstrukturen und Prozeduren, die „nach außen sichtbar“ sind, d.h. mindestens:

– Die Strukturen des Kommunikationsbereichs (KB) und des Standard-Primären Arbeitsbereichs (SPAB) sowie ggf. weiterer UTM-Speicherbereiche, die bei der Generierung mit der KDCDEF-Anweisung AREA definiert sind, als Typdeklarationen

– Die Teilprogramme, die Sie in diesem Paket realisieren wollen, als ENTRY-Prozeduren

– Dateivariablen, die Sie in der Programmparameterliste aufgeführt haben, als Variablendeklarationen.

2.1.3 Konstanten und Datenstrukturen für UTM-Pascal-XT-Programme

Um die Datenbereiche zu strukturieren, werden mit *open*UTM Pakete ausgeliefert, die diese Konstanten und Datenstrukturen enthalten. Im folgenden finden Sie eine Übersicht über Umfang und Funktion der vordefinierten Pakete.

Paket-Name **wichtigste Datentypen**	**Inhalt und Bedeutung**
KCKBL	definiert folgendes: – die KDCS-Operationsschlüssel. Sie sind als symbolische Konstante definiert. Ihre Verwendung garantiert die Gültigkeit der Operationscodes.
KCKB	– den durch UTM in seiner Struktur festgelegten Kopf des Kommunikationsbereiches. Er enthält: - aktuelle Daten des Vorgangs und Programms - Rückgaben nach einem Aufruf an UTM. An diesen KB-Kopf kann sich ein Programmbereich anschließen, den Sie selbst definieren müssen (siehe Beispiele).
KCPAL	– den KDCS-Parameterbereich, der die Parameter eines Aufrufs an UTM aufnimmt. Meist wird der Parameterbereich an den Anfang des Standard Primären Arbeitsbereiches (SPAB) gelegt, dessen Struktur Sie selbst definieren müssen (siehe Beispiele).
KCATL (FIELD_ATTRIBUTE_PACKAGE)	definiert die symbolischen Namen, mit denen Sie bei Verwendung von +Formaten die Attributfelder von Formaten ändern können. Für eine Reihe von Standard-Attributkombinationen sind Konstantennamen definiert.
KCDFL	definiert die KDCS-Bildschirmfunktionen: Zur Versorgung des Feldes KCDF des KDCS-Parameterbereichs können Sie diese Konstantendefinitionen benutzen, wenn Sie bestimmte Funktionen eines Terminals anfordern (nur sinnvoll, wenn der Partner ein Terminal ist).

Paket-Name **wichtigste Datentypen**	**Inhalt und Bedeutung**
KCINL KCINFL	Datenstruktur für die Informationen, die der UTM-Aufruf INFO liefert. Wenn Sie einen (für ein Teilprogramm) globalen Nachrichtenbereich definieren, sollten Sie KCINFL als eine der Varianten dieses RECORD-Typs deklarieren.
KCMSL	stellt Ihnen Datenstrukturen für die UTM-Meldungen zur Verfügung. Sie benötigen diese, wenn Sie Meldungen in einer MSGTAC-Routine auswerten wollen.
KCINPL	Datenstruktur für den Event-Exit INPUT. KCINPL enthält Ein- und Ausgabeparameter; die Ausgabeparameter bestimmen die Wirkung der Eingabe.
KCDADL	Datenstruktur für den DADM-Aufruf. Diese Datenstruktur sollten Sie beim Aufruf DADM RQ über den Nachrichtenbereich legen.
KCPADL	ist eine Datenstruktur für den PADM-Aufruf. Diese Datenstruktur sollten Sie beim Aufruf PADM AI bzw. PI über den Nachrichtenbereich legen.
KCAPROL	definiert einen optionalen zweiten Parameterbereich für den Funktionsaufruf APRO. KCAPROL dient zur Auswahl spezieller OSI TP-Funktionskombinationen.
KCINIL	definiert einen zweiten Parameterbereich für den Funktionsaufruf INIT PU. In diesem Parameterbereich liefert UTM die mit INIT PU angeforderten Informationen zurück.
KCCFL	definiert den zweiten Parameter, den UTM beim Event-Exit INPUT übergibt. In diesem Parameter übergibt UTM die Inhalte der Steuerfelder von Bildschirmformaten an das Teilprogramm. Dieser zweite Parameter wird deshalb auch Steurfeldbereich (Control Fields) genannt.

Die Pakete, die Sie verwenden wollen, importieren Sie mit einer WITH-Klausel in Ihr Paket. Namen aus den importierten Paketen, die Sie direkt, d.h. ohne sie voll zu qualifizieren, verwenden wollen, müssen Sie in einer USE-Klausel aufführen.

Beispiele für die Deklaration Ihrer Datenbereiche finden Sie in den Beispielen im folgenden Abschnitt und ab Seite 33. Eine Auflistung dieser Pakete finden Sie in Kapitel „Datenstrukturen für Pascal-XT“ auf Seite 67ff.

2.2 Compiler, Laufzeitsysteme und Generierungsoptionen

Die folgende Tabelle zeigt die Compiler, Laufzeitsysteme und Generierungsoptionen, die Sie nutzen können, um Pascal-XT-Teilprogramme zu erzeugen und in einem UTM-Anwenderprogramm ablaufen zu lassen.

Die erste Spalte der Tabelle enthält alle Compilerversionen, die Sie zur Erzeugung der Objektmodule des Teilprogramms nutzen können.
Die zweite Spalte der Tabelle enthält das Laufzeitsystem, das Sie bei Verwendung des entsprechenden Compilers einsetzen müssen.
Die dritte Spalte enthält den Wert des COMP-Parameters der UTM-Generierungsanweisung PROGRAM, den Sie bei Verwendung der jeweiligen Compiler angeben müssen.

Pascal-XT-Compiler	**Laufzeitsystem**	**COMP=**
V2.1	V2.1	PASCAL-XT
V2.1	V2.2	ILCS
V2.2	V2.2	ILCS

In einer UTM-Anwendung darf nur ein Pascal-XT-Laufzeitsystem gebunden werden.

Pascal-XT nutzt die ILCS-Schnittstelle. Über ILCS lassen sich Teilprogramme aus mehreren Quellcodes unterschiedlicher Programmiersprachen verbinden. In der Freiabemitteilung finden Sie eine Auflistung aller Compiler und Laufzeitsysteme, die eine Mischung erlauben.

2.3 Namenskonventionen

Die Namen der deklarierten ENTRY-Prozeduren müssen innerhalb der ersten 8 Zeichen eindeutig sein, da nur die ersten 8 Zeichen in das Externadreßbuch der erzeugten Bindemodule übernommen werden.

Der Name des Pakets muß sich ebenfalls in den ersten 8 Zeichen von den Namen der ENTRY-Prozeduren unterscheiden, um Konflikte beim Binden der Anwendung zu vermeiden.

Alle Namen, die mit „KDC“, „KC“ oder „I“ beginnen, sind reserviert.

Alle übrigen Namen können Sie frei nach den Regeln der Sprache Pascal-XT vergeben.

2.4 Deklarationen

Dieser Abschnitt behandelt folgende Themen:

– Deklaration der ENTRY-Prozeduren
– Typdeklaration
– Datenbereiche als Pascal-XT-Pakete

2.4.1 Deklaration der ENTRY-Prozeduren

Jedes Teilprogramm deklarieren Sie als ENTRY-Prozedur nach folgendem Schema:

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

tpname ist der Name des Teilprogramms, der in der PROGRAM-Anweisung angegeben wurde; er muß innerhalb der ersten 8 Zeichen eindeutig sein.

kb ist der Name des Kommunikationsbereichs (vom Typ *kckbc*; s.o.)

spab ist der Name des Standard Primären Arbeitsbereiches (vom Typ *kcspab*; s.o.).

p1 ... pn sind die Namen der mit der AREA-Anweisung definierten und als Assembler-CSECT übersetzten zusätzlichen Datenbereiche. Ihre Datenstrukturen müssen zuvor mit den Namen *id_p1 ... id_pn* deklariert worden sein. Die Reihenfolge dieser formalen Parameter muß der Reihenfolge der AREA-Anweisungen entsprechen. Es müssen alle Bereiche bis zum letzten hier verwendeten in der Parameterliste aufgeführt sein.
Wird keiner dieser Bereiche in dem Teilprogramm *tpname* verwendet, kann die Angabe entfallen.

2.4.2 Typdeklarationen

Die aktuellen Parameter, mit denen ein UTM-Teilprogramm aufgerufen wird, sind strukturierte Datenfelder. Für die formalen Parameter der ENTRY-Prozeduren müssen Sie daher entsprechende Datentypen deklarieren.
Dies sind mindestens der Kommunikationsbereich (KB) und der Standard Primäre Arbeitsbereich (SPAB). Der erste Teil beider Strukturen ist durch *open*UTM festgelegt. Die Deklarationen dieser festen Teile sind in der Paket-Spezifikation KCKBL enthalten, die unter dem Namen „KCKBLS“ in der Bibliothek „SYSLIB.UTM.040.PASC“ ausgeliefert wird. Um Kompatibilität der Teilprogramme zu erreichen und ihre Lesbarkeit zu erhöhen, wird empfohlen, diese Deklarationen (in einer WITH-Klausel) zu importieren.

Beispiel

```
with Kckbl;                                                    1)
from Kckbl use kckb, kcpal;                                    2)
 ...
package Anwendung;
   ...
type
  kckbc = record                                               3)
    kb_head: kckb;                                             4)
    kb_beliebig: packed array[1..22] of char;                  5)
    kb_startort: packed array [1..2] of char;                  5)
    kb_zielort: packed array [1..2] of char;                   5)
    kb_flgtag: packed array [1..5] of char;                    5)
    kb_flgnr1: packed array [1..5] of char;                    5)
    kb_flgnr2: packed array [1..5] of char;                    5)
  end;

  iforma = record                                              6)
    ...
  end;

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

1) Importieren der Paket-Spezifikation KCKBL

2) Die hier aufgeführten Namen können ohne Selektoren verwendet werden (z.B.: KCPAL statt: KCKBL.KCPAL)

3) Typdeklaration für den Kommunikationsbereich

4) Kopf des Kommunikationsbereichs (durch UTM in seiner Struktur festgelegt)

5) Anwendungsspezifischer Programmbereich

6) Datenstruktur eines vorher erstellten Bildschirmformats

7) Struktur des SPAB

8) Parameterbereich

Zusätzlich zu dem Kontrollbereich und dem SPAB können Sie maximal 99 weitere Speicherbereiche als Prozedurparameter für die ENTRY-Prozeduren definieren, die dann als gemeinsame Datenbereiche innerhalb einer UTM-Anwendung verwendet werden können. Diese Bereiche können in einem anwendungsglobalen oder -lokalen Common Memory Pool liegen oder zum Anschlußprogramm statisch gebunden werden.

Das zum Binden erforderliche Bindemodul erstellen Sie als Assembler-CSECT, deren Länge Sie mit DS-Anweisungen bestimmen (Durch DC-Anweisungen können Sie statische Vorbesetzungen definieren). Die Typdeklaration, die die Struktur dieses Bereichs beschreibt, muß in der Paket-Spezifikation der Teilprogramme, die diese Bereiche verwenden, enthalten sein oder aus einer anderen Paket-Spezifikation importiert werden.

Beispiel

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

Bei der Generierung definieren Sie diesen Bereich mit der Anweisung AREA (nicht mit der PROGRAM-Anweisung). Hier wird auch die Art des Bereichs (lokal oder global) festgelegt.

Beim Aufruf eines Teilprogramms werden die so definierten Bereiche in der Reihenfolge der AREA-Anweisungen übergeben. Bei der Deklaration der ENTRY-Prozeduren müssen Sie diese Reihenfolge in der Prozedurparameterliste beachten. Liegen diese Bereiche in Common Memory Pools, so sind allein die Teilprogramme für die Synchronisation der Zugriffe verantwortlich.

Hinweis

Diese Funktion gehört nicht zur DIN-Norm 66 265.

2.4.3 Datenbereiche als Pascal-XT-Pakete

Das Paketkonzept von Pascal-XT bietet die weitere Möglichkeit, gemeinsame Datenbereiche als globale Variable in eigenen Paketen zu definieren.

Die Paket-Spezifikation enthält:

– Typ- und Variablen-Deklarationen für diejenigen Bereiche, auf die Sie von Ihren Teilprogrammen aus direkt zugreifen wollen

– Deklarationen privater Zeigertypen für Bereiche, deren Struktur Sie verbergen wollen

– Deklarationen von Zugriffsprozeduren für die privaten Datentypen (und ggf. Deklarationen weiterer Prozeduren und Funktionen zur Bearbeitung dieser Bereiche).

Die Paket-Implementierung enthält:

– Die Deklaration der Domänentypen zu den privaten Zeigertypen

– Die Rümpfe der in der Paket-Spezifikation deklarierten Prozeduren und Funktionen

– Im Rumpf (dem äußersten Block) der Implementierung ggf. Anweisungen zur Initialisierung der Paket-Variablen.

Die Spezifikationen dieser Pakete importieren Sie mit einer WITH-Klausel (und ggf. mit USE-Klauseln) in die Pakete Ihrer Teilprogramme.
Dabei müssen Sie folgende Regeln beachten:

– Die so definierten Bereiche dürfen bei der Generierung weder in einer PROGRAM- noch in einer AREA-Anweisung genannt werden.

– Diese Bereiche werden beim Aufruf eines Teilprogramms **nicht** als aktuelle Parameter übergeben.

Werden solche Bereiche nur lokal verwendet, können Sie diese Pakete mit dem ROOT-Modul zusammenbinden. Sie können solche Pakete aber auch in einen (lokalen oder globalen) Common Memory Pool laden (siehe auch Seite 19).

Regeln zur Bildung von Paketnamen

Bei diesem Verfahren müssen Sie die vom Pascal-XT Compiler verwendeten Regeln zur Bildung von Paketnamen beachten:

– Der Paketname wird auf 7 Zeichen gekürzt; ist er kürzer als 7 Zeichen, wird mit dem Zeichen „#“ auf 7 Zeichen aufgefüllt.

– Das achte Zeichen dient der Unterscheidung der Module, z. B. „C“ für das Code-Modul und „D“ für das Datenmodul.

– Unterstriche werden durch „#“ ersetzt.

Beispiel

```
package BEREICHE;
type
  vrec = record
    record_length: short_integer;
    dummy: short_integer;
    data: packed array[1..2044] of char;
  end;

var
  bereich1 : vrec;
  bereich2 : vrec;
end.

package body BEREICHE;
begin
  with bereich1 do begin record_length := 4; dummy := #4040 end;
  with bereich2 do begin record_length := 4; dummy := #4040 end;
end.
```

Die vom Pascal-XT Compiler erzeugten Namen sind in diesem Beispiel:

```
BEREICHC          für das Code-Modul
BEREICHD          für das Datenmodul
```

Beide Module müssen Sie einbinden bzw. in den Memory Pool laden.

2.5 Paket-Implementierung

Die Paket-Implementierung (PACKAGE BODY) kann außer den Rümpfen der ENTRY-Prozeduren, die Sie in der Paket-Spezifikation deklariert haben, auch noch Konstanten-, Typ- und Variablendeklarationen sowie Prozeduren und Funktionen enthalten, die Sie nur lokal innerhalb dieses Paketes verwenden wollen (z.B. INLINE-Prozeduren zur bequemeren Formulierung der KDCS-Aufrufe).

Bei der Implementierung der ENTRY-Prozeduren müssen Sie nur die wenigen Regeln der Transaktionsverarbeitung beachten, wie sie im Kapitel „Aufbau und Einsatz von UTM-Programmen“ im Basishandbuch „Anwendungen programmieren mit KDCS für COBOL, C und C++“ ausführlich beschrieben sind:

– Reentrant-Fähigkeit
– Strenger Dialog (in Dialogprogrammen)

Auf Reentrant-Fähigkeit für Shared Code brauchen Sie nicht zu achten. Pascal-XT-Programme bzw. -Prozeduren sind immer reentrant-fähig.

Aufruf von UTM-Funktionen

Die Kommunikation Ihrer Teilprogramme mit der Main Routine KDCROOT geschieht ausschließlich durch Aufrufe der externen Prozedur „KDCS“.

UTM erwartet beim Aufruf dieser Prozedur mindestens einen aktuellen Parameter, die Adresse des Parameterbereichs, den Sie vor dem Aufruf von KDCS mit dem Operationscode und ggf. mit zusätzlichen Werten versorgen müssen.

Bei den meisten KDCS-Operationen wird zusätzlich als zweiter Parameter die Adresse des Nachrichtenbereichs verlangt.

In Pascal-XT ist es (im Unterschied zu COBOL) nicht möglich, Prozeduren mit einer variablen Anzahl von Parametern zu deklarieren oder aufzurufen. Die Datentypen der aktuellen und formalen Parameter müssen außerdem verträglich sein (siehe Handbuch „Pascal-XT, Beschreibung“). Um korrekt vorzugehen, gibt es zwei Wege:

– Sie deklarieren einen Datentyp (z.B. *kcnb*) für den Nachrichtenbereich als RECORD-Typ. Alle Nachrichtenstrukturen, die Sie verwenden, formulieren Sie als Varianten dieses RECORD-Typs. Die Prozedur KDCS deklarieren Sie dann wie folgt:

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

  (KCPAL ist der im Paket KCKBL deklarierte RECORD-Typ für den Parameterbereich).

– Sie schreiben für die verschiedenen KDCS-Operationen, die Sie verwenden, spezielle Prozeduren in Pascal-XT, in deren Vereinbarungsteil Sie die Prozedur KDCS deklarieren.

Beispiel

```
function INIT: boolean;
   procedure KDCS (var p: kckbl.kcpal); COBOL;
begin
   spab.kcpac.kcop := kckbl.init;
   kdcs (spab.kcpac);
   init := (kckbc.kb_kopf.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 := '         '; (* Zeilenmodus einschalten *)
   end;
   kdcs (spab.kcpac, line);
   put_line := (kckbc.kb_kopf.kcrccc = '000');
end (*GET_LINE*);
```

Die Typ- und Variablennamen dieses Beispiels sind den Beispielen des vorigen Abschnitts entnommen.

Hinweis

Die Direktive COBOL bei der Deklaration der Prozedur KDCS ist obligatorisch, da nur so die aktuellen Parameter beim Aufruf der KDCS-Prozedur COBOL-konform übergeben werden, wie es *open*UTM verlangt.

2.6 Event-Exits

Die Event-Exits INPUT, START, SHUT und VORGANG dürfen keine KDCS-Aufrufe enthalten. Sie müssen über das normale Prozedurende verlassen werden.

Event-Exit START

Entdeckt das START-Teilprogramm einen Fehler (weil z.B. versucht wurde, eine nicht vorhandene Datei zu eröffnen), und soll deshalb der Start beendet werden, so muß der Event-Exit für eine abnormale Programmbeendigung sorgen (TERM-Makro mit entsprechendem Returncode). Dazu müssen Sie sich eine Assembler-Prozedur schreiben, die Sie in diesem Fall aufrufen (als EXTERNAL- oder INTERNAL-Prozedur, siehe „Pascal-XT Benutzerhandbuch“). Der Aufruf der Prozedur SET_RETURN_CODE aus dem vordefiniertem Paket BS2000CALLS reicht hier nicht aus!

Event-Exit SHUT

Der Event-Exit SHUT wird in folgenden Fällen aufgerufen:

– Bei Beendigung eines Anwendungsprogramms
– Bei PEND ER, auch als Folge eines nicht behandelten STXIT-Ereignisses in einem Pascal-XT-Teilprogramm (in einem Exception-Teil)
– Beim Beenden des Anwendungsprogramm-Austauschs.

2.7 Pascal-XT-spezifische Besonderheiten

In diesem Abschnitt erfahren Sie:

– Was Sie beim Binden beachten müssen
– Welche Module Sie shareable laden können und wie Sie Shared Code erzeugen
– Wie Sie Pascal-XT-Adressierungshilfen erstellen und einsetzen
– Wie Sie im erweiterten Zeilenmodus arbeiten können

2.7.1 Hinweise zum Binden

Sie sollten das Pascal-XT-Laufzeitsystem entweder statisch zum UTM-Anwendungsprogramm binden oder den shareable Teil gemeinsam benutzbar machen.

Beim statischen Binden des Pascal-Laufzeitsystems müssen Sie beachten, daß die CSECT-Namen ILMSINI und IMLEND im Pascal-Laufzeitsystem und im CRTE (Common Runtime Environment) gleich sind. Daher sind beim Binden des UTM-Anwendungsprogrammes zusätzliche Maßnahmen erforderlich. Diese sind abhängig davon, welchen Binder Sie benutzen, BINDER oder TSOSLNK.

BINDER

Bevor beim Binden des UTM-Anwendungsprogramms die Laufzeitsysteme eingebunden werden, ist folgender Bindeabschnitt nötig:

```
//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
```

Danach folgt:

```
// 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

Hier müssen Sie ein Modul (PASPART) vorbinden, deren CSECTs (IMLEND und IMLSINI) Sie mit Hilfe von LMS umbenennen müssen.
Beim Binden des UTM-Anwendungsprogramms ist folgendes zu beachten: Bevor das Laufzeitsystem mit einer RESOLVE-Anweisung eingebunden wird, müssen Sie das Modul (PASPART) mit einer INCLUDE-Anweisung in das UTM-Anwendungsprogramm einfügen.

```
/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'
/
```

Beim Binden des UTM-Anwendungsprogramms sieht die Anweisungsfolge zum Einbinden der Laufzeitsysteme wie folgt aus:

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

2.7.2 Shareable Module

Folgende Module können shareable (gemeinsam benutzbar) geladen werden:

- Teilprogramme (genauer: die Code-Module der Pakete, in denen die Teilprogramme implementiert sind)
- Formate
- Die Formatierungsroutine MFHSROUT
- Das Datenbankverbindungsmodul, falls dieses shareable ist
- Das Meldungsmodul KCSMSGS
- Code- und Datenmodule der Pakete, mit denen Sie shareable Datenbereiche definiert haben
- Das Pascal-XT-Laufzeitsystem
  Beachten Sie, daß das Laufzeitsystem einen Externverweis auf ein ILCS-Datenmodul (IT0INITS) enthält, der unbedingt versorgt werden muß (siehe „Beispiel zum Erzeugen von Shared Code“ auf Seite 20).

Die Datenmodule dürfen Sie nur dann für shareable erklären, wenn nur lesender Zugriff erlaubt ist. Um Module shareable zu laden, gibt es bei *open*UTM folgende Möglichkeiten:

- Shareable Module werden in den Klasse 3/4 Speicher geladen (bis einschließlich BS2000/OSD V2.0 - mit ADD-SHARED-PROGRAM).
- Shareable Module werden als nicht priviligiertes Subsystem geladen.
- Shareable Module werden in den Common Memory Pool im Benutzerspeicher (Klasse 6 Speicher) geladen (siehe auch *open*UTM-Handbuch „Konzepte und Funktionen“).

Shareable Teile des Pascal-XT-Laufzeitsystems können Sie als Subsystem oder in einen von UTM generierten Common Memory Pool laden (siehe „Beispiel zum Erzeugen von Shared Code“ auf Seite 20).

Der Pascal-XT-Compiler erzeugt bei der Übersetzung eines Pakets stets drei Module, deren Namensgebung unter der Überschrift „Regeln zur Bildung von Paketnamen“ auf Seite 12 beschrieben ist. Im einzelnen sind dies folgende Module:

- Ein Code-Modul, z.B. mit dem Namen ECHO##C, das immer shareable und reentrantfähig ist.
- Ein Datenmodul, z.B. mit dem Namen ECHO##D, das **nicht** reentrant-fähig ist. Dieses enthält auch die (auf 8 Zeichen gekürzten) Namen der in diesem Paket definierten ENTRY-Prozeduren (Unterstriche in den Namen der ENTRY-Prozeduren werden durch „#“ ersetzt).
- Ein Testtabellenmodul, z.B. mit dem Namen ECHO##T.
  Die Testhilfe PATH ermöglicht es, ein Pascal-XT-Teilprogramm im Dialog symbolisch zu testen. Die Vorgehensweise entnehmen Sie bitte dem Pascal-XT Benutzerhandbuch.

Alle beim Übersetzen des Quellprogramms erzeugten Module müssen Sie in einer Modulbibliothek ablegen, auch die Teile der Teilprogramme, die nicht shareable sind.

Beispiel zum Erzeugen von Shared Code

Das folgende Beispiel soll verdeutlichen, wie Sie für Teilprogramme und für das Pascal-XT-Laufzeitsystem Shared Code erzeugen. Das Beispiel enthält zwei Varianten, eine Variante nutzt die BLS-Schnittstelle (Binder, Lader, Starter), die andere Variante arbeitet ohne BLS.

In der „Prozedur zum Erzeugen von Shared Code“ auf Seite 23 wird für das Teilprogramm „ECHO“ Shared Code erzeugt. Das Teilprogramm enthält zwei Einsprungpunkte (ECHOSYN und ECHOASYN).

Neben dem Teilprogramm wird eine Assembler-Source PASDUMMY benötigt, um den Externverweis auf die ILCS-Prozedur IT0INITS im Pascal-Laufzeitsystem zu versorgen. Die Assembler-Source PASDUMMY besteht aus folgenden Zeilen:

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

UTM liefert diese Assembler-Source in der Bibliothek SYSLIB.UTM.040.EXAMPLE aus.

Der Shared Code (BLS: Public Slice von PASCAL-XT-PROGRAM-UNITS-RTS;
Nicht-BLS: Modul PASHARED) wird in einen Common Memory Pool „MPOOL001“ geladen.

Abhängig davon, ob die BLS-Schnittstelle genutzt wird oder nicht, sieht das KDCDEF-Generierungsteilstück folgendermaßen aus:

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
```

Nicht-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 einer Generierung, die die BLS- Schnittstelle nicht unterstützt, muß der ENTRY IP@#RT2C immer angegeben werden.

Die Namen „&(PRIVATE-MODULE-LIB)“ und „&(SHARED-MODULE-LIB)“ entsprechen denen in der nachfolgenden Prozedur.

Nach dem KDCDEF-Generierungslauf wird die Assembler-Source PASDUMMY assembliert. Anschließend werden die Pascal-Sourcen compiliert.

Die hierdurch gewonnenen Objekte werden nun gebunden, das Ergebnis ist abhängig davon, ob BLS genutzt wird:

BLS:

Die Module werden zu einem LLM (Link and Load Module) namens PASCAL-XT-PROGRAM-UNITS-RTS gebunden, das in eine Public und eine Private Slice unterteilt ist.

Nicht-BLS:

Die Module werden zu einem Private-Teil (PAPRIVAT) und einem Shared-Teil (PASHARED) mit den dazugehörigen Laufzeitsystemmodulen gebunden.
Zum Private-Teil der Laufzeitsystemmodule (alle IP@*xxxx*D, *xxxx* = beliebig) wird noch „ECHO###D“ der Beispiel-Source sowie der Datenteile der dazugehörigen UTM-Pascal-Pakete gebunden. Der Shared-Teil (PASHARED) besteht aus folgenden Komponenten:

– Den Code-Laufzeitsystemmodulen (alle IP@*xxxx*C , *xxxx* = beliebig)

– Laufzeitsystemfunktionen zu einigen mathematischen Routinen (alle IML*xxxx* , *xxxx*=beliebig)

– Den Shared-Teilen des Teilprogramms (ECHO###C , KCKBL##C)

– Dem Modul IT0ENTR aus der Bibliothek SYSLNK.CRTE.IT0ENTR.
Die Version des IT0ENTR muß zu dem gebundenen CRTE passen.

Die CSECTs IT0ENTR, IT0INITS, IMLSINI und IMLEND müssen für nachfolgende Bindeoperationen unsichtbar gemacht werden, damit beim dynamischen Binden keine Duplikate gemeldet oder Externverweise falsch aufgelöst werden. Dies erreicht man für Bindemodule, indem man die CSECTs nach dem Binden mit Hilfe des LMS umbenennt.
Bei LLMs reicht es, die CSECTs mit Hilfe des Binders unsichtbar zu machen.

Bemerkungen

Bei Nutzung der BLS-Schnittstelle ist das Zusammenbinden der Teilprogramme mit dem Pascal-XT-Laufzeitsystem nicht notwendig. Alle Pascal-XT-Teilprogramme müssen nur dynamisch (ggf. in anderen Lademodulen) zum UTM-Anwendungsprogramm gebunden werden.
Gibt es ein Pascal-XT-Laufzeitsystem-Lademodul und weitere Lademodule mit Pascal-XT-Modulen, so müssen Sie in der Generierung das Lademodul mit dem Pascal-XT-Laufzeitsystem als erstes im KDCDEF-Input definieren. Das Lademodul mit dem Pascal-XT-Laufzeitsystem darf nicht im laufenden Betrieb ausgetauscht werden.

Wenn Sie nicht mit BLS arbeiten, müssen alle Module der Pascal-XT-Teilprogramme des UTM-Anwendungsprogramms entweder zum Private-Teil (nicht shareable Module) oder zum Shared-Teil (shareable Module) gebunden werden.
Wird das UTM-Anwendungsprogramm mit dem Binder TSOSLNK und der TSOSLNK-PROGRAM-Anweisung gebunden, so dürfen die nachgebundenen Module nur folgende V-Konstanten offen lassen, deren jeweils passenden CSECTs und ENTRYs sich im ROOT-Code befinden: KDCS, KDCFHS, KDCSCUR, KDCDATF und KDCERRE.
Wollen Sie eine weitere CSECT oder ENTRY nutzen, z.B. ein statisch gebundenes Assembler-Unterprogramm, so müssen Sie statt der TSOSLNK-PROGRAM-Anweisung die MODULE-Anweisung verwenden und das UTM-Anwendungsprogramm mit folgenden Anweisungen starten:

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

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

Prozedur zum Erzeugen von Shared Code

Die folgende Prozedur ist so gestaltet, daß Sie man wahlweise mit oder ohne BLS-Schnittstelle arbeiten kann.

```
/SET-PROCEDURE-OPTIONS                                    -
/       ,DATA-ESC             = STD                       -
/       ,IMPLICIT-DECLARATION = NO
/
/ " ------------------------------------------------------------------ "
/ "                                                                    "
/ " Procedure:  MAKE-PASCAL-SHARED-CODE                                "
/ "                                                                    "
/ "                                                                    "
/ " Purpose:                                                           "
/ "   This procedure generates one private and shareable part by       "
/ "   an 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 Formatierung

Formaterstellung mit dem IFG

Wie Sie Formate mit dem IFG erstellen können, ist ausführlich im IFG-Handbuch beschrieben. Wenn diese Formate für den Einsatz mit UTM erstellt werden, so beachten Sie bitte folgendes:

– Der Formatname darf höchstens 7 Zeichen lang sein

– Im Benutzerprofil wählen Sie die „Struktur des Datenübergabebereichs“
  – für #Formate: getrennte Attributblöcke und Feldinhalte
  – für *Formate: nicht ausgerichtet, ohne Attributfelder
  – für +Formate: nicht ausgerichtet, mit Attributfeldern

  Adressierungshilfen für Pascal-XT können nur "nicht ausgerichtet" erstellt werden.

– Vereinbaren Sie *nur eine* Adressierungshilfe.

  Beispiel

  Das folgende Beispiel zeigt, wie Sie die vom IFG erstellten Adressierungshilfen einsetzen:

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

  Dabei ist FORMA der mit IFG festgelegte Formatname und T_ das Namenspräfix für die Typdeklaration. Beim Einsatz dieses Formats geben Sie beim MPUT- bzw. FPUT- oder DPUT-Aufruf den Formatnamen im Feld KCMF an als „*FORMA“ (bei Adressierungshilfen ohne Attributfelder) bzw. als „+FORMA“ (bei Adressierungshilfen mit Attributfeldern).

– Bitte beachten Sie bei der Definition der Adressierungshilfen, daß UTM zu Vorgangsbeginn beim MGET bzw. FGET den Transaktionscode aus der Nachricht entfernt, sofern dies nicht mit einem INPUT-Exit verhindert wird. Wenn das erste Feld im Format den Transaktionscode enthält, so müssen Sie dies bei den Adressierungshilfen für Eingabeformatierung berücksichtigen. Sie haben folgende Möglichkeiten:

  - mit dem IFG ein eigenes Format für die Eingabe definieren, das das TAC-Feld nicht enthält (für den MGET-Aufruf müssen Sie natürlich den Namen des vollständigen Formats in das Parameterfeld eintragen)
  - unterschiedliche Varianten für die Struktur des Nachrichtenbereichs deklarieren.

- Bei der Einsatzvorbereitung bringen Sie die Formate in die Formateinsatzdatei (Formatbibliothek). Diesen Namen geben Sie bei den FHS-Startparametern an.

- Der IFG erzeugt bei der Einsatzvorbereitung für jedes Format ein Paket, das den gleichen Namen hat wie das Format. Die Spezifikation enthält die Typdeklaration für die Struktur des Formats; die Implementierung (PACKAGE BODY) ist leer.
  Beide Teile müssen Sie vor der Verwendung übersetzen, damit der Name des Pakets in Ihrer Projektdatei bekannt ist. Die erzeugten Module müssen Sie in Ihre Modulbibliothek bringen.
  Nach einer Modifikation eines Formates ist keine Neuübersetzung notwendig (da das Paket nur eine Typdeklaration enthält, wird nur der Standard-Code für die Initialisierung eines Pakets erzeugt).

  Hinweise

  - IFG verwendet für die Attributfelder den Datentyp T_FIELD_ATTRIBUTE_SET.
    Er ist in dem Paket FIELD_ATTRIBUTE_PACKAGE deklariert, das zusammen mit UTM ausgeliefert wird.

  - Die übrigen Felder eines Formats werden als PACKED ARRAY [1..n] OF CHAR definiert (n = Feldlänge).
    Sie können PACKED ARRAY durch ARRAY ersetzen. Dann können die Standardprozeduren PACK und UNPACK für den Transport von Daten aus den Feldern bzw. in diese Felder verwendet werden

  - IFG verwendet die folgenden Namenskonventionen, die Sie nicht ändern können:

```
T_formatname       für die Typdeklaration eines Formats,
A_feldname         für die Attributfelder,
DUMMY_OOOn         für unbenannte, dem Programm zugängliche Felder.
```

Modifizieren von KDCS-Attributen

Zur Unterstützung der Programmierung stellt UTM in dem Paket FIELD_ATTRIBUTE_PACKAGE alle unterstützten Kombinationen zur Verfügung. Wird in einem Attributfeld X'0000' angegeben, so werden die Attribute von der Formaterstellung genommen.

Positionieren des Cursors

UTM bietet Ihnen zwei Möglichkeiten für die Positionierung der Schreibmarke bei der Ausgabe eines Formates. Die Auswahl treffen Sie mit dem FHS-Startparameter CURSOR=ATTR bzw. CURSOR=NOATTR (siehe „FHS-Handbuch").

Bei CURSOR=NOATTR müssen Sie die Adressierungshilfen Ihrer Formate ändern. In den Deklarationen der Felder, die Sie für die Schreibmarkenpositionierung verwenden wollen, müssen Sie PACKED ARRAY durch ARRAY ersetzen. Beachten Sie dabei die Regeln für die Zuweisungsverträglichkeit von Zeichenkettentypen in Pascal-XT. Verwenden Sie dann die Standardprozeduren PACK und UNPACK für den Datentransport.

Sie deklarieren folgende Prozedur:

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

Dann können Sie durch folgenden Aufruf die Schreibmarke an jede beliebige Stelle von Ausgabefeldern setzen:

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

„feldname" ist ein in Ihrem Format deklariertes Feld; „i" muß innerhalb der Array-Grenzen liegen.

Erweiterter Zeilenmodus

Bei der Verwendung von Terminals im Zeilen-Modus können Sie die Ausgabenachricht mit logischen Steuerzeichen strukturieren.
Erlaubt sind alle Steuerzeichen der Zugriffsmethode TIAM im Zeilenmodus.

Für die Arbeit im erweiterten Zeilenmodus müssen sie die Steuerzeichen selbst definieren. Welche Steuerzeichen es gibt und welche Werte sie haben müssen, finden Sie im TIAM-Benutzerhandbuch, z.B. bei der Beschreibung des Makros VTCSET. Ein Beispiel dafür, wie Sie die Steuerzeichen selbst definieren, finden Sie auf Seite 130.

3 Beispiele in Pascal-XT

In diesem Kapitel finden Sie sowohl einfache Beispiele zur Codierung eines KDCS-Aufrufs als auch ein Beispiel für eine komplette UTM-Anwendung einschließlich der KDCDEF-Generierung.

3.1 Beispiele zu einzelnen KDCS-Aufrufen

Dieser Abschnitt enthält Codierbeispiele für folgende KDCS-Aufrufe:

– MGET
– MPUT
– DPUT
– APRO mit MPUT bei verteilter Verarbeitung

Da die übrigen KDCS-Aufrufe auf analoge Weise codiert werden, wird an dieser Stelle auf eine explizite Darstellung verzichtet.

In den Beispielen werden für die Datenstrukturen und Konstanten die in den vordefinierten Paketen definierten Namen benutzt (siehe Seite 5 und Kapitel „Datenstrukturen für Pascal-XT“). Die Namen der anwendungsspezifischen Strukturen orientieren sich an den Beispielen ab Seite 9 (*kb* für den Kommunikationsbereich; *kbk* für dessen Kopf; *spab* für den Standard Primären Arbeitsbereich; *kcpal* für den Parameterbereich; *nb* für den Nachrichtenbereich).

MGET-Aufruf

Ein laufender Vorgang kann durch eine Eingabe, die aus einer Kurznachricht besteht, unterbrochen werden. Die Kurznachricht wird mit der Funktionstaste F2 erzeugt und besteht aus weiteren Daten von 10 Zeichen. Die Eingabe soll eine Sonderfunktion auslösen.

```
...
...
INLINE PROCEDURE SONDER_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 SONDER_MGET;
   IF KB.KBK.KCRCCC <> '000' THEN MGET_FEHLER;
        .
```

1) Für die 10 Zeichen ist ein weiterer MGET erforderlich.

2) Eine Sonderfunktion wird abgefragt.

MPUT-Aufruf

1. Eine unformatierte Nachricht von 80 Bytes soll gesendet werden.

```
   .
   .
   .
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_FEHLER;
```

2. Die letzte Nachricht (in diesem Beispiel formatiert und 500 Bytes lang) in einem Vorgang soll an ein Format-Terminal geschickt werden. Der Name des Formats ist „*BILD15“. Der Bildschirm soll vorher gelöscht werden.

```
   .
   .
   .
WITH SPAB.KCPAL DO BEGIN
  KCOP := KCKBL.MPUT;
  KCOM := 'NE';
  KCLM := 500;
  KCRN := '        ';
  KCMF := '*BILD15 ';
  KCDF := KCDFL.KCREPL;          1)
END;
WITH SPAB DO KDCS (KCPAL, NB);
IF KB.KBK.KCRCCC <> '000'
      THEN MPUT_FEHLER;
```

1) REPLACE wird bei Formatwechsel standardmäßig ausgeführt. Die Ausgabe erfolgt, um Fehler wegen undefinierter Feldinhalte auszuschließen.

3. In einem *Format „BILD10“, das laut letzter Eingabe am Terminal noch vorhanden ist, sollen als Antwort alle variablen d.h. überschreibbaren Felder gelöscht werden. Die geschützten Felder sollen erhalten bleiben.

```
   .
   .
   .
WITH SPAB.KCPAL DO BEGIN
  KCOP := KCKBL.MPUT;
  KCOM := 'NE';
  KCLM := ZEROES;
  KCRN := SPACES;
  KCMF := '*BILD10 ';
  KCDF := KCDFL.KCERAS;
END;
WITH SPAB DO KDCS (KCPAL, NB);
IF KB.KBK.KCRCCC <> 'OOO'
     THEN MPUT_FEHLER;
```

DPUT-Aufruf

1. Ein Asynchron-Auftrag mit einer Nachricht von 11 Zeichen soll am 11.11. (= 315. Tag im Jahr) um 11.11 Uhr an ein Folgeprogramm gesendet werden (absolute Zeitangabe). Der TAC lautet „ALAAF“.

```
      .
      .
      .
with spab.kcpal do begin
    kcop := 'DPUT';
    kcom := 'NE';
    kclm := 11;
    kcmf := '        ';
    kcdf := 0;
    kcrn := 'ALAAF   ';
    kcmod := 'A';
    kctag :='315'; kcstd := '11'; kcmin := '11'; kcsek := '00';
end;
with spab do kdcs (kcpal, nb);
if kb.kbk.kcrccc <> '000' then dput_fehler;
```

2. Eine Asynchron-Nachricht von 80 Zeichen soll nach einer Stunde an das Terminal „DSS1“ gesendet werden (relative Zeitangabe). Dabei soll die Bildschirmfunktion „akustischer Alarm“ (BEL) ausgelöst werden.

```
      .
      .
      .
with spab.kcpal do begin
    KCOP := kckbl.DPUT;
    KCOM := 'NE';
    KCLM := 80;
    KCRN := 'DSS1    ';
    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_fehler;
```

APRO-Aufruf mit MPUT bei verteilter Verarbeitung

Vom Auftraggeber-Vorgang aus soll der Dialogvorgang mit dem Transaktionscode „LTAC1“ der Auftragnehmer-Anwendung „PARTNER1“ adressiert werden (zweistufige Adressierung). Dabei soll dem Auftragnehmer-Vorgang die Vorgangsidentifikation „>VGID1“ zugeordnet werden. Anschließend wird eine MPUT-Nachricht mit Länge 100 im Zeilenmodus an die Partneranwendung geschickt.

```
        .
        .
with spab.kcpal do begin
   kcop := kckbl.apro; kcom := 'DM'; kclm := 0;
   kcrn := 'LTAC1   ';
   kcpa := 'PARTNER1';
   kcpi := '>VGID1  ';
end;
kdcs (kcpal ...);
if kb.kbk.kcrccc <> '000'
then apro_fehler;
        .
        .
        .
with spab.kcpal do begin
  kcop := kckbl.mput;
  kcom := 'NE';
  kclm := 100;
  kcrn := '>VGID1  ';
  kcmf := '        ';
  kcdf := 0;
end;
with spab do kdcs (kcpal, nb);
if kb.kbk.kcrccc <> '000'
then mput_fehler;
```

3.2 Beispiel für ein Asynchron-Teilprogramm MSGTAC

Das MSGTAC-Teilprogramm DASCHU soll verhindern, daß nicht berechtigte Benutzer sich an eine UTM-Anwendung anschließen. Wenn mehr als dreimal an einem Anschlußpunkt (LTERM-Partner) ein KDCSIGN-Versuch mit ungültigem Benutzer, falschem Kennwort oder falschem Ausweis versucht wird, soll die Verbindung zu diesem Terminal abgebaut werden.

Zu den Vorbereitungen siehe auch Basishandbuch „Anwendungen programmieren mit KDCS für COBOL, C und C++“.
UTM ist in diesem Beispiel unter der Benutzerkennung $UTM installiert.

Realisierung des MSGTAC-Teilprogramms

Das MSGTAC-Teilprogramm DASCHU zählt die Anzahl der Fehlversuche in einem TLS. Wenn UTM ein KDCSIGN akzeptiert (d.h. Meldung K008 oder K033), so wird dieser TLS wieder gelöscht.
Falls nach drei ungültigen KDCSIGN-Versuchen der 4. KDCSIGN-Versuch wieder fehlerhaft ist, so soll das entsprechende Terminal über „asynchrone Administration“ diskonnektiert werden. Dies geschieht mit dem Inhalt „PTERM=*pterm*, PRO=*proname*, ACT=DIS“ (siehe auch *open*UTM-Handbuch „Anwendungen generieren und betreiben“).
Das Administrationskommando wird dann mit LPUT in der Benutzer-Protokolldatei protokolliert und der TLS gelöscht.

Die K-Meldungen werden jeweils mit FGET vom MSGTAC-Teilprogramm gelesen. Nach der „Verarbeitung“ einer K-Meldung wird mit FGET sofort die nächste K-Meldung gelesen, innerhalb desselben Teilprogrammlaufs.

Paket-Spezifikation

```
with kckbl;
from kckbl use
   kckb, kcpal, redefines;
with kcmsl;
from kcmsl use
   kcmsgl;
package DASCHU_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; (* Programmbereich; wird nicht verwendet *)
   end;

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

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

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

end.
```

Paket-Implementierung

```
with kckbl;
from kckbl use
   kckb, kcpal;
with kcmsl;
from kcmsl use
   kcmsgl;
package body DASCHU_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 DASCHU (var kb: kckbc; var spab: kcspab);

var
   fehlerzeile : string;
   pt,
   pp : pchar8;

   procedure arbeit;
      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_anz);
         kcrn := hacker_lterm;
         kdcs (pf, ccc);
         if kcrccc <> '000'
         then return;
         if kcrlm = 0
         then begin
            if not ((msgnr = 'K008') or (msgnr = 'K033'))
            then begin (* sonst kein TLS vorhanden *)
               kcop := 'PTDA';
               kcla := sizeof (hack_anz);
               hack_anz := 1;
               kcrn := id_hack_tls;
               kclt := hacker_lterm;
               kdcs (pf, ccc);
            end;
         end
         else begin
            if ((msgnr = 'K008') or (msgnr = 'K033'))
```

```
then begin (* ok; TLS löschen *)
   kcop := 'PTDA';
   kcla := 0;
   kcrn := id_hack_tls;
   kdcs (pf, ccc);
end
else begin (* pruef_anz*)
   hack_anz := hack_anz + 1;
   if hack_anz <= hack_max
   then begin (* darf weiter üben *)
      kcop := 'PTDA';
      kcla := sizeof (hack_anz);
      kcrn := id_hack_tls;
      kclt := hacker_lterm;
      kdcs (pf, ccc);
      return;
   end
   else begin (* DISCONNECT !! *)
      if msgnr = 'K004'
      then with k004
      do begin
         pt := ptrm; pp := prnm;
      end
      else
      if msgnr = '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*)  (* Protokoll auf USER-LOGGING *)
```

```
                kcop := 'LPUT';
                kcla := length (str);
                kdcs (pf, ccc);
                if kcrccc <> '000'
                then return;
             (*P_PTDA*)  (* TLS löschen *)
                kcop := 'PTDA';
                kcla := 0;
                kcrn := id_hack_tls;
                kclt := hacker_lterm;
                kdcs (pf, ccc);
             end (*disconnect*)
          end (* pruef_anz*);
       end;
   end (*with*);
end (*arbeit*);
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 msgnr = 'K004' (* ungültige Benutzerkennung *)
         then hacker_lterm := k004.ltrm
         else
         if msgnr = 'K006' (* ungültiges Kennwort *)
         then hacker_lterm := k006.ltrm
         else
         if msgnr = 'K008' (* kdcsign akzeptiert *)
         then hacker_lterm := k008.ltrm
         else
         if msgnr = 'K031' (* falscher Ausweis *)
         then hacker_lterm := k031.ltrm
         else
         if msgnr = 'K033' (* Startformat *)
         then hacker_lterm := k033.ltrm
         else begin
            spab.pf.kcop := msgnr;
            raise (go_pend_rset);
         end;
      end;
      arbeit;
      if kb.kbk.kcrccc <> '000'
      then raise (go_pend_rset);
      fget := true; (* Meldung vorhanden *)
   end (*fget*);
   procedure kdcs (var pf: kcpal); cobol;

   procedure fmeld (meldung: string);
      procedure kdcs (var pf: kcpal; var msg: char80); cobol;

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

begin (*DASCHU*)
   init;
   while fget do ; (* solange wie Meldungen vorhanden sind *)
   (*PEND_ANF*)
   with spab.pf
   do begin
```

```
      kcop := 'PEND';
      kcom := 'FI';
   end;
   kdcs (spab.pf);

exception (* Behandlung der Fehlerfälle *)
   with spab, spab.pf, kb.kbk
   do begin
      case error_number of
         go_pend_rset:
            begin
               (*PEND_RSET*)
               writestring (fehlerzeile,'FEHLER IM TEILPR. DASCHU; ',
                                        'VORG./TAC   ', kctacvg,
                                        '/', kctacal, ' WG. ', kcop,
                                        '  (RC: ', kcrccc, kcrckz, kcrcdc,
                                        ')');
               kcop := 'RSET';
               kdcs (spab.pf);
               (*PEND_RSET_LPUT*)
               fmeld (fehlerzeile);
               (*PEND_RSET_ANF*)
               kcop := 'PEND';
               kcom := 'FI';
               kdcs (spab.pf);
            end;
         else : raise (error_number); (* Laufzeitfehler etc. *)
      end;
   end;
end (*DASCHU*);

begin (*evaluation*)
end.
```

3.3 Beispiel für eine komplette UTM-Anwendung

Das nachfolgende Beispiel für eine komplette UTM-Anwendung behandelt die Adressenverwaltung.

Mit diesem Anwendungsbeispiel können Adreßdaten verwaltet werden, die in einer ISAM-Datei stehen. Die Anwendung stellt dazu die nachfolgenden Funktionen zur Verfügung, die durch Eintrag des jeweiligen TACs in das dafür vorgesehene Feld aufgerufen werden. Die Ein- und Ausgaben erfolgen in einem Format.

TAC	Funktion	
1	Anzeige	Gibt eine in der Datei vorhandene Adresse aus. Suchbegriff (ISAM-Schlüssel) ist dabei der Name und die ersten zwei Buchstaben des Vornamens, welche in den zugehörigen Feldern anzugeben sind.
2	Neueintrag	Trägt eine neue Adresse in die Datei ein. Eine Adresse mit dem gleichen Suchbegriff (s.o.) darf dort nicht schon vorhanden sein.
3	Ändern	Ändert einen Adresseintrag. Die Adresse muß in der Datei schon vorhanden sein.
4	Löschen	Löscht eine in der Datei vorhandene Adresse.

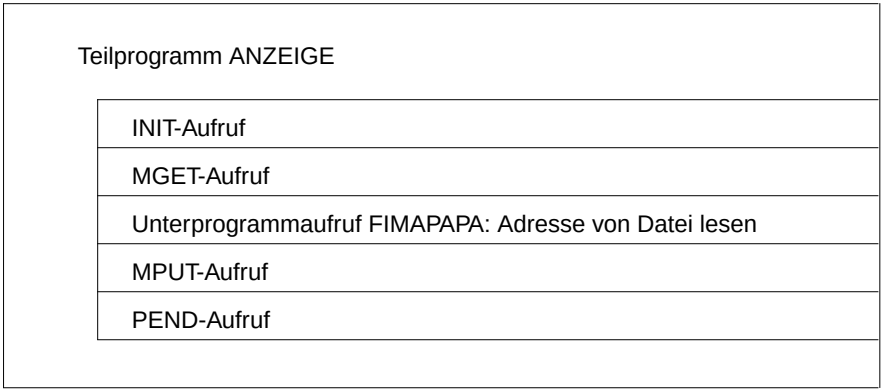
Bei Fehlbedienung erscheint in der untersten Zeile des Formats eine Fehlermeldung.

Die oben genannten Ziffern sind die Transaktionscodes (TACs), die die Anwendung steuern. Dabei rufen der Tansaktionscode 1 das Teilprogramm ANZEIGE auf und die Transaktionscodes 2, 3 und 4 das Teilprogramm AENDERN. Diese Teilprogramme verzweigen dann jeweils in das Teilprogramm FIMAPAPA. Dieses Teilprogramm wird als START- und SHUT-Event-Exit eingesetzt und enthält die Unterprogramme, die die Ein-/Ausgaben auf die Adreßdatei durchführen.
Das Teilprogramm BADTACS wird von UTM automatisch aufgerufen, wenn ein ungültiger TAC eingegeben wird. Nach dem Aufbau der Verbindung mit der Anwendung und erfolgtem KDCSIGN gibt UTM sofort das Format aus (Startformat). Die Arbeit mit dem Benutzer erfolgt dann im strengen Dialog, d.h. auf die Eingabe eines TACs und des ISAM-Schlüssels reagiert die Anwendung mit der Ausgabe des Formats, das die gesuchte Adresse enthält bzw. mit einer Erfolgs- oder einer Fehlermeldung in der untersten Zeile.

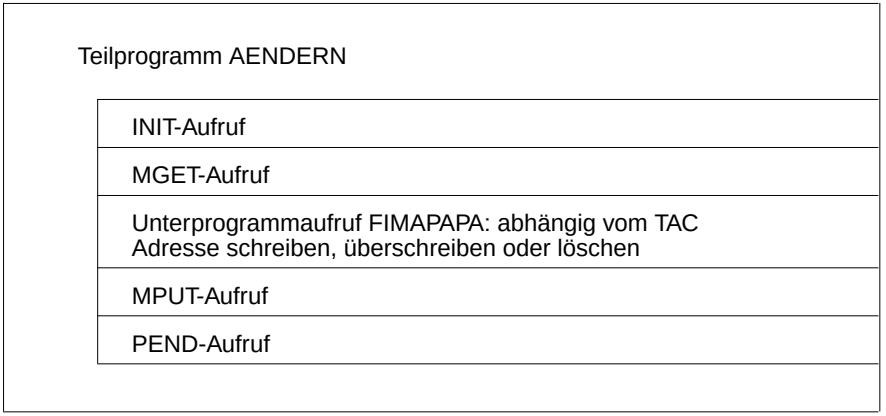
Hinweis

Dieses Programm soll zeigen, wie man mit UTM programmiert. Die ISAM-Dateizugriffe sind nicht über das UTM-Transaktionskonzept gesichert. Für eine „echte“ Anwendung verwenden Sie am besten ein Datenbanksystem oder LEASY. In diesem Beispiel haben wir darauf verzichtet, um das Beispiel nicht mit Datenbank-spezifischen Programmteilen zu belasten.

Die folgenden Struktogramme zeigen den Aufbau der Teilprogramme:



Struktogramm des Teilprogramms ANZEIGE



Struktogramm des Teilprogramms AENDERN

Der Vollständigkeit halber ist im Anschluß an das Pascal-XT-Programm die Generierung dieser Anwendung aufgeführt. Die genaue Bedeutung der einzelnen Operanden und Anweisungen entnehmen Sie bitte dem *open*UTM-Handbuch „Anwendungen generieren und betreiben“.

Das folgende Bild zeigt das Format, das bei dieser Anwendung verwendet wurde:

```
********************************************************************************
                              A d r e s s e n v e r w a l t u n g
********************************************************************************
        Bitte waehlen Sie eine Funktion aus: ________
--------------------------------------------------------------------------------
        Aktuelle Funktion:  nnnnnnnnnnnnnnnnnnnnnnnnn


        Name:_____________                      Vorname:__________________

        Strasse:___________________________     Nr:_________

        Postleitzahl: ####                      Ort: _______________________

        Telefon:_________________

--------------------------------------------------------------------------------
                                   Funktionsauswahl
     1  =  Anzeige von Adressen             |        4  =  Loeschen von Adressen
     2  =  Neueintrag von Adressen          |
     3  =  Aendern von Adressen             |              Beenden mit kdcoff
--------------------------------------------------------------------------------
nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn

```

*-Format „FORMA“, mit dem diese Anwendung arbeitet.

Auf den folgenden Seiten finden Sie die IFG-Attributliste für das *-Format „FORMA“ sowie die für das Format erzeugte Paket-Spezifikation und -Implementierung.

IFG-Attributliste für das *-Format „FORMA“

```
POSITION
ZL SP  FELDNAME          LAENGE ATTRIBUTE
                         ( (*) KENNZEICHNET ABWEICHUNGEN VON DEN
                               BENUTZERPROFILANGABEN )
01 01                    80  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
02 23                    35  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
03 01                    80  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
04 07                    37  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
04 44  TAC               08  EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                             ZUGAENGLICH
                             NUR GROSSBUCHSTABEN
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
05 01                    80  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
06 07                    20  TEXTFELD, GESCHUETZT, HELL, DEM PROGRAMM NICHT
                             ZUGAENGLICH
                             KURSIV
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                             (*)
06 27  FUNKTION          26  AUSGABEFELD, GESCHUETZT, HELL, DEM PROGRAMM
                             ZUGAENGLICH
                             KURSIV
                             AUTOMATISCHE EINGABE
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                             (*)
09 07                    05  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                             NICHT ZUGAENGLICH
                             AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                             FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
09 12  NAME              14  EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
```

```
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
09 43                          08   TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
09 51  VNS                     02   EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                                    GRUPPENANFANG   NAME:  VORNAME
09 53                          18   EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                                    GRUPPENENDE
11 07                          08   TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
11 15  STRASSE                 26   EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
11 41                          05   TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
11 46  NR                      10   EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
13 07                          13   TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '

13 20  PLZ                     04   EINGABEFELD-NUMERISCH, UNGESCHUETZT, HELL, DEM
                                    PROGRAMM ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: RECHTS/RECHTS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                                    (*)
13 43                          04   TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
```

```
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
13 47  ORT                  24  EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
15 07                       08  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
15 15  TEL                  16  EINGABEFELD, UNGESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: LINKS/LINKS
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                                    (*)
17 01                       80  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
18 32                       16  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
19 05                       64  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
20 05                       36  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
21 05                       61  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
22 01                       80  TEXTFELD, GESCHUETZT, HALBHELL, DEM PROGRAMM
                                    NICHT ZUGAENGLICH
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
23 01  MELDUNGSTEXT         80  AUSGABEFELD, GESCHUETZT, HELL, DEM PROGRAMM
                                    ZUGAENGLICH
                                    NUR GROSSBUCHSTABEN
                                    AUSRICHTUNG  BEI EINGABE/AUSGABE: KEINE/KEINE
                                    FUELLZEICHEN BEI EINGABE/AUSGABE: ' '/' '
                                    (*)
```

Paket-Spezifikation und -Implementierung

Nachfolgend ist die Paket-Spezifikation und -Implementierung abgedruckt, die von IFG für das Format erzeugt wird.

```
PACKAGE FORMA   ;

(* USER AREA LENGTH: 0228 *)
TYPE T_FORMA    =
  RECORD
    TAC                             (0000) : PACKED ARRAY
                                             (.01..08.) OF CHAR;
    FUNKTION                        (0008) : PACKED ARRAY
                                             (.01..26.) OF CHAR;
    NAME                            (0034) : PACKED ARRAY
                                             (.01..14.) OF CHAR;
    VORNAME                         (0048) :
      RECORD
        VNS                             (0000) : PACKED ARRAY
                                                 (.01..02.) OF CHAR;
        DUMMY_0001                      (0002) : PACKED ARRAY
                                                 (.01..18.) OF CHAR;
      END;
    STRASSE                         (0068) : PACKED ARRAY
                                             (.01..26.) OF CHAR;
    NR                              (0094) : PACKED ARRAY
                                             (.01..10.) OF CHAR;
    PLZ                             (0104) : PACKED ARRAY
                                             (.01..04.) OF CHAR;
    ORT                             (0108) : PACKED ARRAY
                                             (.01..24.) OF CHAR;
    TEL                             (0132) : PACKED ARRAY
                                             (.01..16.) OF CHAR;
    MELDUNGSTEXT                    (0148) : PACKED ARRAY
                                             (.01..80.) OF CHAR;
  END;

END.



PACKAGE BODY FORMA   ;
BEGIN
END.
```

Hinweis

Spezifikation und Implementierung der verwendeten Formate müssen mit dem PASCAL-XT Compiler übersetzt und die erzeugten Module (Code- und Datenmodul) in die Bindemodul-Bibliothek der UTM-Anwendung kopiert werden, bevor sie für die Übersetzung der Teilprogramme verwendet werden können.

Paket-Spezifikation für das Beispiel Adressenverwaltung

```
with kckbl;
from kckbl use kckb, kcpal, redefines;
with forma;
from forma use t_forma;
package BEISPIEL_1 (adrdatei);

type

   (* Struktur eines Satzes der Adreßdatei *)

   id_adressatz = record
      nachname (000) : packed array [1..14] of char;
      vns      (014) : packed array [1..02] of char;
      vnrest   (016) : packed array [1..18] of char;
      strasse  (034) : packed array [1..26] of char;
      nr       (060) : packed array [1..10] of char;
      plz      (070) : packed array [1..04] of char;
      ort      (074) : packed array [1..24] of char;
      tel      (098) : packed array [1..16] of char;
   end;


   (* Struktur des Nachrichtenbereiches *)

   id_data = packed array [1..220] 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 : (adresse     (034) : id_adressatz);
         v5 : (meldung     (148) : array [1..80] of char);
         v6 : (chars       (000) : array [1..228] of char);
         else: ();
   end;


   (* Struktur des Kommunikationsbereiches *)

   id_kdckb = record
      kbkopf : kckb;                    (* KB-Kopf *)
      kbprg  : id_nb;                   (* Programmbereich *)
   end;
   (* Struktur des Standard Primären Arbeitsbereiches *)
```

```
   id_spab = record
      kcpal : kcpal;                    (* Parameterfeld *)
      nb   : id_nb;                    (* Nachrichtenbereich *)
   end;


   (* Beschreibung der (globalen) Datei "adrdatei" *)

   type
      id_adrdatei = file of id_adressatz;

   var
      adrdatei : id_adrdatei;

      (* Hinweis
            Die Angaben zur Dateiorganisation (ISAM, Position und
            Länge des Schlüssels, Satzformat) sind nicht Bestand-
            teil von PASCAL-XT; sie müssen daher durch ein
            /FILE-Kommando und/oder durch die Standardprozedur
            assignfile  definiert werden *)



   (* Teilprogramm ANZEIGE *)

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


   (* Teilprogramm AENDERN *)

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


   (* Teilprogramm FIMAPAPA *)

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


   (* Teilprogramm BADTACS *)

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

end (*BEISPIEL_1*).
```

Paket-Implementierung für das Beispiel Adressenverwaltung

Im folgenden Beispiel ist zur Demonstration des Paketkonzepts in PASCAL-XT das (nur geringfügig modifizierte) Übersetzungsprotokoll abgedruckt.

```
*** 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, BEISPIEL-S(*STD,S))


CURRENT COMPILATION UNIT (SOURCE FILE)

               ($xy.TUTM.SRCLIB, BEISPIEL-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;
```

```
10      from bs2000calls use cmd;
11      package body BEISPIEL_1 (adrdatei);
12
13      type
14         range = 1..sizeof (id_nb);
15         id_spaces = array [range] of char;
16      var
17         space_area: id_spaces;
18      const
19         space_constant = id_spaces(' ':sizeof (id_spaces));
20
21         no_addr = '*** ADRESSE MIT DIESEM NAMEN NICHT VORHANDEN ***';
22         dup_addr = '*** ADRESSE MIT DIESEM NAMEN SCHON VORHANDEN ***';
23
24
25      inline procedure spaces(var c: packed array[lo..hi:range] of
        char);
26      begin
27         pack (space_area, 1, c);
28      end (* spaces *);
29
30
31      inline function hexa (h: integer): string;
32      type
33         t_hexdig = array [0..15] of char;
34      const
35         c_hexdig = t_hexdig ('0','1','2','3','4','5','6','7',
36                              '8','9','A','B','C','D','E','F');
37      var
38         i: 0..7;
39         s: string[8];
40         n: integer;
41      begin
42         n := h + #10000;
43         s := '';
44         for i := 0 to 7
45         do begin
46            n := (n * #10) mod #100000;
47            insert (c_hexdig[(n div #10000)], s, length (s) + 1);
48         end;
49         hexa := s;
50      end (* hexa *);
51
52
53      inline function dvserr (errcode: integer): string;
54      begin
55      dvserr := concat ('*** DATEIFEHLER #', hexa (system_code),
        ' ***');
56      end (* dvserr *);
57
```

```
58
59
60
61      function LESEN (var adresse: id_adressatz): string;
62      begin
63         (* ISAM-Schlüssel versorgen *)
64         adrdatei^.nachname := adresse.nachname;
65         adrdatei^.vns := adresse.vns;
66         (* übrige Felder löschen *)
67         with adresse
68         do begin
69            spaces (strasse);
70            spaces (nr);
71            spaces (plz);
72            spaces (ort);
73            spaces (tel);
74         end;
75         getkey (adrdatei);
76         if nokey (adrdatei)
77         then lesen := no_addr
78         else begin
79            adresse := adrdatei^;
80            lesen := '';
81         end;
82      exception
83         if error_number = file_error
84         then lesen := dvserr (system_code)
85         else raise (error_number);
86      end (* LESEN *);
87
88
89      function SCHREIBEN (var adresse: id_adressatz): string;
90      begin
91         (* prüfen, ob der Eintrag schon existiert *)
92         adrdatei^.nachname := adresse.nachname;
93         adrdatei^.vns := adresse.vns;
94         getkey (adrdatei);
95         if nokey (adrdatei)
96         then begin (* Satz existiert noch nicht *)
97            adrdatei^ := adresse;
98            put (adrdatei);
99            schreiben := '';
100        end
101        else schreiben := dup_addr;
102     exception
103        if error_number = file_error
104        then schreiben := dvserr (system_code)
105        else raise (error_number);
106     end (* SCHREIBEN *);
107
```

```
108
109      function UEBERSCHREIBEN (var adresse: id_adressatz): string;
110      begin
111         adrdatei^.nachname := adresse.nachname;
112         adrdatei^.vns := adresse.vns;
113         getkey (adrdatei);
114         if nokey (adrdatei)
115         then ueberschreiben := no_addr
116         else begin
117            adrdatei^ := adresse;
118            put (adrdatei);
119            ueberschreiben := '';
120         end;
121      exception
122         if error_number = file_error
123         then ueberschreiben := dvserr (system_code)
124         else raise (error_number);
125      end (* UEBERSCHREIBEN *);
126
127
128      function LOESCHEN (var adresse: id_adressatz): string;
129      begin
130         adrdatei^.nachname := adresse.nachname;
131         adrdatei^.vns := adresse.vns;
132         getkey (adrdatei);
133         if nokey (adrdatei)
134         then loeschen := no_addr
135         else begin
136            elim (adrdatei);
137            loeschen := '';
138         end;
139      exception
140         if error_number = file_error
141         then loeschen := dvserr (system_code)
142         else raise (error_number);
143      end (* LOESCHEN *);
144
145
146      (* Teilprogramm FIMAPAPA *)
147
148      procedure FIMAPAPA (var kb: id_kdckb; var spab: id_spab);
149      var
150         filecmd: string;
151         not_ok: boolean;
152      begin
153         if kb.kbkopf.kctacvg = 'STARTUP '
154         then begin
155            writestring (filecmd, '/FILE ADRESSEN,LINK=ADRDATEI,'
156                                  ,'FCBTYPE=ISAM,RECFORM=V,BLKSIZE=STD,'
157                                  ,'KEYPOS=5,KEYLEN=16,DUPEKY=NO');
```

```
158            cmd (filecmd, not_ok);
159            if not not_ok
160            then replace (adrdatei)
161            else raise (999); (* Awendung abbrechen, falls Komm.Feh. *)
162         end;
163         if kb.kbkopf.kctacvg = 'SHUTDOWN'
164         then close (adrdatei);
165      end (* FIMAPAPA *);
166
167      (* gemeinsame Prozeduren für die folgenden Teilprogramme *)
168
169      procedure INIT (var pf: kcpal; var nb: id_nb);
170         procedure kdcs (var pf: kcpal); COBOL;
171      begin
172         with pf
173         do begin
174            kcop := 'INIT';
175            kclkbprg := 0;
176            kclpab := sizeof (id_spab);
177         end;
178         with nb
179         do begin
180            spaces (tac);
181            spaces (data);
182         end;
183         kdcs (pf);
184      end (*INIT*);
185
186
187      procedure MGET (var pf: kcpal; var nb: id_nb);
188         procedure kdcs (var pf: kcpal; var daten: id_data); COBOL;
189      begin
190         with pf
191         do begin
192            kcop := 'MGET';
193            kcla := sizeof (nb);
194            kcmf := '*FORMA  ';
195         end;
196         kdcs (pf, nb.data);
197      end (*MGET*);
198
199      procedure MPUT (var pf: kcpal; var nb: id_nb; len: integer);
200         procedure kdcs (var pf: kcpal; var nb: id_nb); COBOL;
201      begin
202         with pf
203         do begin
204            kcop := 'MPUT';
205            kcom := 'NE';
206            kclm := len;
207            if len = sizeof (t_forma)
```

```
208            then kcmf := '*FORMA  '
209            else spaces (kcmf);
210            spaces (kcrn);
211         end;
212         kdcs (pf, nb);
213      end (*MPUT*);
214
215      procedure PEND (var pf: kcpal; opmod: pic_xx);
216         procedure kdcs (var pf: kcpal); COBOL;
217      begin
218         with pf
219         do begin
220            kcop := 'PEND';
221            kcom := opmod;
222         end;
223         kdcs (pf);
224      end (*PEND*);
225
226      procedure FEHLER (tp: string; f_op: pic_x_4; f_rc: pic_xxx;
         spab: id_spab);
227      var
228         temp: string;
229      begin
230         writestring (temp, '*** F E H L E R ***  TEILPROGRAMM: ', tp:8,
231                            ' OPERATIONSCODE: ', f_op,
232                            ' RETURNCODE: ', f_rc);
233         unpack (temp, spab.nb.chars, 1);
234         spab.kcpal.kcdf := 0; (* keine Bildschirmfunktionen *)
235         mput (spab.kcpal, spab.nb, length (temp));
236         pend (spab.kcpal, 'ER');
237      end (*FEHLER*);
238
239
240
241      (* Teilprogramm ANZEIGE *)
242
243      procedure ANZEIGE (var kb: id_kdckb; var spab: id_spab);
244      var
245         dvrc : string; (* DVS-Fehlermeldung *)
246      begin
247         with spab, kb.kbkopf
248         do begin
249            init (kcpal, nb);
250            if kcrccc <> '000'
251            then fehler ('ANZEIGE', kcpal.kcop, kcrccc, spab);
252            mget (kcpal, nb);
253            if kcrccc <> '000'
254            then fehler ('ANZEIGE', kcpal.kcop, kcrccc, spab);
255            dvrc := lesen (nb.adresse);
```

```
256            if dvrc <> ''
257            then unpack (dvrc, nb.meldung, 1);
258            mput (kcpal, nb, sizeof (id_nb));
259            if kcrccc <> '000'
260            then fehler ('ANZEIGE', kcpal.kcop, kcrccc, spab);
261            pend (kcpal, 'FI');
262         end;
263      end (*ANZEIGE*);
264
265
266
267      (* Teilprogramm  AENDERN *)
268
269      procedure AENDERN (var kb: id_kdckb; var spab: id_spab);
270      label
271         9; (* Fehlerausgang *)
272      var
273         dvrc : string; (* DVS-Fehlermeldung *)
274         bintac: integer;
275      begin
276         with spab, kb.kbkopf
277         do begin
278            init (kcpal, nb);
279            if kcrccc <> '000' then goto 9;
280            mget (kcpal, nb);
281            if kcrccc <> '000' then goto 9;
282            begin
283               readstring (kctacvg, bintac);
284            exception
285               bintac := -1; (* Fehler, falls nicht nur Ziffern u. ZRW *)
286            end;
287            case bintac of
288               2 : dvrc := schreiben (nb.adresse);
289               3 : dvrc := ueberschreiben (nb.adresse);
290               4 : dvrc := loeschen (nb.adresse);
291             else: dvrc := concat('*** UNGÜLT. TAC (', kctacvg, ') *** );
292            end;
293            if dvrc <> ''
294            then unpack (dvrc, nb.meldung, 1);
295            mput (kcpal, nb, sizeof (id_nb));
296            if kcrccc <> '000' then goto 9;
297            pend (kcpal, 'FI');
298
299         9: (* Fehler ; kehrt von FEHLER nicht zurueck *)
300            fehler ('AENDERN', kcpal.kcop, kcrccc, spab);
301         end;
302      end (*AENDERN*);
303
304
305
```

```
306      (* Teilprogramm BADTACS *)
307
308      procedure BADTACS (var kb: id_kdckb; var spab: id_spab);
309      label
310         9; (* KDCS-Fehler *)
311      var
312         temp: string;
313      begin
314         with spab, kb.kbkopf,nb
315         do begin
316            init (kcpal, nb);
317            if kcrccc <> '000' then goto 9;
318            mget (kcpal, nb);
319            if kcrccc = '05Z'
320            then spaces (nb.data)
321            else
322            if kcrccc <> '000' then goto 9;
323            spaces (form.meldungstext);
324            unpack (concat
               ('******FALSCHER TAC - BITTE EINGABE WIEDERHOLEN.******'),
325                    meldung, 1);
326            spaces (form.tac);
327            mput (kcpal, nb, sizeof (id_nb));
328            if kcrccc <> '000' then goto 9;
329            pend (kcpal, 'FI');
330
331         9: (* KDCS-Fehler *)
332            fehler ('BADTACS', kcpal.kcop, kcrccc, spab);
333
334         end;
335      end (*BADTACS*);
336
337
338      begin (* package body is empty *)
339      end.

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

                    PROCEDURE ENTRY VECTOR

       PEV-ADDRESS      MODULE-OFFSET        PROCEDURE / FUNCTION
      24 (00000018)      96 (00000060)   INITIAL PROCEDURE
      28 (0000001C)       0 (00000000)   ANZEIGE
      32 (00000020)       0 (00000000)   AENDERN
      36 (00000024)       0 (00000000)   FIMAPAPA
      40 (00000028)       0 (00000000)   BADTACS
      44 (0000002C)       0 (00000000)   LESEN
      48 (00000030)       0 (00000000)   SCHREIBEN
      52 (00000034)       0 (00000000)   UEBERSCHREIBEN
      56 (00000038)       0 (00000000)   LOESCHEN
```

```
 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)   FEHLER


                GLOBAL CONSTANTS OF THE UNIT

MODULE-OFFSET     TYPE       NAME          VALUE
 96 (00000060)   STRUCT   space_constant
324 (00000144)   STRING   no_addr
                 '*** ADRESSE MIT DIESEM NAMEN NICHT VORHANDEN ***'
372 (00000174)   STRING   dup_addr
                 '*** ADRESSE MIT DIESEM NAMEN NICHT VORHANDEN ***'
420 (000001A4)   STRUCT   c_hexdig
 -1 (FFFFFFFF)   STRING                  ''
436 (000001B4)   STRING                  '*** DATEIFEHLER #'
453 (000001C5)   STRING                  ' ***'
 -1 (FFFFFFFF)   STRING                  ''
 -1 (FFFFFFFF)   STRING                  ''
 -1 (FFFFFFFF)   STRING                  ''
 -1 (FFFFFFFF)   STRING                  ''
457 (000001C9)   STRING                  'STARTUP '
465 (000001D1)   STRING                  '/FILE ADRESSEN,LINK=ADRDATEI,'
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             '*** F E H L E R ***
                                     TEILPROGRAMM: '
634 (0000027A)   STRING             ' OPERATIONSCODE: '
651 (0000028B)   STRING             ' RETURNCODE: '
664 (00000298)   STRING             'ER'
666 (0000029A)   STRING             'OOO'
669 (0000029D)   STRING             'ANZEIGE'
676 (000002A4)   STRING             'OOO'
679 (000002A7)   STRING             'ANZEIGE'
 -1 (FFFFFFFF)   STRING             ''
```

```
     686 (000002AE)   STRING              'OOO'
     689 (000002B1)   STRING              'ANZEIGE'
     696 (000002B8)   STRING              'FI'
     698 (000002BA)   STRING              'OOO'
     701 (000002BD)   STRING              'OOO'
     704 (000002C0)   STRING              '*** UNGÜLTIGER TAC ('
     724 (000002D4)   STRING              ') ***'
      -1 (FFFFFFFF)   STRING              ''
     729 (000002D9)   STRING              'OOO'
     732 (000002DC)   STRING              'FI'
     734 (000002DE)   STRING              'AENDERN'
     741 (000002E5)   STRING              'OOO'
     744 (000002E8)   STRING              '05Z'
     747 (000002EB)   STRING              'OOO'
     750 (000002EE)   STRING              '******FALSCHER TAC -
                                           BITTE EINGABE WIEDERHOLEN.******'
     804 (00000324)   STRING              '******FALSCHER TAC -
                                           BITTE EINGABE WIEDERHOLEN.******'
     859 (0000035B)   STRING              'OOO'
     862 (0000035E)   STRING              'FI'
     864 (00000360)   STRING              'BADTACS'

                    GLOBAL VARIABLES OF THE UNIT

      32 (00000020)   adrdatei
     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-Anweisungen

```
REM ************************************************************
REM ***             D E F  -  A N W E I S U N G E N          ***
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-ANWEISUNGEN         ************
REM ************************************************************
PROGRAM KDCADM,COMP=ILCS
PROGRAM ANZEIGE,COMP=ILCS
PROGRAM AENDERN,COMP=ILCS
PROGRAM FIMAPAPA,COMP=ILCS
PROGRAM BADTACS,COMP=ILCS
REM ************************************************************
REM ************           EXIT-ANWEISUNGEN           ************
REM ************************************************************
EXIT PROGRAM=FIMAPAPA,USAGE=START
EXIT PROGRAM=FIMAPAPA,USAGE=SHUT
REM ************************************************************
REM ************            TAC-ANWEISUNGEN           ************
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=ANZEIGE
TAC 2,LOCK=2,PROGRAM=AENDERN
TAC 3,LOCK=2,PROGRAM=AENDERN
TAC 4,LOCK=2,PROGRAM=AENDERN
REM ***************************************************************
REM ************           USER-ANWEISUNGEN           ************
REM ***************************************************************
USER  SUSI,PASS=C'UTM4EVER',KSET=BUND1,PERMIT=ADMIN,FORMAT=*FORMA
USER  TRUDI,PASS=C'UTMNEVER',KSET=BUND2,STATUS=ON,FORMAT=*FORMA
USER  BAERBEL,KSET=BUND3,STATUS=ON,FORMAT=*FORMA
REM ***************************************************************
REM ************        PTERM/LTERM-ANWEISUNGEN       ************
REM ***************************************************************
DEFAULT PTERM PRONAM=DSR01,PTYPE=T9750
PTERM DSS01,LTERM=UTMDST1
PTERM DSS02,LTERM=UTMDST2
PTERM DSS03,LTERM=UTMDST3
DEFAULT PTERM PRONAM=DSR01,PTYPE=T9022,USAGE=O
PTERM G01,LTERM=DRUCKER,CONNECT=A
LTERM UTMDST1,KSET=BUND1
LTERM UTMDST2,LOCK=4,KSET=BUND1
LTERM UTMDST3,LOCK=5,KSET=BUND1
LTERM DRUCKER,USAGE=O
REM ***************************************************************
REM ************           KSET-ANWEISUNGEN           ************
REM ***************************************************************
KSET  BUND1,KEYS=(1,2,3,4,5)
KSET  BUND2,KEYS=(1,2,4)
KSET  BUND3,KEYS=(1)
REM ***************************************************************
REM ************            TLS-ANWEISUNGEN           ************
REM ***************************************************************
TLS    TLSA
TLS    TLSB
END
```

4 Datenstrukturen für Pascal-XT

Zu den PASCAL-XT Datenstrukturen gibt es pro Paket jeweils 2 Dateien, die sich im letzten Buchstaben des Dateinamens unterscheiden:

S kennzeichnet die „Paket-Spezifikation“, sie enthält die Beschreibung der Datenstrukturen.

B kennzeichnet die „Paket-Implementierung“, welche die Implementierung des Pakets enthält. Die Implementierung kann leer sein, muß aber immer vorhanden sein.

Paket 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  *)
```

Paket 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}
```

Paket 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}
```

Paket 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}           KCDADMl      = 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}
```

Paket 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 }
```

Paket 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}          KCINIl      = 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}
```

Paket 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}           KCINFO      = 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}
```

Paket 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}
```

Paket 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 recorde 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}
```

Paket 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}
```

Paket 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}
```

Paket TIAMCTRL (Beispiel)

Dieses Paket wird nicht mit UTM ausgeliefert !

```
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.
```

Stichwörter

Inhalt

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

Anwendungen programmieren mit KDCS für Pascal-XT

Zielgruppe
Programmierer von UTM-Pascal-XT-Anwendungen.

Inhalt
– Umsetzung der Programmschnittstelle KDCS in die Sprache Pascal-XT
– Alle Informationen, die der Programmierer von UTM-Pascal-XT-Anwendungen benötigt

Ausgabe: Februar 1997

Datei: utm_pas.pdf

SINIX und BS2000 sind eingetragene Warenzeichen der
Siemens Nixdorf Informationssysteme AG

Copyright © Siemens Nixdorf Informationssysteme AG, 1997.

Alle Rechte vorbehalten.
Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller

Information on this document

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

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

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

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

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

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

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

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009