ASYS3 Kurs Landesbank Berlin

Praxis Workshop

Ullrich Barmeyer
Dipl.Ing. (FH), Dipl.Inf. (TU)
BCS Beratung Computer Software
Kleinmachnow 2006

Inhalt

1 Benutzung der Services

Die Serviceleistungen sind in folgende **Gruppen** aufgeteilt:

1. **Datenbank-Service**	Zugriff auf IMS-Datenbank (D**BSERV)**
2. Parameter-Service	stellt institutsspezifische Daten bereit, -analog der früheren IPAR49-Anforderung **(PRMSERV)**
3. Datengruppen-Service	Zugriff auf ASYS3-Dateneinheiten (z.B. lEs) **(PFSERV/IESERV/DGSERV)**
4. **Umsatz-Service**	Erfassung der Umsätze für Btx-Überweisungen und Zugriffsmöglichkeit der Anwendungsmodule auf die Umsatzerfassungs-DB **(UMSSERV)**
5. Bediener-Service	Arbeit mit dem dvg-Sicherungsverfahren (z.B. Bedienerprüfung) **(MBBSERV)**
6. **Statistik-Service**	schreibt Log-Sätze für die Statistik (FIBU), **(STATSERV)**

7. Protokoll-Service	gibt Es als Protokoll-Sätze aus, die von NAFA verarbeitet werden **(PROTSERV)**
8. Vorgangs-Service	übernimmt die Kommunikation zwischen Anwendung und Vorgangssystem **(VORSERV)**
9. System-Service	Fehlerbehandlungsservice, Speicherauszugs- und Ablaufaufzeichnungsservice **(ABSERV / TRCSERV)**
10. **Nachrichtenaufbereitung,**	unterstützt die Aufbereitung von Daten, u.a. **Variablenbeschaffung,** im Bereich der Textverarbeitung **Textverarbeitung (VARSERV / DARSERV)**
11. 3-Schemata Architektur	Konzept zur Umsetzung und Bereitstellung von Daten entsprechend der logischen Anwendersicht und der phys. Speicherung **(VIEWSERV)**

12. Kommunikationssystem	System zur rechnerübergreifenden Kommunikation (z.B. Postsystem) **(KOMMSERV**

2 Service-Aufruf-Makros

Die Service-Aufruf-Makros erfüllen folgende Funktionen:

- Sie stellen die Verbindung zwischen dem Anwendungsprogramm und dem Serviceprogramm her und stellen die Verarbeitungsparameter für die Service-Verarbeitungsprogramme zur Verfügung
- Die Makros werten Teile der von den Serviceprogrammen zurückgegebenen Informationen aus (Statuscodes).

Beipiele:

Definitions-Makros

IESERV, DBSERV

Sie definieren besondere Antwortbereiche für die Serviceprogramme im

Preprozessor-Funktionen

- Sie stellen die Verbindung zwischen der C- Anwendung und dem Serviceprogramm her und stellen die Verarbeitungsparameter für die Service-Verarbeitungsprogramme zur Verfügung

Beipiel:

EXEC DBSERV

Die Anwendungsprogramme und Tabellen im Produktions-Release können in 3 Teile geteilt werden: Bestandteile, die

in der **Front-End- Verarbeitung** benutzt werden in der **Back-End- Verarbeitung** benutzt werden **gemeinsam** benutzt werden

Im **Front-End** werden benutzt:

Aufbereitungsvorschriften (AV)
Logische Vordrucke (LV)
Formular-Bildschirmmasken (FO-Modul für IBM 3270 und dvg-Terminal)
Auswahlknoten (KA) für ADS und SDS
Funktionsknoten (KF) für ADS und SDS
Routing-Tabelle
Transaktionszuordnungs-Tabelle (TB- Modul)
Hilfe-Module (Hl)
Hilfe Dokument (HD) (nur Test)

Im **Back-End** werden benutzt:

Dialogfolgen (DF)
Aktionsfolgen (AF)
Aktionsmodule (AM oder AMC)
Druckmodule (DR oder DRC)
Dokumente (DO) aus der dvg-Textverarbeitung
Format-Dokument (FD) der dvg-Textverarbeitung
Datentrafos (DT) (Element der 3 Schemata-Architektur)
Viewbeschreibungen (VB) (siehe "3 Schemata-Architektur")

Gemeinsam werden benutzt:

E-Beschreibungen (IE)
Datengruppenkatalog
(TB000001) Tabellenmodule
(TB)

Vom Back-End benutzte Bestandteile

Dialogfolgen sind nicht selbständig ausführbare Module, die in einer Makrosprache kodiert werden. Sie legen die Folge der einzelnen Dialogschritte fest. In einem Dialogschritt wird eine Aktionsfolge aufgerufen.

In IE-Beschreibungen werden die Verarbeitungsdaten mit Beschreibung des Aufbaus dieser Datentypen (z.B. Einteilung in Untertypen, Formate,...) festgelegt. Diese Festlegung erfolgt meist durch den Anwendungsprogrammierer in einer Makro-Sprache.

Der **Datengruppenkatalog** beinhaltet die Beschreibung der möglichen ASYS3-Dateneinteilungen. Jede Informationseinheit (IE) befindet sich in einem Datengruppenausschnitt. Beschreibungen des DGA-Typs erfolgt im Datengruppenkatalog. Weiterhin werden Datengruppen mit Ihren Zugriffsberechtigungen festgelegt. Dieser zentrale Katalog wird von der 'OE Zentrale Anwendungsplattform' festgelegt und verwaltet.

3 Einbindung in C/370

3.1 Parameterservice

Aufrufbeispiel

```
memcpy (INSTFELD.instfeld," x09 x97",2);
memcpy (kst_nr," x00 x00 x00 x00 x22 x2F",6);
memcpy (kst_bez,"TEST PRMSERV EINFUEGEN                    ",40);
memcpy (kst_bezk,"EINFUEGEN ",10);
memcpy (kst_typ,"M",1);


EXEC PRMSERV_EINFUEGEN
     INST INSTFELD.instfeld
     TABELLE "KR.KST"
     BEDINGUNG "KSTNR",==,kst_nr
     PARAMETER "BEZL",kst_bez,
               "BEZK",kst_bezk,
               "KSTART",kst_typ
     ANART "BUENDEL";
```

Asys 3 Sfa

```
switch(as3_stcd)
   »
   case IO:printf ("ALLES KLAR ==> IO n");
           fflush(stdout);
           break;
   case FF:printf ("FEHLER ==> FF n");
           fflush(stdout);
           break;
   case IF:printf ("FEHLER ==> IF n");
           fflush(stdout);
           break;
   ¤
```

4 Beispielprogramm AMC0029 Übernahme Darlehen

4.1 Programm Header

```
# #pragma pagesize(69)
 #pragma title("AMC0029")
 #pragma strings(readonly)
// Testhilfen
// #define TEST
#ifdef TEST
   #define xctst_progstat_test          // Test ist AKTIV
   #define stichprobe_nein_danke
   #define xctst_as3                    // Umleiten nach as3.bt......
   #define TRACE_LEVEL 1                // 2=AVTEST
   #define KONTROLLE 0                  // keine Kontrollpflicht
#else
   #define xctst_progstat_prod          // Test ist nicht AKTIV
   #define TRACE_LEVEL 0                // 2=AVTEST
   #define KONTROLLE 10                 // Kontrollpflicht
#endif
/* DOCANF ************************************************************
  ====================================================================
  ³                                                                  ³
  ³  AMC0029 :   CSB Uebernahme Darlehen (KV)                        ³
  ³              ( LBS - Berlin )                                    ³
  ³                                                                  ³
  ³              Programmdokumentation                               ³
  ³                                                                  ³
  ³              1. Lebenslauf                                       ³
  ³              2. Programmbeschreibung                             ³
  ³                                                                  ³
  ³==================================================================³
  ³1³                                                                ³
  ³1³  Programm:      AMC0029                                        ³
  ³1³   Version:      001                                            ³
  ³1³                                                                ³
  ³1³ Durchführung    OE              : LBB, OR33                    ³
  ³1³                 Name            : Barmeyer                     ³
  ³1³                 Termin          :                              ³
  ³1³                                                                ³
  ³1³ Auftragsinhalt                                                 ³
  ³1³   Version 007: Barmeyer Prisma GmbH        BA981126            ³
  ³1³                Rest Buergschaftsgebuehr löschen LB5BUEGR       ³
  ³1³   Version 006: Barmeyer Prisma GmbH        BA980819            ³
  ³1³                Korrektur Mahnstatus Initialisierung 00         ³
  ³1³   Version 005: Barmeyer Prisma GmbH        BA980109            ³
  ³1³                Korrektur Aufruf von Rechnungsabgrenzung        ³
  ³1³   Version 004: 12.12.97 Barmeyer Prisma GmbH                   ³
  ³1³                KA 30 nur bei LB5LDAT=leer abweisen             ³
  ³1³                Pruefen Kontonummernkreis                       ³
  ³1³   Version 003: 17.10.97 Barmeyer Prisma GmbH                   ³
  ³1³                Korrektur do_dummy_history                      ³
  ³1³   Version 002: Neuerstellung enthält die Aktion                ³
```

```
  ³1³                  XLCUND                                          ³
  ³1³   Version 001: Dummy-Version                                     ³
  ³1³                                                                  ³
  ³1³ Testdokumente:  Protokoll eines Testlaufes                       ³
  ³1³                                                                  ³
  ³1³ Abhängigkeiten: keine                                            ³
  ³1³                                                                  ³
* DOCEND *****.plau*************************************************/
   /*-* C/370 - Funktionen   --*/
   #include <stdio.h>           /*  Standard IO-Funktionen       */
   #include <stdlib.h>          /*  Standard C- Funktionen       */
   #include <string.h>          /*  String-Handling              */

   /*-* dvg - Funktionen     --*/
   #include <cvgisys.h>         /*  Systemfunktionen dvg         */
   #include <cvgitrc.h>         /*  Tracefunktionen  dvg         */
   #include <amcdef.h>          /*  ASYS3-Aktionsmodul-Defin.    */
   #include <cvgifbib.h>
   #include <cvgipvb.h>         /*  Programmverbindungs-Funkt.   */
   #include <ims.h>             /*  IMS-Funktionen               */
   #include <cvgipak.h>         /*  gepackte Arithmetik dvg      */
   #include <stdas3.h>          /*  Standard ASYS3-Definitionen  */
   #include <srviedb.h>         /*  IE-DB Services               */
   #include <time.h>            /*  Time-Funktionen              */

   /*-* LBS Funktionen       --*/
   #include <xcpfad.as3>        /*  Pfade öffnen                 */
   #define   xc_progtyp_main    /*  DIES IST DAS HAUPTPROGRAMM   */

   #define RESTSEGMFUNKT 2      /*  LBS RESTE-DB                 */
   #include <restsegm.h>        /*  LBS RESTE-DB                 */
   #include <xcbbest>           /*  Segmentbeschreibungen Best.DB*/
   #include <xcpbest>           /*  Segmentbeschreibungen Prov.DB*/
   #include <xctbest>           /*  Segmentbeschreibungen Umsatz */
   #include <xclib>             /*  LBS-Standardfunktionen       */
   #include <xctstlib>          /*  LBS-Test-Funktionen          */
   #include <xci02>             /*  LBS-Fehlerkatalog            */
   #include <xcspe>             /*  Sperren-Daten                */
   #include <xcu01.h>           /*  LBS-Standard-Unterprogramme  */
   xc_object_label(AMC0029,AMCEXE);
```

4.2 Deklaration zum Schreiben der IE

```
/*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx¯
³ Typ-Definition für Geschäftsvorfall-IE: XUEBDARL                   ³
¿xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*/
typedef _Packed struct
  {
  EXEC DECLARATION
  FELD iktonr       FORMAT numz   LAENGE   10  SPKLASSE struct
  FELD iekey        FORMAT undef  LAENGE 50    SPKLASSE struct
  FELD ikzkrdr      FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib5korue     FORMAT pack   LAENGE 6     SPKLASSE struct
  FELD ib5gerue     FORMAT pack   LAENGE 6     SPKLASSE struct
  FELD ib5zsrue     FORMAT pack   LAENGE 6     SPKLASSE struct
```

```
  FELD ib5aupfl     FORMAT pack   LAENGE 6     SPKLASSE struct
  FELD iktonr2      FORMAT numz   LAENGE 10    SPKLASSE struct
  FELD ib5tisal     FORMAT pack   LAENGE 6     SPKLASSE struct
  FELD ib5saeum     FORMAT pack   LAENGE 6     SPKLASSE struct
  FELD ib5stund     FORMAT numz   LAENGE 8     SPKLASSE struct
  /* Attribute */
  FELD at_iktonr2   FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5korue  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5gerue  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5zsrue  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5aupfl  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5ktonr2 FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5tisal  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5saeum  FORMAT bin    LAENGE    2  SPKLASSE struct
  FELD at_ib5stund  FORMAT bin    LAENGE    2  SPKLASSE struct
  /* XUEBKRED-Anf
  FELD $feld-1$     FORMAT $$$f   LAENGE $$$l  SPKLASSE struct
  XUEBKRED-End */
  FELD ibhinwkz     FORMAT char   LAENGE 1     SPKLASSE struct
  FELD iktonr1      FORMAT numz   LAENGE 10    SPKLASSE struct
  FELD ib1kunr      FORMAT numz   LAENGE 10    SPKLASSE struct
  FELD ib1gebl      FORMAT numz   LAENGE 10    SPKLASSE struct
  FELD ib2uebg      FORMAT pack   LAENGE 4     SPKLASSE struct
  FELD ib1kfgeb     FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib1hause     FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib1dsch      FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib1dup       FORMAT numz   LAENGE 10    SPKLASSE struct
  FELD ib1bzsne     FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib5zahl      FORMAT pack   LAENGE 16    SPKLASSE struct
  FELD ibdrue       FORMAT char   LAENGE 1     SPKLASSE struct
  FELD ib5abrd      FORMAT numz   LAENGE 8     SPKLASSE struct
  FELD ib1kusy      FORMAT char   LAENGE 8     SPKLASSE struct
  /* XUEBDARL-Anf
  FELD $feld-1$     FORMAT $$$f   LAENGE $$$l  SPKLASSE struct
  XUEBDARL-End */
```

.....

4.3 Datengruppen öffnen, alte IE löschen

```
// Datengruppenservice
  EXEC DGSERV
  OEFFNE "LV_KUN" AUF ALTEN DGA "LEVELSPEICHER";
  switch (as3_stcd)
    {
  case IO: break;
    }

  // alte XLIKUNDD loeschen, falls vorhanden
  while (as3_stcd == IO)
    {
    EXEC DGSERV
    POSITIONIERE ERSTE IE "XLIKUNDD" AUF "LV_KUN";
    switch (as3_stcd)
      {
```

```
    case NV : break;
    case IO :
      EXEC DGSERV
      LOESCHE IE AUF "LV_KUN";
      switch (as3_stcd)
        {
      case NV:  break;
      case IO:  break;
        }
      }
    }
```

4.4 IE auf LV einfügen

```
EXEC DGSERV
  FUEGE ERSTE IE "XLIKUNDD" EIN AUF "LV_KUN";
  switch (as3_stcd)
    {
  case IO:  break;
    }

  // notwendige Input Parameter
  EXEC DGSERV
  SCHREIBE FELD "KUNR", "FUNKT","KTONR","KFGEB"
  IMMER IN "XLIKUNDD" AUF "LV_KUN"
  AUS p_kun->kunr, p_kun->funkt, p_kun->ktonr, p_kun->kfgeb;
  switch (as3_stcd)
    {
  case IO:  break;
    }

  EXEC DGSERV
  GIB IE FREI AUF "LV_KUN";
  switch (as3_stcd)
    {
  case IO:  break;
    }
```

4.5 Unterprogrammaufruf

```
// LINKSERVICE
EXEC LINKSERV
  AK "XLUKUNDD" AKTAK "XLCUND";
  switch (as3_stcd)
    {
  case 0:
    /* Linkserve rc=OK */
    EXEC DGSERV
    POSITIONIERE ERSTE IE "XLIKUNDD" AUF "LV_KUN";
    switch (as3_stcd)
      {
    case NV :
      lvput_error (101, "IE XLIKUNDD nicht gefunden","");
      rc_up = FEHLER;
```

```
    break;

  case IO :
    // XLIKUNDD Ergebnis lesen
    // 1. Fehlermeldung vorhanden ?
    // wenn ja, liegt ein Fehler, z.B. Postverbot vor
    EXEC DGSERV
    LIES FELD "TXTCL78"
    IN "XLIKUNDD" AUF "LV_KUN"
    NACH       p_kun->txtcl78;
    switch (as3_stcd)
      {
    case IO:
      xcstr_init (workc78);
      xcstr_copy (workc78, p_kun->txtcl78) ;
      workc78[77] = '\x0';
      lvput_error (102, workc78,"");
      rc_up = FEHLER;
      break;
    case NV:
      rc_up = OK;
      break;
      }

    if (strstr ("KUSYMA" , p_mode) NE NULL)
      {
      /* XLIKUNDD Ergebnis lesen */
      EXEC DGSERV
      LIES FELD "KUST", "GEBDAT", "KUSYMA", "SPERRE",
                "HINWEIS", "KUNDART"
      IN "XLIKUNDD" AUF "LV_KUN"
      NACH       p_kun->kust,
                 p_kun->gebdat,
                 p_kun->kusyma,
                 p_kun->sperre,
                 p_kun->hinweis,
                 p_kun->kundart;
      switch (as3_stcd)
        {
      case IO:
        break;
      case NV:
        break;
        }
      }
    else
      {
      /* XLIKUNDD Ergebnis lesen */
      EXEC DGSERV
      LIES FELD "ANSCHRZ1", "ANSCHRZ2", "ANSCHRZ3", "ANSCHRZ4",
                "ANSCHRZ5", "ANSCHRZ6", "ANSCHRZ7", "ANSCHRZ8",
                "NAME", "TITEL" , "STRASSE", "PLZORT"
      IN "XLIKUNDD" AUF "LV_KUN"
      NACH       p_kun->anschrz1,
                 p_kun->anschrz2,
                 p_kun->anschrz3,
                 p_kun->anschrz4,
```

```
                    p_kun->anschrz5,
                    p_kun->anschrz6,
                    p_kun->anschrz7,
                    p_kun->anschrz8,
                    p_kun->name   ,
                    p_kun->titel  ,
                    p_kun->strasse ,
                    p_kun->plzort  ;
          switch (as3_stcd)
            {
          case IO:
            break;
          case NV:
            break;
            }
          }
```

.....
.....

4.6 Eingabe (TPI) lesen

```
/* DOCANF ************************************************************
 * Name:       tpiget_anwie                                          *
 * Titel:      Geschäftsvorfall-IE mit Daten aus TPI überschreiben   *
 * Parameter:  IE-Struktur mit Initialwerten                         *
 * Return:     IE-Struktur mit Eingabedaten                          *
 * Ablauf:     Die Funktion erhält als Parameter die IE zum GV,      *
 *             wie sie aus SPEICHER gelesen wurde und aktualisiert   *
 *             jedes Feld, das in TPI vorhanden ist.                 *
 *                                                                   *
 * Hinweise:   Felder, die in TPI nicht vorhanden sind, bleiben      *
 *             unverändert.                                          *
 * Beispiel:                                                         *
 * DOCEND *****.tpi***************************************************/
static ANWIE tpiget_anwie /* Output: IE-Struktur mit Eingabedaten    */
  (ANWIE ie)              /* Input:  Initialisierte IE-Struktur      */
{
  xctst_begin ("tpiget_anwie");
  /*-----------------------------------*/
  /* IE positionieren                  */
  /*-----------------------------------*/
  EXEC DGSERV
  POSITIONIERE ERSTE IE "XUEBDARL" AUF "TPI";
  switch (as3_stcd)
    {
  case IO:
    /*-------------------------------------------*/
    /* Eingabefelder mit Attribut aus IE lesen   */
    /*-------------------------------------------*/

    EXEC DGSERV
    LIES FELD "IKTONR1" IN "XUEBDARL" AUF "TPI"
    NACH ie.iktonr1 ;
    switch (as3_stcd)
      {
```

```
    case IO: break;
    case NV: break;
      }
    EXEC DGSERV
    LIES FELD "IKTONR" IN "XUEBDARL" AUF "TPI"
    NACH ie.iktonr MIT ATTRIBUTEN ie.at_iktonr;
    switch (as3_stcd)
      {
    case IO: break;
    case NV: break;
      }
    EXEC DGSERV
    LIES FELD "IB1KUNR" IN "XUEBDARL" AUF "TPI"
    NACH ie.ib1kunr MIT ATTRIBUTEN ie.at_ib1kunr;
    switch (as3_stcd)
      {
    case IO: break;
    case NV: break;
      }
    EXEC DGSERV
    LIES FELD "IB1GEBL" IN "XUEBDARL" AUF "TPI"
    NACH ie.ib1gebl MIT ATTRIBUTEN ie.at_ib1gebl;
    switch (as3_stcd)
      {
    case IO: break;
    case NV: break;
      }


  xctst_ende ("tpiget_anwie");
  return ie;
}
```

……..
……

4.7 Ausgabe (TPO) auf Bildschim schreiben

```
  /*-----------------------------------------*/
  /* IE positionieren, falls schon vorhanden  */
  /* ansonsten wird die IE hier eingefügt     */
  /*-----------------------------------------*/
  EXEC DGSERV
  POSITIONIERE ERSTE IE "XUEBKRED" AUF "TPO";
  switch (as3_stcd)
    {
  case IO: break;
  case NV:
    EXEC DGSERV
    FUEGE ERSTE IE "XUEBKRED" EIN AUF "TPO";
    switch (as3_stcd)
      {
    case IO: break;
      }
    break;
```

```
    }
  /*-------------------------------------------------------*/
  /* IE-Felder schreiben,  Felder ohne Attribut,           */
  /* die immer angezeigt werden sollen.                    */
  /*-------------------------------------------------------*/
  /*-------------------------------------------------------*/
  /* IE-Felder schreiben, Felder ohne Attribut,            */
  /* die nur angezeigt werden, wenn sie nicht leer sind.   */
  /*-------------------------------------------------------*/
  if (NOT xcstr_leer (ie.ikzkrdr))
    {
    EXEC DGSERV
    SCHREIBE FELD "IKZKRDR" IMMER IN "XUEBKRED" AUF "TPO"
    AUS ie.ikzkrdr ;
    switch (as3_stcd)
      {
    case IO: break;
      }
    }
  if (NOT xcnum_leer (ie.iktonr1))
    {
    EXEC DGSERV
    SCHREIBE FELD "IKTONR1" IMMER IN "XUEBKRED" AUF "TPO"
    AUS ie.iktonr1 ;
    switch (as3_stcd)
      {
    case IO: break;
      }
    }
  if (NOT xcpak_leer (ie.ib5korue))
    {
    EXEC DGSERV
    SCHREIBE FELD "IB5KORUE" IMMER IN "XUEBKRED" AUF "TPO"
    AUS ie.ib5korue MIT ATTRIBUTEN ie.at_ib5korue;
    switch (as3_stcd)
      {
    case IO: break;
      }
    }


  xctst_ende ("tpoput_bild1");
  return ;
}
```

```
/* DOCANF ************************************************************/
/* Name.....:  do_umsatz_das_haus                                    */
/* Titel....:  Aufruf des XLUHAUS                                    */
/* Return...:  0 - OK                = rc(XLUHAUS)                   */
/*             1 - Parameret-Fehler  = rc(XLUHAUS)                   */
/* Parameter:  1. Kontonummer                          IN            */
/*             2. Datum für das anteilige Gebühr       IN            */
/*             3. Buchungs-KA                          IN            */
/* Globals..:  g. ,                                                  */
```

```
/* Hinweise.:  Fehlertext in XLIMSG                                        */
/*                                                                         */
/* Beispel..:  -                                                           */
/*   rc = do_umsatz_das_haus ( "8100014010", g.lbsbudat, "10" );           */
/*                                                                         */
/* DOCEND ******************************************************************/
short do_umsatz_das_haus (char * parm_ktonr ,
                          Datum  parm_datum ,
                          char * parm_ka )
{

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/*.  IE-Definition                                                        */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

/* EXEC DECLARATION generiert am:  11 Jan 1995 um:  16:01:56 */
/* aus:  TSSPK.GRULB.IE(XLIHAUS) */
  typedef _Packed struct
  {
    EXEC DECLARATION
    FELD ktonr        FORMAT numz  LAENGE 10     SPKLASSE struct
    FELD flnoxt       FORMAT char  LAENGE 1      SPKLASSE struct
    FELD aufruf       FORMAT char  LAENGE 1      SPKLASSE struct
    FELD datum        FORMAT numz  LAENGE 8      SPKLASSE struct
    FELD budat        FORMAT numz  LAENGE 8      SPKLASSE struct
    FELD wert         FORMAT numz  LAENGE 8      SPKLASSE struct
    FELD ka           FORMAT numz  LAENGE 2      SPKLASSE struct
    FELD txt1cl39     FORMAT char  LAENGE 39     SPKLASSE struct
    FELD txt2cl39     FORMAT char  LAENGE 39     SPKLASSE struct
    ;
  } XLIHAUS;
```

4.8 Protokollfile (PROTO) schreiben

```
/* DOCANF ******************************************************************
 * Name:       dbupd_umsatz_und_proto                                      *
 * Titel:      Insert Umsatzsegment (XT1) mit Protokollservice             *
 * Ablauf:     Insert XT1 und einfügen XNACHUMS                            *
 * Hinweise:   -                                                           *
 * Beispiel:                                                               *
 * DOCEND ******************************************************************/
static short dbupd_umsatz_und_proto (XT1SEG  LT1)
{
  /* XNACHUMS für Umsatzprotokoll */
  EXEC DECLARATION
  FELD ktosich  FORMAT char LAENGE 10 SPKLASSE auto
  FELD inbudat  FORMAT numz LAENGE 8  SPKLASSE auto
  FELD inwert   FORMAT numz LAENGE 8  SPKLASSE auto
  FELD inverst  FORMAT char LAENGE 3  SPKLASSE auto
  FELD inbkz    FORMAT char LAENGE 2  SPKLASSE auto
  FELD inkaums  FORMAT char LAENGE 2  SPKLASSE auto
  FELD inbetrag FORMAT pack LAENGE 6  SPKLASSE auto
  ;
  xctst_begin ("dbupd_umsatz_und_proto");
```

```
  // XT1 hinzufuegen
  xcdb_serv1  ( , ISRT, LT1, g.inst, ssa_u (XT1SEG) );


  // Feldaufbereitungen
  xckon_dat2c (&inbudat, DATUM LT1BUDAT, tmj);
  xckon_dat2c (&inwert , DATUM LT1WERT , tmj);
  xcstr_copy  (inverst, LT1VERST);
  xcstr_copy  (inbkz  , LT1BKZ  );
  xcstr_copy  (inkaums, LT1KA   );
  xcpak_copy  (inbetrag, LT1BETR  );

  // IE einfuegen
  EXEC DGSERV
  FUEGE ERSTE IE "XNACHUMS" EIN AUF "PROTO";
  switch (as3_stcd)
    {
  case IO: break;
    }

  // IE-Felder schreiben
  EXEC DGSERV
  SCHREIBE FELD "INBUDAT","INWERT","INVERST","INBKZ","INKAUMS","INBETRAG"
  IMMER IN "XNACHUMS" AUF "PROTO"
  AUS inbudat,inwert,inverst,inbkz,inkaums,inbetrag;
  switch (as3_stcd)
    {
  case IO: break;
    }

  // Protokollservice
  xcstr_copy (ktosich, LT1KTONR);
  EXEC PROTSERV
  HGV "AEND"
  GA  "LBS"
  GV  "UMSATZ"
  KTO  ktosich;
  switch (as3_stcd)
    {
  case IO :
    break;
    }

  xctst_ende  ("dbupd_umsatz_und_proto");
}
/* DOCANF ************************************************************
 * Name:       do_init_buchen                                        *
 * Titel:      Standardwerte fuer Buchungssätze XT                   *
 * Parameter   p_ib1   IE-Inhalt für Eingaben zu LB1                 *
 * Ablauf:     Feldaufbereitung und Insert XT1                       *
 * Hinweise:   -                                                     *
 * Beispiel:                                                         *
 * DOCEND ************************************************************/

/* DOCANF ************************************************************
 * Name:       Kontrolle                                             *
 * Titel:      Prüfen, ob ein kontrollpflichtiger Sachverhalt vorliegt*
```

```
 * Parameter:  IE-Struktur aus Eingabe                                *
 * Return:      0 = keine Kontrolle notwendig                         *
 *             10 = es liegt ein kontrollpflichtiger Fall vor         *
 * Ablauf:     ...                                                    *
 * Hinweise:   -                                                      *
 * Beispiel:                                                          *
 * DOCEND *************************************************************/
static short kontrolle
  (ANWIE tpi, XB1SEG *P_LB1)
{
  short rc = 0, rc_betrang = 0;
  PL6   b1summe_max;   /* Obergrenze für Vertragssumme */
  PL6   b1summe_min;   /* Obergrenze für Vertragssumme */
  PL6   workp6;
  PL6   pl0;
  XB5SEG LB5, *P_LB5 = &LB5;

  if (NOT xcnum_leer (P_LB1UEBA))
    {
    tpo_printf ("Vertrag wurde bereits übernommen von  %10.10s",
                P_LB1UEBA);
    return KONTROLLE;
    }

  // Sperren vorhanden
  if (xcspe_ist_sperre_aktiv (P_LB1KTONR))
    {
    tpo_printf ("Sperre(n) zum alten Vertrag vorhanden");
    return KONTROLLE;
    }

  /*-------------------------------------------------------*/
  /* Tabellenzugriff zum Ermitteln der Kontrollgrenzen      */
  /*-------------------------------------------------------*/
  xcpak_init (pl0);
  xctab (
    /* Output: */     &b1summe_max,
    /* Input:    */   6, T60PRUE, T60MAX, g.t60key, g.lbsbudat);

  if (xcpak_cmp (P_LB1SUMME, >, b1summe_max))
    {
    tpo_printf ("Bausparsumme übersteigt Kontrollgrenze");
    return KONTROLLE;
    }

  xctab (
    /* Output: */     &b1summe_min,
    /* Input:    */   6, T60PRUE, T60MIN, g.t60key, g.lbsbudat);

  if (xcpak_cmp (P_LB1SUMME, <, b1summe_min))
    {
    tpo_printf ("Bausparsumme unterschreitet Kontrollgrenze");
    return KONTROLLE;
    }

  // Keine Kontofuehrungsgebuehr bei NICHT-Betriebsangehoerigen
  if (tpi.ib1kfgeb[0] == 'N')
```

```
    {
    if (NOT ist_kunde_betriebsangehoerig (tpi.ib1kunr," "))
      {
      tpo_printf ("Kontoführungsgebuehr ist 'N'");
      return KONTROLLE;
      }
    }
  return OK;
}
```

4.9 Link für Unterprogramme herstellen

```
/* DOCANF ************************************************************
 *                                                                   *
 * Name:       object_reference                                      *
 * Titel:      alle Namen der Objektmodule für AMC0029               *
 * Return:     -                                                     *
 *             -                                                     *
 * Ablauf:     Das Modul definiert die Namen der Object-Module,      *
 *             die beim Linken eingebunden werden müssen, um         *
 *             die in 'xoammain.h' definierten Funktionen aufzulösen.*
 * Hinweise:   Diese Funktion wird nie ausgeführt!                   *
 * Beispiel:                                                         *
 *                                                                   *
 * DOCEND ************************************************************/
static void object_reference (void)
{
/* xo$NAM$01  ();   ... ggf. anwendungsspezifische Objekte ergänzen */
  return;
}
```

4.10 Logische Aktion (LOGAK)

Hier findet der Einsprung aus der Aktionsfolge statt

```
/* DOCANF ************************************************************
 * Name:       AMC0029                                               *
 * Titel:      Aufruf der Aktionen dieses Aktionsmoduls              *
 * Return:     kein Returncode                                       *
 * Ablauf:     siehe Beschreibung der Aktionen                       *
 * Hinweise:   zu jedem LOGAK xxxx INST yyy muß eine Funktion        *
 *             xxxx_yyy vorhanden sein, die bei Aufruf der           *
 *             Aktion xxxx von Institut yyy ausgeführt wird.         *
 * Beispiel:                                                         *
 * DOCEND ************************************************************/
  EXEC AMSTART
  LOGAK XLCUND INST std;
```

ASSEMBLER zum Vergleich

```
         MVC   INSTFELD,=X'0997'
         MVC   KST_NR,=P'222'
         MVC   KST_BEZ,=C'TEST PRMSERV EINFUEGEN                '
         MVC   KST_BEZK,=C'EINFUEGEN '
         MVI   KST_TYP,C'M'
*
         PRMSERV                                                 *
               FUNK=EINFUEGEN,                                   *
               INST=INSTFELD,                                    *
               TABELLE=KR.KST,                                   *
               BEDINGUNG=((KSTNR,EQ,KST_NR)),                    *
               PARAMETER=((BEZL,KST_BEZ),                        *
               (BEZK,KST_BEZK),                                  *
               (KSTART,KST_TYP)),                                *
               ANART=BUENDEL
*
         STCD  IO
         ...
         STCD  FF
         ...
         STCD  IF
         ...
         STCDENDE
```

5 Fehlerbehandlung

TRACE Service Aufruf

JCL - Beispiele:

```
//*********************************************************************
//*
//*  AUSGABE        VON DATUM UND UHRZEIT IN TPO
//*  (ASYS3-BATCHVERARBEITUNG OHNE IMS(DFSRRCOO) )
//*********************************************************************
//STEPl   EXEC PGM=XMSDCTL0,REGION=6500K
//XMSSTART DD     *
START SS(XEOl)    ,IMS(IME),ATTACH(DIAM900)
//STEPLIB  DD     DSN=OWUSER.A101292.ASY11.EXE        DISP=SHR
//       DD       DSN=DVGIME.BIBL,DISP=SHR
//       DD       DSN=IME.RESLIB,DISP=SHR
//DFSRESLB DD     DSN=IME.RESLIB,DISP=SHR
//PRINTDD  DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
//*
//D900GEN    DD     *
*****   SVPRM INST=998
*****   TRACE AM800=J
*****   TRACE AM804=J
*****   TRACE BM004=J
*****   ESPIE NEIN
//*
//AS3PMOD    DD     *
*****  MOD          AM99020   N
//D900SPEZ DD       *
*****    RAK          LOGAK=DATUHR
***ENTER
//*
//AS3PDOKU DD     SYSOUT=*
//D9 00PROT DD    SYSOUT=*,DCB=(LRECL=133 , RECFM=     =FBM,BLKSIZE  =133)
//D9 00KART DD    SYSOUT=*,DCB=(LRECL=133 , RECFM=     =FBM,BLKSIZE  = 133)
//S991SNAP DD     SYSOUT=*, DCB=(LRECL=133,RECFM=     FBM,BLKSIZE=  133)
```