CSYSTEMS™

PROGRAMSKI PAKET ZA KNJIGOVODSTVO

COBA Systems

Alaska Xbase++
DBF to PostgreSQL database application

(PART 5b)
ARTICLES-CODEBOOK3.EXE

01.11.2023

UVOD
INTODUCTION

Ovo je Deo 5b koji je nastavak Dela 5a ove knjige.
This is Part 5b which is a continuation of Part 5a of this book.

Deo 5 ove knjige podeljen je na Deo 5a, na Deo 5b i na Deo 5c, jer se u međuvremenu pokazalo da će na ovaj način biti uprošćeno izlaganje i bolje objašnjena softverska tehnologija koja je primenjivana na manjoj i prostijoj aplikaciji.
Zbog toga je za ovaj prikaz izabrana aplikacija ARTICLES-CODEBOOK.EXE koja formira i koristi registar i šifarnik artikala (ovde su artikli: roba, proizvodi, materijal, usluge, komisiona roba). Ova aplikacija je deo svakog poslovnog programa za robno knjigovodstvo i može se smatrati univerzalnim softverom.

Part 5 of this book is divided into Part 5a, Part 5b and Part 5c, because in the meantime it turned out that this way will simplify the presentation and better explain the software technology that was applied to a smaller and simpler application.
That is why the ARTICLES-CODEBOOK.EXE application was chosen for this display, which forms and uses the register and codebook of articles (here are articles: goods, products, materials, services, commission goods). This application is a part of every business program for goods accounting and can be considered as a universal software.

Deo 5a

A) Prikazuje istu aplikaciju ARTICLES-CODEBOOK.EXE pod nazivom **ARTICLES-CODEBOOK1.EXE** koja je urađena u tehnologiji Alaska Xbase++ DBF-DBT-NTX databaze (bez local ili remote Advantage Database Servera) uz upotrebu Xbase++ ISAM komandi i funkcija.

B) Prikazuje istu aplikaciju ARTICLES-CODEBOOK.EXE pod nazivom **ARTICLES-CODEBOOK2.EXE** koja je urađena u tehnologiji Alaska Xbase++ UPSIZE PostgreSQL databaze, prevođenjem sa DBF-DBT-NTX fajlova na PostgreSQL UPSIZE TABLE putem UPSIZE MAŠINE, a uz upotrebu ISAM komandi i funkcija i PGDBE SQL komandi i funkcija u komunikaciji sa PostgreSQL UPSIZE TABLAMA.

Part 5a

A) It shows the same ARTICLES-CODEBOOK.EXE application named **ARTICLES-CODEBOOK1.EXE** which is done in Alaska Xbase++ DBF-DBT-NTX database technology (without local or remote Advantage Database Server) using Xbase++ ISAM commands and functions.

B) It shows the same application ARTICLES-CODEBOOK.EXE called **ARTICLES-CODEBOOK2.EXE** which was made in Alaska Xbase++ UPSIZE PostgreSQL database technology, by translating from DBF-DBT-NTX files to PostgreSQL UPSIZE TABLE via UPSIZE MACHINE, and using ISAM command and function and PGDBE SQL command and function in communication with PostgreSQL UPSIZE TABLES.

Deo 5b

C) Prikazuje istu aplikaciju ARTICLES-CODEBOOK.EXE pod nazivom **ARTICLES-CODEBOOK3.EXE** koja je urađena u tehnologiji Alaska Xbase++ PostgreSQL databaze, bez upotrebe UPSIZE MAŠINE, već samo uz korišćenje PGDBE SQL komandi i funkcija i jednog broja PGDBE simuliranih ISAM funkcija.

Part 5b

C) It shows the same ARTICLES-CODEBOOK.EXE application called ARTICLES-CODEBOOK3.EXE which is done in Alaska Xbase++ PostgreSQL database technology, without using UPSIZE MACHINE, but only using PGDBE SQL commands and functions and a number of PGDBE simulated ISAM functions.

OVA APLIKACIJA IGNORIŠE UPSIZE MAŠINU UGRAĐENU U PGDBE MAŠINU I NE KORISTI NI JEDAN OD UPSIZE MEHANIZAMA, PA VELIKI NEOBAVEZNI DEO UPSIZE MAŠINE NE MORA NI POSTOJATI U DATABAZI APLIKACIJE:
- UPSIZE ISAM KOLONE U TABLAMA
- UPSIZE ISAM KOLONE ZA ISAM INDEKSE U TABLAMA
- UPSIZE ISAM INDEKSI U INDEKSIMA
- TRIGERI I TRIGER FUNKCIJE ZA TE ISAM INDEKSE
- OBAVEZNE PGDBE ISAM TABLE:
"alaska-software.isam.orders"
"alaska-software.isam.tables"
"alaska-software.system.connections"
moraju postojati jer ih PGDBE automatski formira,
ali mogu biti prazne (bez redova).

OVA APLIKACIJA MOŽE KORISTITI IZVORNE-STANDARDNE POSTGRESQL TABLE BEZ IKAKVIH UPSIZE DODATAKA.
OVA APLIKACIJA ZA KOMUNIKACIJU SA BAZOM PODATAKA KORISTI PGDBE SQL KOMANDE I FUNKCIJE, A TAKOĐE I JEDAN DEO ISAM KOMANDI I FUNKCIJA KOJE SU SIMULIRANE KROZ PGDBE.

THIS APPLICATION IGNORES THE UPSIZE MACHINE BUILT INTO THE PGDBE MACHINE AND IT DOESN'T USE ANY OF THE UPSIZE MECHANISMS, SO A LARGE OPTIONAL PART MACHINE UPSIZE DOESN'T EVEN EXIST IN THE APPLICATION DATABASE:
- **UPSIZE ISAM COLUMNS IN TABLES**
- **UPSIZE ISAM COLUMNS FOR ISAM INDEXES IN TABLES**
- **UPSIZE ISAM INDEXES IN INDEXES**
- **TRIGGER AND TRIGGER FUNCTIONS FOR THOSE ISAM INDEXES**
- **MANDATORY PGDBE ISAM TABLES:**
"alaska-software.isam.orders"
"alaska-software.isam.tables"
"alaska-software.system.connections"
they must exist because PGDBE creates them automatically,
but can be empty (no rows).

THIS APPLICATION CAN USE THE NATIVE-STANDARD POSTGRESQL TABLES WITHOUT ANY UPSIZE PLUGINS.
THIS APPLICATION USES PGDBE SQL COMMANDS AND FUNCTIONS TO COMMUNICATE WITH THE DATABASE, AND ALSO ONE PART OF ISAM COMMANDS AND FUNCTIONS THAT ARE SIMULATED THROUGH PGDBE.

Deo 5c

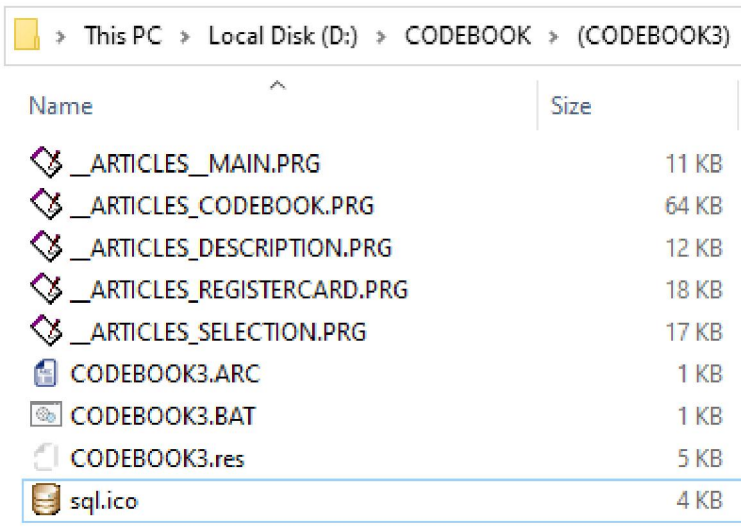
D) Prikazuje aplikaciju KASA-STOLOVI.EXE koja u sebi sadrži i aplikaciju ARTICLES-CODEBOOK.EXE urađenu po tehnologiji iz tačke B).
E) Na kraju ovog dela, uz ovu knjigu prilažem za download izvorni kod svih projekata koji su ovde izloženi i obrađeni (deo1, deo2, deo3, deo4, deo5a, Deo5b, Deo5c)

Part 5c

D) It shows the KASA-STOLOVI.EXE application, which also contains the ARTICLES-CODEBOOK.EXE application made according to the technology from point B).
E) At the end of this part, along with this book, I attach for download the source code of all the projects presented and processed here (part 1, part 2, part 3, part 4, part 5a, part 5b, part 5c)

Picture 3. ARTICLE-CODEBOOK3.EXE

Picture 7. PROJECT ARTICLES-CODEBOOK3.EXE




```
// ARTICLES-CODEBOOK3.XPJ --- START

[PROJECT]
    COMPILE       = xpp
    COMPILE_FLAGS = /q /dUSE_POSTGRES
    DEBUG         = no
    GUI           = yes
    LINKER        = alink
    LINK_FLAGS    =
    OBJ_DIR       = __________ARTICLES-CODEBOOK3.EXE
    RC_COMPILE    = arc
    RC_FLAGS      = /v
    ARTICLES-CODEBOOK3.XPJ

[ARTICLES-CODEBOOK3.XPJ]
ARTICLES-CODEBOOK3.EXE
[ARTICLES-CODEBOOK3.EXE]
// $START-AUTODEPEND
//---------------------------------
//  (CODEBOOK3)
//---------------------------------
  __ARTICLES__MAIN.OBJ  // MAIN()
  __ARTICLES_CODEBOOK.OBJ
  __ARTICLES_SELECTION.OBJ
  __ARTICLES_DESCRIPTION.OBJ
  __ARTICLES_REGISTERCARD.OBJ

// $STOP-AUTODEPEND
  (CODEBOOK3)\CODEBOOK3.RES
  XBTBASE1.LIB              // XbToolsIII
  XBTBASE2.LIB              // XbToolsIII
  dclipx.lib                // eXpress++

  BAZNE.LIB
  BAZNEx.LIB
  C-PGDB.LIB
```

```
// C-PGDB.INI
// KASA.INI
//-------------------------------------------------------
//  (CODEBOOK3)
//-------------------------------------------------------
  (CODEBOOK3)\__ARTICLES__MAIN.PRG
  (CODEBOOK3)\__ARTICLES_CODEBOOK.PRG
  (CODEBOOK3)\__ARTICLES_SELECTION.PRG
  (CODEBOOK3)\__ARTICLES_DESCRIPTION.PRG
  (CODEBOOK3)\__ARTICLES_REGISTERCARD.PRG

// ARTICLES-CODEBOOK3.XPJ --- END


// ARTICLES-CODEBOOK3.BAT --- START
@ECHO OFF
PBUILD.EXE ARTICLES-CODEBOOK3.XPJ > ARTICLES-CODEBOOK3________.TXT
NOTEPAD ARTICLES-CODEBOOK3________.TXT
// ARTICLES-CODEBOOK3.BAT --- END
```

PRG FOR ARTICLES-CODEBOOK3.EXE

__ARTICLES__MAIN.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES__MAIN.PRG                              SQL            //
//                                                                    //
//   01-11-2023                                                       //
//                                                                    //
//   www.cobasystems.com --- COBA Systems --- Slobodan Stanojević Coba //
//   Open Source Project BAST Business Account Software Technology    //
//   www.Alaska-Software.com --- Alaska Xbase++ version 2.0.1503       //
//   www.Donnay-software.com --- eXpress++      version 2.0.268        //
//   Sergej Spirin --- FastReport for Xbase++   version 27.03.2015     //
//                                                                    //
//   Database Server PostgreSQL                version 9.4.4.         //
//                                                                    //
//                                                                    //
////////////////////////////////////////////////////////////////////////

/*
-------------------------------
  PACKAGE:
  __ARTICLES__MAIN.PRG
  __ARTICLES_CODEBOOK.PRG
  __ARTICLES_SELECTION.PRG
  __ARTICLES_DESCRIPTION.PRG
  __ARTICLES_REGISTERCARD.PRG
-------------------------------

 APPLICATION (main program)
 Alaska Xbase++ and eXpress++ PGDBE and  U P S I Z E  PostgreSQL DATABASE
 ŠIFARNIK SVIH ARTIKALA PRODAJNOG OBJEKTA ILI RESTORANA
 SA FILTEROM PO GRUPI ARTIKALA

 APPLICATION (main program)
 Alaska Xbase++ and eXpress++ PGDBE and  U P S I Z E  PostgreSQL DATABASE
 ARTICLES OF SALES OBJECT OR RESTAURANT
 WITH FILTER BY GROUP OF ARTICLES

-------------------------------
 PROCEDURE AppSys()
 PROCEDURE dbesys()
 PROCEDURE main()



-----
NOTE:
-----
```

```
OVA APLIKACIJA IGNORIŠE UPSIZE MAŠINU UGRAĐENU U PGDBE MAŠINU
I NE KORISTI NI JEDAN OD UPSIZE MEHANIZAMA, PA VELIKI NEOBAVEZNI DEO
UPSIZE MAŠINE NE MORA NI POSTOJATI U DATABAZI APLIKACIJE:
   - UPSIZE ISAM KOLONE U TABLAMA
   - UPSIZE ISAM KOLONE ZA ISAM INDEKSE U TABLAMA
   - UPSIZE ISAM INDEKSI U INDEKSIMA
   - TRIGERI I TRIGER FUNKCIJE ZA TE ISAM INDEKSE
   - OBAVEZNE PGDBE ISAM TABLE:
         "alaska-software.isam.orders"
         "alaska-software.isam.tables"
         "alaska-software.system.connections"
     moraju postojati jer ih PGDBE automatski formira,
     ali mogu biti prazne (bez redova).
OVA APLIKACIJA MOŽE KORISTITI IZVORNE-STANDARDNE POSTGRESQL TABLE BEZ IKAKVIH UPSIZE DODATAKA.
OVA APLIKACIJA ZA KOMUNIKACIJU SA BAZOM PODATAKA KORISTI PGDBE SQL KOMANDE
I FUNKCIJE, A TAKOĐE I JEDAN DEO ISAM KOMANDI I FUNKCIJA KOJE SU SIMULIRANE
KROZ PGDBE.


-----
NOTES:
-----

THIS APPLICATION IGNORES THE UPSIZE MACHINE BUILT INTO THE PGDBE MACHINE
AND IT DOESN'T USE ANY OF THE UPSIZE MECHANISMS, SO A LARGE OPTIONAL PART
MACHINE UPSIZE DOESN'T EVEN EXIST IN THE APPLICATION DATABASE:
    - UPSIZE ISAM COLUMNS IN TABLES
    - UPSIZE ISAM COLUMNS FOR ISAM INDEXES IN TABLES
    - UPSIZE ISAM INDEXES IN INDEXES
    - TRIGGER AND TRIGGER FUNCTIONS FOR THOSE ISAM INDEXES
    - MANDATORY PGDBE ISAM TABLES:
         "alaska-software.isam.orders"
         "alaska-software.isam.tables"
         "alaska-software.system.connections"
      they must exist because PGDBE creates them automatically,
      but can be empty (no rows).
THIS APPLICATION CAN USE THE NATIVE-STANDARD POSTGRESQL TABLES WITHOUT ANY UPSIZE PLUGINS.
THIS APPLICATION USES PGDBE SQL COMMANDS AND FUNCTIONS TO COMMUNICATE WITH
THE DATABASE, AND ALSO ONE PART OF ISAM COMMANDS AND FUNCTIONS THAT ARE
SIMULATED THROUGH PGDBE.


*/

*-----------------------------------
* Xbase++
#include "Xbp.ch"
#include "AppEvent.ch"
#include "Common.ch"
*-----------------------------------
* XbToolIII++
#include "xbtsys.ch"
*-----------------------------------
* eXpress++
#include "dcdialog.ch"
*-----------------------------------
* Xbase++ PGDBE
#include "pgdbe.ch"
#include "std.ch"
```

```
#include "sql.ch"
#include "dac.ch"
#pragma library("adac20b.lib")
*----------------------------------
* Xbase++ APPBROWSE, APPDISPLAY
* for testing the program:
  #include "Appbrow.ch"
  MEMVAR appObject
*----------------------------------


*************************************************************************
PROCEDURE AppSys()
*************************************************************************
 // prvo se poziva:
 // init() pa appsys() pa dbesys() pa main()
 // first call:
 // init() then appsys() then dbesys() then main()

 SET DATE GERMAN
 SET CENTURY ON
 SET CHARSET TO ANSI

 PUBLIC APLIKACIJA := "SQL"

* --- 1. spreči drugu instancu

* ako je ova aplikacija već startovana i u memoriji je
* a sada se startuje druga instanca - ovde se prekida sa QUIT

* if this application has already been started and is in memory
* and now the second instance is started - here it is terminated with QUIT

 *************
  xRUN_STOP() // BAZNE.DLL
 *************

* --- 1. spreči drugu instancu



RETURN
*************************************************************************
PROCEDURE dbesys()
*************************************************************************

   DbeLoad("pgdbe")
   DbeSetDefault("pgdbe")

RETURN
*************************************************************************
PROCEDURE MAIN()
*************************************************************************

  nOldIcon := DC_ICONDEFAULT(1) /* postavi ikonu programa */
  SETCANCEL(.F.)      /* Isključi prekid aplikacije sa Alt+C */
  DC_DotHotKey(180)   /* taster Alt+D, koji startuje DOT Prompt, prebaci na 180 */


  *================================================================ START
```

```
  * CONTROL APP DATABASE - CONNECT-CONTROL-DISCONNECT
  *====================================================================

  ************
  __pg__ini()  // C-PGDB.DLL
  ************
  *  daje databases PUBLIC variables iz "C-PGDB.INI"
  * _server_,_uid_,_pwd_,_db_,_she_,_tab_,_oSession_,
  * _dblist_,_shelist_,_tablist_,lAlert,lMsg,_lDBcontrol_
  * _MAG_,_KAS_,
  * _version_, _cConnStr_, cConnStrTest_ , _cdbc_



    IF _lDBcontrol_=.T. // C-PGDB.INI
      // vrši se kontrola PgSQL database
      // kontrola database or create database if not exists
        ************************
        qq := __pg__connect_app() // C-PGDB.DLL
        ************************
        * qq=.T. or qq=.F.
    ELSE
      // ne vrši se kontrola PgSQL database
        qq:=.T.
        *" N A S T A V A K  P R O G R A M A
        *" DATABAZA JE OK SVE TABLE SU OK
        *" CONTINUE PROGRAM
        *" DATABASE IS OK ALL TABLES IS OK
    ENDIF

    IF qq==.F.
       ***************
       __pg__appquit() // C-PGDB.DLL
       ***************
       *" P R E K I D  P R O G R A M A
       *" NEDOSTAJU TABLE U DATABAZI
       *" STOP PROGRAM
       *" TABLES IN DATABASE ARE MISSING
    ENDIF

  *====================================================================
  * CONTROL APP DATABASE - CONNECT-CONTROL-DISCONNECT
  *==================================================================== END




//------------------------------------------------------ START
// G R U P E - G R O U P S
//------------------------------------------------------
*
* NAZIVI 16 GRUPA UČITAVAJU SE IZ KASA.INI
* NAMES OF 16 GROUPS ARE LOADED FROM KASA.INI
* INI_READ(), INI_WRITE() -> BAZNE.DLL
*
PRIVATE inifil:= GDE_EXE()+"\KASA.INI"
PRIVATE;
  qq1  :=INI_READ("C","GRUPE","1"  ,inifil),;
  qq2  :=INI_READ("C","GRUPE","2"  ,inifil),;
```

```
  qq3  :=INI_READ("C","GRUPE","3"  ,inifil),;
  qq4  :=INI_READ("C","GRUPE","4"  ,inifil),;
  qq5  :=INI_READ("C","GRUPE","5"  ,inifil),;
  qq6  :=INI_READ("C","GRUPE","6"  ,inifil),;
  qq7  :=INI_READ("C","GRUPE","7"  ,inifil),;
  qq8  :=INI_READ("C","GRUPE","8"  ,inifil),;
  qq9  :=INI_READ("C","GRUPE","9"  ,inifil),;
  qq10 :=INI_READ("C","GRUPE","10" ,inifil),;
  qq11 :=INI_READ("C","GRUPE","11" ,inifil),;
  qq12 :=INI_READ("C","GRUPE","12" ,inifil),;
  qq13 :=INI_READ("C","GRUPE","13" ,inifil),;
  qq14 :=INI_READ("C","GRUPE","14" ,inifil),;
  qq15 :=INI_READ("C","GRUPE","15" ,inifil),;
  qq16 :=INI_READ("C","GRUPE","16" ,inifil)


IF EMPTY(qq1) // ako ne postoji upis u F1

/*
  INI_WRITE("C","GRUPE","1" ,"TOPLI;NAPITCI"    ,inifil)
  INI_WRITE("C","GRUPE","2" ,"VINA;VINJACI"     ,inifil)
  INI_WRITE("C","GRUPE","3" ,"ŽESTOKA;PIĆA"     ,inifil)
  INI_WRITE("C","GRUPE","4" ,"PIVA"             ,inifil)
  INI_WRITE("C","GRUPE","5" ,"RAKIJE"           ,inifil)
  INI_WRITE("C","GRUPE","6" ,"SOKOVI;GAZIRANO" ,inifil)
  INI_WRITE("C","GRUPE","7" ,"JELA;A LA CART"   ,inifil)
  INI_WRITE("C","GRUPE","8" ,"ROŠTILJ"          ,inifil)
  INI_WRITE("C","GRUPE","9" ,"RIBA"             ,inifil)
  INI_WRITE("C","GRUPE","10","SALATE"           ,inifil)
  INI_WRITE("C","GRUPE","11","PREDJELO"         ,inifil)
  INI_WRITE("C","GRUPE","12","PEČENJA"          ,inifil)
  INI_WRITE("C","GRUPE","13","ČORBE;SUPE"       ,inifil)
  INI_WRITE("C","GRUPE","14","DESERT;SLATKIŠI" ,inifil)
  INI_WRITE("C","GRUPE","15","SENDVIČI"         ,inifil)
  INI_WRITE("C","GRUPE","16","DORUČAK"          ,inifil)
*/

*                      GRUPA  NAME
  INI_WRITE("C","GRUPE","1" ,"HOT;DRINKS"         ,inifil)
  INI_WRITE("C","GRUPE","2" ,"WINES;COGNAC"       ,inifil)
  INI_WRITE("C","GRUPE","3" ,"STRONG;DRINKS"      ,inifil)
  INI_WRITE("C","GRUPE","4" ,"BEER"               ,inifil)
  INI_WRITE("C","GRUPE","5" ,"BRANDIES"           ,inifil)
  INI_WRITE("C","GRUPE","6" ,"JUICES;CARBONATED" ,inifil)
  INI_WRITE("C","GRUPE","7" ,"FOOD;A LA CART"     ,inifil)
  INI_WRITE("C","GRUPE","8" ,"BARBECUE"           ,inifil)
  INI_WRITE("C","GRUPE","9" ,"FISH"               ,inifil)
  INI_WRITE("C","GRUPE","10","SALADS"             ,inifil)
  INI_WRITE("C","GRUPE","11","APPETIZER"          ,inifil)
  INI_WRITE("C","GRUPE","12","ROASTING"           ,inifil)
  INI_WRITE("C","GRUPE","13","BROTH;SOUPS"        ,inifil)
  INI_WRITE("C","GRUPE","14","DESSERT;SWEET"      ,inifil)
  INI_WRITE("C","GRUPE","15","SANDWICHES"         ,inifil)
  INI_WRITE("C","GRUPE","16","BREAKFAST"          ,inifil)


  qq1  :=INI_READ("C","GRUPE","1"  ,inifil)
  qq2  :=INI_READ("C","GRUPE","2"  ,inifil)
  qq3  :=INI_READ("C","GRUPE","3"  ,inifil)
```

```
  qq4  :=INI_READ("C","GRUPE","4"  ,inifil)
  qq5  :=INI_READ("C","GRUPE","5"  ,inifil)
  qq6  :=INI_READ("C","GRUPE","6"  ,inifil)
  qq7  :=INI_READ("C","GRUPE","7"  ,inifil)
  qq8  :=INI_READ("C","GRUPE","8"  ,inifil)
  qq9  :=INI_READ("C","GRUPE","9"  ,inifil)
  qq10 :=INI_READ("C","GRUPE","10" ,inifil)
  qq11 :=INI_READ("C","GRUPE","11" ,inifil)
  qq12 :=INI_READ("C","GRUPE","12" ,inifil)
  qq13 :=INI_READ("C","GRUPE","13" ,inifil)
  qq14 :=INI_READ("C","GRUPE","14" ,inifil)
  qq15 :=INI_READ("C","GRUPE","15" ,inifil)
  qq16 :=INI_READ("C","GRUPE","16" ,inifil)

ENDIF

GRUPA := EUPIS0("  ","GROUP NO 1-16") // BAZNEX.DLL
GRUPA:=ALLTRIM(GRUPA)
IF VAL(GRUPA)<1 .OR. VAL(GRUPA)>16
   GRUPA := ""
   NAME  := ""
ELSE
   NAME:=&("qq"+GRUPA)
ENDIF


//---------------------------------------------------------
// G R U P E - G R O U P S
//--------------------------------------------------------- END


*" C O N N E C T
*" uspostavi konekciju na ulazu u main()
* lRet:=SQL_connection(.T.) // connect // PG_DATABASE_CONNECT.PRG
************************
lRet := __pg__connect()     // connect // C-PGDB.DLL
************************


* STOP("P R O G R A M  W O R K I N G",0)
      **********************************
      ARTICLES_CODEBOOK(GRUPA,NAME)
      **********************************

*" D I S C O N N E C T
*" prekini konekciju na izlazu iz main()
* lRet:=SQL_connection(.F.) // disconnect // PG_DATABASE_CONNECT.PRG
**************************
lRet := __pg__disconnect()  // disconnect // C-PGDB.DLL
**************************



****************
 __pg__AppQuit() // disconnect all sessions // C-PGDB.DLL
****************
RETURN
```

__ARTICLES__CODEBOOK.PRG

```
//////////////////////////////////////////////////////////////////////////
//                                                                      //
//                                                                      //
//   __ARTICLES_CODEBOOK.PRG                                SQL         //
//                                                                      //
//   01-11-2023                                                         //
//                                                                      //
//   www.cobasystems.com --- COBA Systems --- Slobodan Stanojević Coba  //
//   Open Source Project BAST Business Account Software Technology      //
//   www.Alaska-Software.com --- Alaska Xbase++ version 2.0.1503         //
//   www.Donnay-software.com --- eXpress++      version 2.0.268          //
//   Sergej Spirin --- FastReport for Xbase++   version 27.03.2015      //
//                                                                      //
//   Database Server PostgreSQL                 version 9.4.4.          //
//                                                                      //
//                                                                      //
//////////////////////////////////////////////////////////////////////////


/*
-------------------------------
  PACKAGE:
  __ARTICLES__MAIN.PRG
  __ARTICLES_CODEBOOK.PRG
  __ARTICLES_SELECTION.PRG
  __ARTICLES_DESCRIPTION.PRG
  __ARTICLES_REGISTERCARD.PRG
-------------------------------


Note:

A.
Aplikacija može da Koristi posgreSQL upsize table koje imaju upsize polja:

  __deleted boolean NOT NULL DEFAULT false,
  __record serial NOT NULL,
  __rowversion integer NOT NULL DEFAULT 1,
  __keyversion integer NOT NULL DEFAULT 0,
  __lock_owner integer NOT NULL DEFAULT 0,
  __order_1_1 character(36),
  __order_2_2 character(16),
  CONSTRAINT kroba_11_pkey PRIMARY KEY (__record)

Ali aplikacija ne koristi ta polja i ona ovde nemaju nikakvu svrhu.
Takođe, ne koristi ni ostale elemente upsize mašine:
 - ne koristi tablu "alaska-software.isam.orders"
 - ne koristi tablu "alaska-software.isam.tables"
 - ne koristi ISAM indekse
 - može da koristi ali ne koristi SQL indekse
Koristi sve mogućnosti PGDBE mašine:
 - sve passtrough SQL komande i funkcije
```

```
 - samo neke od ISAM komandi i funkcija

B.
Aplikacija Može da Koristi posgreSQL no-upsize table odnosno standardne
postgreSQL table, koje nemaju ova upsize polja.



Notes:

A.
An application can use posgreSQL upsize tables that have upsize fields:

   __deleted boolean NOT NULL DEFAULT false,
   __record serial NOT NULL,
   __rowversion integer NOT NULL DEFAULT 1,
   __keyversion integer NOT NULL DEFAULT 0,
   __lock_owner integer NOT NULL DEFAULT 0,
   __order_1_1 character(36),
   __order_2_2 character(16),
   CONSTRAINT kroba_11_pkey PRIMARY KEY (__record)

But the application doesn't use those fields and they have no purpose here.
Also, it does not use the other elements of the upsize engine:
  - don't use table "alaska-software.isam.orders"
  - do not use table "alaska-software.isam.tables"
  - does not use ISAM indexes
  - can use but not SQL indexes
It uses all the possibilities of the PGDBE machine:
  - all passthrough SQL commands and functions
  - just some of the ISAM commands and functions

B.
Can use posgreSQL no-upsize tables or standard ones
postgreSQL tables, which do not have these upsize fields.




 ŠIFARNIK SVIH ARTIKALA I FILTER PO GRUPI ARTIKALA
 ============================================================
 (FILTER PO GRUPA = DOBS_ = "1" do "16")
 BROWSER ARTIKALA - LISTA ARTIKALA
 ARTIKAL: DODAJ, OBRIŠI, EDITUJ, NAĐI, POŠALJI U APLIKACIJU

 ret .T. article taken
 ret .F. article not taken
 formira fajl:  ARTIKAL.CFG = "articlecode/articlebarcode"
   artikal_CFG_upisi1( cfg_asifra, cfg_abarcod )
   arttikal_CFG_citaj1()
   __ARTICLES_REGISTERCARD.PRG

 CODE BOOK OF ALL ARTICLES AND FILTER BY GROUP OF ARTICLES
 ============================================================
 (FILTER BY GROUP = DOBS_ = "1" to "16")
 BROWSER ARTICLES - LIST OF ARTICLES
 ITEM: ADD, DELETE, EDIT, FIND, SEND TO APPLICATION
```

```
 ret .T. article taken
 ret .F. article not taken
 create file:  ARTIKAL.CFG = "articlecode/articlebarcode"
   artikal_CFG_upisi1( cfg_asifra, cfg_abarcod )
   arttikal_CFG_citaj1()
   __ARTICLES_REGISTERCARD.PRG

 ============================================================
 ARTICLES_CODEBOOK()
 STATIC FUNCTION take_article(nn)
 STATIC FUNCTION Maintitle(cTitle)
 STATIC FUNCTION vidi_help(cTitle)

 STATIC FUNCTION ARTICLES_DESCRIPTION_(oBrowse)
 STATIC FUNCTION ARTICLES_REGISTERCARD_(lPar,GetList,oBrowse,GRUPA)

 STATIC FUNCTION find_name(oBrowse)
 STATIC FUNCTION find_code()
 STATIC FUNCTION find_BarCod()
 STATIC FUNCTION find_cataloque()

 STATIC FUNCTION find_word(oBrowse,oDlg)
 STATIC FUNCTION find_supplier(oBrowse,oDlg)
 STATIC FUNCTION find_link(oBrowse,oDlg)

 FUNCTION DELETE_ARTICLE(oBrowse,GetList)
 FUNCTION INSERT_ARTICLE(oBrowse,GetList)
   STATIC FUNCTION INSERT_ARTICLE_(oBrowse,GetList,cUpis)

 STATIC FUNCTION SORT_BY_NAME(cschemename,ctablename,cgrupa,oBrowse)
 STATIC FUNCTION SORT_BY_CODE(cschemename,ctablename,cgrupa,oBrowse)

* koristi se samo ako postoji polje (kolona)  __record
* used only if field (column) __record exists

   FUNCTION RENUMBERATION(table)      // for any upsize table
   FUNCTION RECONSTRUCTION(table)     // for any upsize table
   FUNCTION RENUMBER__record(table)   // for any upsize table




*/


#include "Xbtsys.ch"   // XbttoolsIII
#include "Appevent.ch"
#include "Common.ch"
#include "Xbp.ch"
#include "dll.ch"
#include "dcdialog.ch"
* #include "appbrow.ch" // for test

*********************************************************************
FUNCTION ARTICLES_CODEBOOK(GRUPA,NAME)
*********************************************************************

* GRUPA = grupa artikala - a group of articles
* NAME  = naziv grupe - group name
```

```
LOCAL radno := SELECT() // na izlazu iz programa SELECT(radno)
                        // on program exit return SELECT(radno)

LOCAL GetList := {}, GetOptions, oDlg, oBrowse, oBrowBox, ;
      oToolBar, bBrojSlogova, bTitle, ;
      cTitle := "ARTICLES CODE BOOK"
      cTitleList := cTitle
      cTit:=" SQL TABLE "

DEFAULT GRUPA TO "", NAME TO ""
GRUPA := alltrim(GRUPA)
NAME  := alltrim(NAME)
PRIVATE xGRUPA := GRUPA, xNAME:=NAME

IF EMPTY(GRUPA)
   cTitleList := "ARTICLES CODE BOOK"
ELSE
   cTitleList := strtran(NAME,";"," ") // ARTICLES GROUP NAME
ENDIF


*********************************
PUBLIC lPreuzet_je_artikal := .F.  // the article has been taken
*********************************
// IZLAZ IZ FUNKCIJE SA ESC ili sa Close -> lPreuzet_je_artikal := .F.
// IZLAZ IZ FUNKCIJE SA ESC ili sa Close -> isprazni ARTIKAL.CFG
// EXIT THE FUNCTION WITH ESC or with Close -> lPreuzet_je_artikal := .F.
// EXIT THE FUNCTION WITH ESC or with Close -> empty ARTIKAL.CFG



    ********************************************************
       PRIVATE BMP_OK, BMP_HLP, BMP_ESC, aCUR

       aCUR := {"USER32.DLL",114}

       BMP_OK := XbpBitmap():new():create()
       BMP_OK:load( "BAZNE.DLL", 11041 )
       BMP_OK:transparentClr := BMP_OK:getDefaultBgColor()

       BMP_HLP := XbpBitmap():new():create()
       BMP_HLP:load( "BAZNE.DLL", 11043 )
       BMP_HLP:transparentClr := BMP_HLP:getDefaultBgColor()

       BMP_ESC := XbpBitmap():new():create()
       BMP_ESC:load("BAZNE.DLL",11042 )
       BMP_ESC:transparentClr := BMP_ESC:getDefaultBgColor()
    ********************************************************




//=========================================================================  START
// T A B L E  N A M E S
//=========================================================================

    PRIVATE SPISAK_ROBE :="kroba_11"
```

```
    PRIVATE ;
    MAG         := _MAG_ ,; // __pg__ini() C-PGDB.DLL
    KAS         := _KAS_ ,; // __pg__ini() C-PGDB.DLL
    cSchemeName := _she_ ,; // __pg__ini() C-PGDB.DLL
    cTablename  := lower(SPISAK_ROBE) ,;
    oSession    := DbSession()

//=========================================================================
// T A B L E  N A M E S
//=========================================================================  END



//=========================================================================  START
// D C B R O W S E  C O N T E N T  F R O M  P O S T G R E S Q L  T A B L E
//=========================================================================

*----------------------------
* bez indeksa
* without index
*----------------------------

    cSchemeName := _she_          // 'public'
    cTablename  := SPISAK_ROBE    // 'kroba_11'
    GRUPA       := ALLTRIM(GRUPA) // '8'



* --- STEP 1 --- SQL FILTERS BY GROUP --- PGDBE SQL


IF EMPTY(GRUPA)

   *" NO FILTER

    cSQL    := "SELECT * FROM "+cschemename+"."+ctablename+" ORDER BY robn_ ;"
    oStmt   := DacSqlStatement():fromChar(cSQL)
    cAlias  := oStmt:build():query(,"ART")

ELSE

   *" YES FILTER

    cSQL    := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE dobs_='"+GRUPA+"' ORDER BY
robn_ ;"
    oStmt   := DacSqlStatement():fromChar(cSQL)
    cAlias  := oStmt:build():query(,"ART")

ENDIF


    ART:=cAlias
    SELECT "ART"         // this is mandatory  !
    svega:=reccount()    // this works correctly !
    * count all to svega // this works correctly !
    GO TOP
```

```
*------
* NOTE:
*------
*
*"OVO OVDE NE FUNKCIONIŠE - THIS DOESN'T WORK HERE:
* SELECT "ART"
* INDEX ON ROBN_ TO 1
* INDEX ON ROBS_ TO 2
* SET INDEX TO 1,2
* U aplikaciji se ne mogu koristiti ISAM komande:
*    SEEK, SET SOFTSEEK, SET ORDER...
* cannot be used in the application ISAM commands:
*    SEEK, SET SOFTSEEK, SET ORDER...
*"OVO OVDE NE FUNKCIONIŠE - THIS DOESN'T WORK HERE:
*
* ---
*
* Šta se ovde gubi a šta se dobija u odnosu na staru DBF-NTX
* aplikaciju ili u odnosu na UPSIZE DBF aplikaciju:
* - Gubi se rad sa indeksima i mogućnost korišćenja
*   SEEK, SOFTSEEK, ORDER
* - Gubi se mogućnost sortiranja spiska po indeksu SET ORDER
* - Gubi se mogućnost upotrebe GO TOP LOCATE
* - sve ovo se može rešiti i na SQL način ali potrebne su veće
*   izmene u aplikaciji
* - Dobija se mnogo brže formiranje DCBROWSE liste (po filteru za
*   grupu) u odnosu na brzinu formiranja DCBROWSE liste kod ISAM
*   komande: USE kroba_11
* - Dobija se spisak sortiran po nazivu, a sortiranje po šifri dobija
*   se generisanjem nove DCBROWSE liste sa ORDER BY robs_ - što se
*   desi trenutno
* - Od stare DBF aplikacije zadržan je samo interfejs, a poslovna
*   logika i komunikacija sa databazom prebačeni su na SQL
*
* ---
*
* What is lost here and what is gained compared to the old DBF-NTX
* application or relative to the UPSIZE DBF application:
* - Working with indexes and the possibility of using SEEK, SOFTSEEK,
*   ORDER are lost
* - The ability to sort the list by the SET ORDER index is lost
* - The ability to use GO TOP LOCATE is lost
* - all this can be solved in the SQL way, but major changes are needed
*   in the application
* - Much faster formation of the DCBROWSE list is obtained (per group
*   filter)
*   in relation to the speed of forming the DCBROWSE list with the ISAM
*   command: USE kroba_11
* - A list sorted by name is obtained, and sorting by code is obtained
*   by generating a new DCBROWSE list with ORDER BY robs_ - which
*   happens now
* - From the old DBF application, only the interface has been retained,
*   and the business logic as well communication with the database has
*   been transferred to SQL

//=========================================================================
// D C B R O W S E  C O N T E N T  F R O M  P O S T G R E S Q L  T A B L E
//=========================================================================  END
```

```
//========================================================================  START
// DCBROWSE LISTA SA ISAM KOMANDAMA PREUZETA IZ STARE DBF-NTX APLIKACIJE
// DCBROWSE LIST WITH ISAM COMMANDS TAKEN FROM OLD DBF-NTX APPLICATION
//========================================================================


*****************
SELECT "ART"
*****************


of2  := Font_codepage(14,"Verdana Bold",238,.t.)
of3  := Font_codepage(12,"Archivo Narrow",238,.t.)
of4  := Font_codepage(11,"Consolas",238,.t.)

HOR := 62
VER := 15 + 8.8

PUBLIC preuzmi_naziv := 0

@  0,2 DCSAY cTitleList SAYFONT of2  SAYCOLOR GRA_CLR_DARKRED SAYSIZE 0
@  0,51 DCSAY cTit SAYFONT "10.Verdana Bold"  ;
SAYCOLOR GRA_CLR_WHITE,GraMakeRGBColor({156,139,18}) SAYSIZE 0

BROWSER_COLOR() // bazne.dll

@  1,2 DCSTATIC TYPE XBPSTATIC_TYPE_RECESSEDBOX SIZE HOR,VER ;
   OBJECT oBrowBox

@ .1,.5 DCBROWSE oBrowse PARENT oBrowBox ALIAS "ART" ;
   SIZE HOR-1,VER-0.4 ;
   CURSORMODE XBPBRW_CURSOR_ROW ;
   HEADLINES 2 ;
   ITEMSELECTED {|| take_article(1),;  // here
                    SetAppFocus(oBrowse) }  ;
   EVAL {|o|o:setfont(of3)}




//-----------------------------------------
// K O L O N E  S P I S K A  A R T I K A L A
// COLUMNS FOR THE LIST OF ITEMS
//-----------------------------------------



DCBROWSECOL FIELD ART->ROBN_ WIDTH 20 HEADER "ARTICLE NAME"   ;
            PARENT oBrowse  TOOLTIP " NAZIV ARTIKLA | ARTICLE NAME "

DCBROWSECOL FIELD ART->MERA_ WIDTH 3  HEADER "JMR" ;
            PARENT oBrowse TOOLTIP " JEDINICA MERE | UNIT OF MEASUREMENT "

DCBROWSECOL FIELD ART->ROBS_ WIDTH 4  HEADER "CODE" ;
            PARENT oBrowse TOOLTIP " ŠIFRA ARTIKLA | ARTICLE CODE "


DCBROWSECOL FIELD ART->CENA_MPR_ ;
```

```
  WIDTH 7;
  HEADER "PRICE" PARENT oBrowse TOOLTIP " MALOPRODAJNA CENA | RETAIL PRICE "

DCBROWSECOL FIELD ART->ROBK_ ;
  WIDTH 15 ;
  HEADER "BARCOD" PARENT oBrowse TOOLTIP " BARCOD EAN13 | GTIN EAN13 "


DCBROWSECOL FIELD ART->GRUP_ ;
  WIDTH 4 ;
  HEADER "H-FOOD  ;P-DRINK" PARENT oBrowse TOOLTIP " H-HRANA P=PIĆE | H=FOOD P=DRINK "

DCBROWSECOL FIELD ART->DOBS_ ;
  WIDTH 4 ;
  HEADER "GRUPA;1-16" PARENT oBrowse TOOLTIP " GRUPA 1 DO 16 | GROUP 1 TO 16 "


DCBROWSECOL FIELD ART->ROBN_ WIDTH 20 HEADER "ARTICLE NAME"   ;
            PARENT oBrowse

//-----------------------------------------
// COLUMNS FOR THE LIST OF ITEMS
// K O L O N E  S P I S K A  A R T I K A L A
//-----------------------------------------



//-------------------------------------------------------------------------------------
// TOOLBAR COMMAND BUTTON:
//-------------------------------------------------------------------------------------

d_tol := HOR
d_bro := 6
d_duz := d_tol/d_bro

of3  := Font_codepage(11,"Archivo Narrow",238,.t.)

@ VER+2,2 DCTOOLBAR oToolBar SIZE d_tol,3 BUTTONSIZE d_duz,3 FONT of3



DCADDBUTTON CAPTION BMP_OK ;
   TOOLTIP "Preuzmi podatke artikla - Download item data" ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {|| take_article(1),;  // here
   SetAppFocus(oBrowse)};
   ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Name;F2" ;
   ACCELKEY {xbeK_F2, asc("N"), asc("n")} ;
   TOOLTIP " Nađi po Nazivu - Find by Name " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {|| find_name(oBrowse), ;  // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse)} ;
   ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Code;F3" ;
   ACCELKEY {xbeK_F3, asc("C"), asc("c")};
   TOOLTIP " Nađi po Šifri - Find by Code " ;
   CURSOR aCUR PARENT oToolBar ;
```

```
   ACTION {||find_code(oBrowse), ; // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
   ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Word;F4" ;
   ACCELKEY {xbeK_F4, asc("W"), asc("w")};
   TOOLTIP " Nadji po reči sadržanoj bilo gde u nazivu ;"+;
           " Find by word contained anywhere in the name " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {||find_word(oBrowse,oDlg),; // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
   ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Barcod;F5" ;
   ACCELKEY {xbeK_F5, asc("B"), asc("b")};
   TOOLTIP " Nađi po Barcodu - Find by Barcode " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {||find_barcod(oBrowse),;  // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
   ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Catalog;F6" ;
   ACCELKEY {xbeK_F6, asc("Q"), asc("q")};
   TOOLTIP " Nađi po katalo&kom broju - Find by catalog number " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {||find_cataloque(oBrowse),; // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
   ALIGNCAPTION BS_MULTILINE


*---
@ VER+2+3,2 DCTOOLBAR oToolBar SIZE d_tol,3 BUTTONSIZE d_duz,3  FONT of3
*---


DCADDBUTTON CAPTION "Supplier;F7"  ;
   ACCELKEY {xbeK_F7, asc("S"), asc("s")};
   TOOLTIP " Nađi po Dobavljaču - find by supplier " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {||find_supplier(oBrowse,oDlg),; // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION "Links;F8"  ;
   ACCELKEY {xbeK_F8, asc("L"), asc("l")};
   TOOLTIP " Nađi po Vezi - Find by links " ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {||find_link(oBrowse,oDlg),; // here
   DC_GetRefresh(GetList), SetAppFocus(oBrowse) } ;
ALIGNCAPTION BS_MULTILINE


**********************
PRIVATE RESTART := .F.
**********************

DCADDBUTTON CAPTION "Descript;F9" ;
   ACCELKEY {xbeK_F9, asc("D"), asc("d")};
   TOOLTIP " Opis artikla - description article " ;
   CURSOR aCUR PARENT oToolBar ;
```

```
   ACTION {|| ARTICLES_DESCRIPTION_(oBrowse)} ; // here
   ALIGNCAPTION BS_MULTILINE

* Varijanta sa ponovnim kreiranjem browsera - update data
* Variant with re-creation of the browser   - update data
* ACTION {|| RESTART:=.F., ;
* ARTICLES_DESCRIPTION_(oBrowse),;  // --> RESTART:=.T. or .F.
* IIF( RESTART=.T., DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList), SetAppFocus(oBrowse) )}



DCADDBUTTON CAPTION "Article;F10" ;
   ACCELKEY {xbeK_F10, asc("A"), asc("a")} ;
   TOOLTIP "Registar kartica artikla - Article card register" ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {|| ARTICLES_REGISTERCARD_(.F.,GetList,oBrowse,GRUPA) };  // here
   ALIGNCAPTION BS_MULTILINE

* Varijanta sa ponovnim kreiranjem browsera - update data
* Variant with re-creation of the browser   - update data
* ACTION {|| RESTART:=.F., ;
* ARTICLES_REGISTERCARD_(.F.,GetList,oBrowse,GRUPA),;  // --> RESTART:=.T. or .F.
* IIF( RESTART=.T., DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList), SetAppFocus(oBrowse) )}


// NO WINDOWSXP.MANIFEST jer BMP gubi transparentnost
// NO WINDOWSXP.MANIFEST because BMP it loses transparency
DCADDBUTTON CAPTION BMP_HLP;
   ACCELKEY xbeK_F1  ;
   TOOLTIP "Help;F1" ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {|| vidi_help(cTitle, cTitleList),;
   DC_GetRefresh(GetList), SetAppFocus(oBrowse)} ;
ALIGNCAPTION BS_MULTILINE

DCADDBUTTON CAPTION BMP_ESC ;
   ACCELKEY xbeK_ESC;
   TOOLTIP "Kraj;Esc" ;
   CURSOR aCUR PARENT oToolBar ;
   ACTION {|| take_article(0), DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList)} ;
ALIGNCAPTION BS_MULTILINE


* STEALT BUTTONS
DCHOTKEY xbeK_ALT_F1 ACTION {|| xc_autor1() }


DCHOTKEY xbeK_ALT_DEL ACTION {|| RECONSTRUCTION(cTableName,oBrowse,1),SetAppFocus(oBrowse) }
// here


DCHOTKEY xbeK_ALT_F2 ACTION {||
SORT_BY_NAME(cschemename,ctablename,grupa,oBrowse),SetAppFocus(oBrowse) }  // here
DCHOTKEY xbeK_ALT_F3 ACTION {||
SORT_BY_CODE(cschemename,ctablename,grupa,oBrowse),SetAppFocus(oBrowse) }  // here


IF GRUPA==""
   DCHOTKEY xbeK_INS ACTION {|| INSERT_ARTICLE(oBrowse,GetList),;
   DC_GetRefresh(GetList), oBrowse:RefreshAll(), SetAppFocus(oBrowse) }
```

```
   *---
   DCHOTKEY xbeK_DEL ACTION {|| DELETE_ARTICLE(oBrowse,GetList),;
   DC_GetRefresh(GetList), oBrowse:RefreshAll(), SetAppFocus(oBrowse) }
ELSE
   DCHOTKEY xbeK_INS ACTION {|| c__greska("ADDING ARTICLE TO GROUP"     +chr(59)+;
                                         "  IS NOT POSSIBLE"          +chr(59)+;
                                         "ADD ARTICLEE IS POSSIBLE IN" +chr(59)+;
                                         "  "+cTitle                  +chr(59)+;
                                         "","ADD ARTICLE",,"G3"),;
   SetAppFocus(oBrowse) }
   *---
   DCHOTKEY xbeK_DEL ACTION {|| c__greska("DELETING ARTICLE FROM GROUP"  +chr(59)+;
                                         "  IS NOT POSSIBLE"           +chr(59)+;
                                         "DELETE ARTICLE IS POSSIBLE IN"+chr(59)+;
                                         "  "+cTitle                   +chr(59)+;
                                         "","DELETE ARTICLE",,"G3"),;
   SetAppFocus(oBrowse) }
ENDIF
//-------------------------------------------------------------------------------------


aahorver:=appdesktop():currentsize()
ahor:=aahorver[1]-(HOR*7)-40
aver:=aahorver[2]-(VER*20)-242



DCGETOPTIONS ICON 1;
   SAYCENTER ;
   AUTOFOCUS ;
   NOMAXBUTTON ;
   NOMINBUTTON ;
   NOESCAPEKEY ;
   WINDOWROW aver WINDOWCOL ahor

 bTitle := {|| Maintitle(cTitle) }

 DCREAD GUI TITLE bTitle OPTIONS GetOptions ;
            FIT ;
            SETFOCUS @oBrowse ;
              PARENT @oDlg ;
              EVAL {|o|SetAppWindow(o) };
              MODAL

SELECT "ART"

IF RESTART==.T.

    //  stop(2,restart,alias(),1)
     SELECT "ART";USE;SELECT(radno)
     CLEAR TYPEAHEAD
     ****************************
      ARTICLES_CODEBOOK(GRUPA,NAME)
     ****************************
ELSE
     // stop(3,restart,alias(),1)
     SELECT "ART"
     USE
ENDIF
```

```
SELECT(radno)
RETURN(lPreuzet_je_artikal)  // .T./.F. the article has been taken



* STATIC FUNCTION take_article(nn)
* STATIC FUNCTION Maintitle(cTitle)
* STATIC FUNCTION vidi_help(cTitle)



*********************************************************************
STATIC FUNCTION take_article(nn)
*********************************************************************
LOCAL radno := SELECT()

IF nn=0
 *********************************
 artikal_CFG_upisi1( "", "" ) // ARTICLE_REGISTERCARD.PRG
 *********************************
   SELECT(radno)
   RETURN NIL
ENDIF

 *********************************
 artikal_CFG_upisi1( ROBS_, ROBK_ ) // ARTICLE_REGISTERCARD.PRG
 *********************************
 * Ažuriran je fajl ARTIKAL.CFG
 * The ARTIKAL.CFG file has been updated
 * Aplikacija uzima artikal sa: artikal_CFG_citaj1()
 * The application takes the article from: article_CFG_citaj1()

c__procitaj(;
"ARTIKAL JE POSLAT U ARTIKAL.CFG FILE"  +chr(59)+;
"ARTICLE HAS SENT TO ARTIKAL.CFG FILE"  +chr(59)+;
"Code       "+ROBS_                     +chr(59)+;
"Name       "+ROBN_                     +chr(59)+;
"Barcod     "+ROBK_                     +chr(59)+;
"Group      "+DOBS_                     +chr(59)+;
"Type       "+GRUP_                     +chr(59)+;
"","SEND ARTICLE TO APPLICATION",,"G3")
SELECT(radno)
*"POSLE SLANJA STAVKE ARTIKLA U APLIKACIJU NE ZATVARA SE ŠIFARNIK
*"AFTER SENDING ARTICLE INTO APPLICATION, CODEBOOK DOES NOT CLOSE
RETURN NIL



*********************************************************************
STATIC FUNCTION Maintitle(cTitle)
*********************************************************************
// ispis naslova u TITLE BAR-u sa brojem slogova
// print title in TITLE BAR with number of records
  LOCAL cBrojac, cNaslov
  * svega = broj redova table sa početka programa
  * svega = number of table rows from the beginning of the program

*" U OVOM ŠIFARNIKU NEMA PRIKAZA IZMENE BROJA REDOVA
```

```
*" THIS CODE BOOK DOES NOT SHOW THE CHANGE OF THE NUMBER OF ROWS
* COUNT ALL TO cBrojac  // reccount() not works
* GO TOP

    cNaslov := ;
    APLIKACIJA+"  |  KASH-REGISTER "+MAG+"-"+KAS+"  |  ARTICLES" + "  " +;
    var2char(svega)

RETURN(cNaslov)



*********************************************************************
STATIC FUNCTION vidi_help(cTitle)
*********************************************************************
LOCAL cr := chr(59), ctxt

ctxt := ;
"Sa ovog SPISKA ARTIKALA"                       +cr+;
cTitleList                                      +cr+;
"vrši se izbor artikla traženjem artikla"       +cr+;
"po nazivu, po šifri artikla, po barkodu,"      +cr+;
"po kataloškom broju, po reči u nazivu..."      +cr+;
"i slanje izabranog artikla u Dokument,"        +cr+;
"u koji se upisuju podaci o artiklu:"           +cr+;
"ili duplim klikom na artikal ili sa Enter"     +cr+;
"_____________________________________________"+cr+;
" "                                             +cr+;
"From this LIST OF ARTICLES"                    +cr+;
cTitleList                                      +cr+;
"article selection is made by searching for"    +cr+;
"article by name, by code, by barcode, by  "    +cr+;
"catalog number, by word anywhere in name..."   +cr+;
"and sending selected article to Document"      +cr+;
"in which data about article is entered:"       +cr+;
"or double click on article or with Enter."     +cr+;
"_____________________________________________"+cr+;
" "                                             +cr+;
" Article is added to the List  [INSERT]"       +cr+;
" Article is deleted from List  [DELETE]"       +cr+;
" Reconstruction List           [ALT]+[DEL]"    +cr+;
" Sorting List by article name  [ALT]+[F2]"     +cr+;
" Sorting List by article code  [ALT]+[F3]"     +cr+;
" "

* alertbox(,ctxt,{"    OK    "},,cTitle,"10.Consolas") // bazne.dll
c__procitaj(ctxt,"ARTICLES CODE BOOK",,"10.Lucida Console")  // bazne.dll


CLEAR TYPEAHEAD
RETURN(nil)



* STATIC FUNCTION ARTICLES_DESCRIPTION_(oBrowse)
* STATIC FUNCTION ARTICLES_REGISTERCARD_(lPar,GetList,oBrowse,GRUPA)
* STATIC FUNCTION find_name(oBrowse)
* STATIC FUNCTION find_code()
* STATIC FUNCTION find_word(oBrowse,oDlg)
```

```
* STATIC FUNCTION find_BarCod()
* STATIC FUNCTION find_cataloque()
* STATIC FUNCTION find_supplier(oBrowse,oDlg)
* STATIC FUNCTION find_link(oBrowse,oDlg)


*************************************************************************
STATIC FUNCTION ARTICLES_DESCRIPTION_(oBrowse)
*************************************************************************
* IF RLOCK() // use exclusive
   // ARTICLES_DESCRIPTION.PRG
   ARTICLES_DESCRIPTION(oBrowse)
*  UNLOCK
* ENDIF
RETURN NIL

*************************************************************************
STATIC FUNCTION ARTICLES_REGISTERCARD_(lPar,GetList,oBrowse,GRUPA)
*************************************************************************
* lPar =.F. editovanje zabranjeno | editing prohibited
*IF RLOCK() // use exclusive
   // ARTICLES_REGISTERCARD.PRG
   ARTICLES_REGISTERCARD(lpar,GetList,oBrowse,GRUPA)
*   UNLOCK
*ENDIF
RETURN NIL

*************************************************************************
STATIC FUNCTION find_name(oBrowse)
*************************************************************************
   xUpis := space(20)
   cUpis := Enter_key("Traži po nazivu","",xUpis,,"top")

   GO TOP
   LOCATE FOR ALLTRIM(ROBN_)=ALLTRIM(cUpis)
   IF FOUND()=.F.
      * confirmbox(setappwindow(),"Nije nađeno",;
      * "Not Found",XBPMB_OK,XBPMB_CRITICAL)
      * c_poruka("NIJE NAĐENO","Not Found")
      * stop("NIJE NAĐENO",0)

      c__greska("NIJE NAĐENO","Not Found")
      GO TOP
   ENDIF

CLEAR TYPEAHEAD
RETURN NIL
*************************************************************************
STATIC FUNCTION find_code()
*************************************************************************
   xUpis := space(5)
   cUpis := Enter_key("Šifra","",xUpis,,"top")
   cUpis := STRZERO(VAL(cUpis),5)

   GO TOP
   LOCATE FOR ALLTRIM(ROBS_)=ALLTRIM(cUpis)
   IF FOUND()=.F.
      c__greska("NIJE NAĐENO","Not Found")
      GO TOP
   ENDIF
```

```
CLEAR TYPEAHEAD
RETURN NIL
*****************************************************************************
STATIC FUNCTION find_barcod()
*****************************************************************************
LOCAL re := RecNo(), xUpis, cUpis, nUpis

   xUpis := space(13)
   cUpis := Enter_key("Traži po BarCodu","",xUpis,,"top")
   cUpis := ALLTRIM(cUpis)
   nUpis := LEN(cUpis)

   GO TOP
   LOCATE FOR SUBSTR(ROBK_,1,nUpis) == cUpis
   IF FOUND()=.F.
      c__greska("NIJE NAĐENO","Not Found")
      GO TOP
   ENDIF

CLEAR TYPEAHEAD
RETURN NIL
*****************************************************************************
STATIC FUNCTION find_cataloque()
*****************************************************************************
LOCAL re := RecNo(), xUpis, cUpis, nUpis

   xUpis := space(25)
   cUpis := Enter_key("Traži po Kataloškom broju","",xUpis,,"top")
   cUpis := ALLTRIM(cUpis)
   nUpis := LEN(cUpis)

   GO TOP
   LOCATE FOR SUBSTR(DOBK_,1,nUpis) == cUpis
   IF FOUND()=.F.
      c__greska("NIJE NAĐENO","Not Found")
      GO TOP
   ENDIF

CLEAR TYPEAHEAD
RETURN NIL
*****************************************************************************
STATIC FUNCTION find_word(oBrowse,oDlg)
*****************************************************************************
// Otvorena radna tabla SELECT "ART" - Open working table SELECT "ART":
LOCAL radno:=SELECT()
LOCAL selekcija := Enter_key("Find by part of name","",space(20),,"top")
      selekcija := ALLTRIM(selekcija)
      * selekcija = PART OF NAME -> "ROBN_"
      ***********************************************
      ARTICLES_SELECTION(selekcija,"ROBN_",oDlg,,xGRUPA)
      ***********************************************
* ARTICLES_SELECTION.PRG
* formiraj listu svih artikla koji imaju reč 'selekcija' u nazivu artikla
* create a list of all articles that have word 'selection' in article name
* No case-sensitive
SELECT(radno)
RETURN(nil)

*****************************************************************************
```

```
STATIC FUNCTION find_supplier(oBrowse,oDlg)
*****************************************************************
// Otvorena radna tabla SELECT "ART" - Open working table SELECT "ART":
LOCAL radno:=SELECT()
LOCAL selekcija := ALLTRIM(DOBS_)
      * selekcija = SUPPLIER CODE -> "DOBS_"
      ***********************************************
      ARTICLES_SELECTION(selekcija,"DOBS_",oDlg,,xGRUPA)
      ***********************************************
* ARTICLES_SELECTION.PRG
* formiraj listu svih artikala koji imaju šifru dobavljača = 'selekcija'
* create a list of all articles that have supplier code = 'selection'
SELECT(radno)
RETURN(nil)

*****************************************************************
STATIC FUNCTION find_link(oBrowse,oDlg)
*****************************************************************
// Otvorena radna tabla SELECT "ART" - Open working table SELECT "ART":
LOCAL radno:=SELECT()
LOCAL selekcija := ALLTRIM(VEZA_)
      * selekcija = LINK CODE -> "VEZA_"
      ***********************************************
      ARTICLES_SELECTION(selekcija,"VEZA_",oDlg,,xGRUPA)
      ***********************************************
* ARTICLES_SELECTION.PRG
* formiraj listu svih artikala koji imaju šifru veze = 'selekcija'
* create a list of all articles that have link code = 'selection'
SELECT(radno)
RETURN(nil)
```

```
* FUNCTION DELETE_ARTICLE(oBrowse,GetList)
* FUNCTION INSERT_ARTICLE(oBrowse,GetList)
* STATIC FUNCTION INSERT_ARTICLE_(oBrowse,GetList,cUpis)
* FUNCTION RENUMBERATION()
* FUNCTION RECONSTRUCTION()
* FUNCTION RENUMBER__record()

*****************************************************************
FUNCTION DELETE_ARTICLE(oBrowse,GetList)
*****************************************************************
LOCAL radno:=SELECT()
LOCAL xCode := robs_, yCode:=robs_, xName := robn_ , cRec
*
* --- tehnika:
* uvijek-odaberi-susednu-stavku (nakon brisanja stavke iz browsera)
* --- technique:
* always-select-adjacent-item (after deleting the item from the browser)
*
 SKIP 1
 IF EOF()
     SKIP -1
     SKIP -1
```

```
     cRec:=robs_
     // stop(ROBS_,"GORE",1)
     SKIP 1
     // stop(ROBS_,"DEL",1)
 ELSE
     cRec:=robs_
     // stop(ROBS_,"DOLE",1)
     SKIP -1
     // stop(ROBS_,"DEL",1)
 ENDIF
// ISAM neće da učita __record
// ISAM will not load __record
* --- technique:
* always-select-adjacent-item (after deleting the item from the browser)

IF c__sifra(2,"password=22")=.F.
   RETURN NIL
ENDIF

IF c__neda("Briše se iz registra-spiska artikala"  +chr(59)+;
           "It is deleted from list of articles:"  +chr(59)+;
           " "                                     +chr(59)+;
           xCode+" "+xName                         +chr(59)+;
           "","DELETION",,"G3")=.F.
           CLEAR TYPEAHEAD
           RETURN NIL
ENDIF



   SELECT(radno)
  * DELETE  // now the __deleted column is set to TRUE // Error
  * COMMIT                                            // Error


   // remove from table ROW with column __deleted=TRUE
     xCode := "'"+xCode+"'"

     IF empty(yCode)
       // ako nema šifre artikla - if no article code
        cSQL := "DELETE FROM ONLY "+cTablename+" WHERE robs_ IS NULL;"
     ELSE
        cSQL := "DELETE FROM ONLY "+cTablename+" WHERE robs_="+xCode+";"
     ENDIF
     oStmt := DacSqlStatement():fromChar(cSQL)
     oStmt:build():execute()
    * ---
     cSQL := "VACUUM FULL "+cTablename+";"
     oStmt := DacSqlStatement():fromChar(cSQL)
     oStmt:build():execute()

   SELECT(radno) // SELECT "ART"


   * --- technique:
   * always-select-adjacent-item (after deleting the item from the browser)
   oBrowse:refreshall()
    GO TOP
    LOCATE FOR alltrim(robs_)=alltrim(cRec)
    IF FOUND()==.F.
```

```
      GO TOP
    ENDIF
   oBrowse:refreshall()
   * --- technique:
   * always-select-adjacent-item (after deleting the item from the browser)

   c__poruka("DELETED","OK",,"G3")


RETURN NIL



*************************************************************************
FUNCTION INSERT_ARTICLE(oBrowse,GetList)
*************************************************************************
LOCAL radno:=SELECT()

   LOCAL xUpis := space(5)  // broj znakova za upis
   LOCAL cUpis := Enter_key("New Code","",xUpis,,"top")
   IF EMPTY(cUpis)
      RETURN NIL
   ENDIF
   cUpis := STRZERO(VAL(cUpis),5)


   GO TOP
   LOCATE FOR ALLTRIM(ROBS_)=ALLTRIM(cUpis)
   IF FOUND()
      c__poruka("ARTICLE EXISTS","OK",,"G3")
      CLEAR TYPEAHEAD
      RETURN NIL
   ENDIF


***********************************
INSERT_ARTICLE_(oBrowse,GetList,cUpis)
***********************************

RETURN NIL


*************************************************************************
STATIC FUNCTION INSERT_ARTICLE_(oBrowse,GetList,cUpis)
*************************************************************************
LOCAL radno:=SELECT()
PRIVATE ROBS, ROBN, DOBS,  cSQL, oStmt

ROBS := cUPIS
ROBN := " *** NEW NAME "+ROBS+" ***"
DOBS := xGRUPA

TEXT INTO cSQL WRAP
INSERT INTO &cTablename ( ROBS_, ROBN_, DOBS_ ) VALUES ('&ROBS', '&ROBN', '&DOBS' );
ENDTEXT
// dc_memoedit(cSQL)

oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
```

```
SELECT "ART"
DBREFRESH()
GO TOP
LOCATE FOR ROBS_==ROBS
DC_GetRefresh(GetList)
oBrowse:RefreshAll()

SELECT(radno)
RETURN NIL



* STATIC FUNCTION SORT_BY_NAME(cschemename,ctablename,cgrupa,oBrowse)
* STATIC FUNCTION SORT_BY_CODE(cschemename,ctablename,cgrupa,oBrowse)

*************************************************************************
STATIC FUNCTION SORT_BY_NAME(cschemename,ctablename,cgrupa,oBrowse)
*************************************************************************
LOCAL radno:=SELECT() // SELECT "ART"
CLOSE ART
cgrupa := padr(cgrupa,4)
IF EMPTY(xGRUPA)
   cSQL := "SELECT * FROM "+cschemename+"."+ctablename+" ORDER BY robn_ ;"
ELSE
   cSQL := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE dobs_='"+cgrupa+"' ORDER BY
robn_ ;"
ENDIF
oStmt   := DacSqlStatement():fromChar(cSQL)
cAlias  := oStmt:build():query(,"ART")
ART:=cAlias
SELECT "ART" // this is mandatory  !
GO TOP
oBrowse:Refreshall()
RETURN NIL
*************************************************************************
STATIC FUNCTION SORT_BY_CODE(cschemename,ctablename,cgrupa,oBrowse)
*************************************************************************
LOCAL radno:=SELECT()  // SELECT "ART"
CLOSE ART
cgrupa := padr(cgrupa,4)
IF EMPTY(xGRUPA)
   cSQL := "SELECT * FROM "+cschemename+"."+ctablename+" ORDER BY robs_ ;"
ELSE
   cSQL := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE dobs_='"+cgrupa+"' ORDER BY
robs_ ;"
ENDIF
oStmt   := DacSqlStatement():fromChar(cSQL)
cAlias  := oStmt:build():query(,"ART")
ART:=cAlias
SELECT "ART" // this is mandatory  !
GO TOP
oBrowse:Refreshall()
RETURN NIL
```

```
* FUNCTION RENUMBERATION()
* FUNCTION RECONSTRUCTION()
* FUNCTION RENUMBER__record()

*************************************************************************
FUNCTION RENUMBERATION(ccTablename,oBrowse,nn)
*************************************************************************
* ccTablename = 'tablename' or 'schemename.tablename'
* Može se koristiti za bilo koju upsize tablu koja ima __record kolonu
* Can be used for any upsize table that has a __record column

LOCAL radno:=SELECT()
PRIVATE cTablename := ccTablename

IF nn=1
c__poruka("START RENUMBERATION","START",,"G3")
ENDIF

* --- PASS 1 START
// Napravi se lista brojeva upisanih u __record za svaki red table
// Create a list of numbers written in __record for each row of the table
* ---
cSQL := "SELECT __record FROM "+cTablename+";"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():query(, @xResult)
SELECT QUERY
aRow:={}
DO WHILE .NOT. EOF()
  AADD(aRow,__record)
SKIP
ENDDO
CLOSE QUERY
SELECT(radno) // SELECT "ART"
* ---
// Izvrši se renumeracija svih redova u tabli u koloni __record Public Key
// All rows in the table in the __record Public Key column are renumbered
* ---
cSQL:=""; ni:=0
FOR ni=1 TO len(aRow)

 alt_No__record := var2char(aRow[ni])  // 1,2,3,5,6      // state after DELETE __record = 4
 new_No__record := var2char(-aRow[ni]) // -1,-2,-3,-5,-6 // state after renumberation pass 1

TEXT INTO txt WRAP
UPDATE &cTablename SET __record = &new_No__record WHERE __record = &alt_No__record;
ENDTEXT

cSQL:=cSQL+txt
NEXT ni
* ---
* dc_memoedit(cSQL) // test
```

```
// sve se to upiše u sve redove table
// everything is written in all the rows of table
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
// DBREFRESH()
* --- PASS 1 END



* --- PASS 2 START
// ponovo se učitaju svi ažurirani redovi table sa negativnim __record
// reload all updated table rows with negative __record
cSQL := "SELECT __record FROM "+cTablename+";"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():query(, @xResult)
SELECT QUERY
aRow:={}
DO WHILE .NOT. EOF()
  AADD(aRow,__record)
SKIP
ENDDO
CLOSE QUERY
SELECT(radno) // SELECT "ART"
* ---
* ---
cSQL:=""; ni:=0
FOR ni=1 TO len(aRow)

 alt_No__record := var2char(aRow[ni]) // -1,-2,-3,-5,-6 // state after renumberation pass 1
 new_No__record := var2char(ni)       // 1,2,3,4,5      // state after renumberation pass 2

TEXT INTO txt WRAP
UPDATE &cTablename SET __record = &new_No__record WHERE __record = &alt_No__record;
ENDTEXT

cSQL:=cSQL+txt
NEXT ni
* ---
* dc_memoedit(cSQL) // test
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
// DBREFRESH()
// APPEND BLANK će dodati novi __record = 6 pošto on sada ne postoji i sve je OK
// APPEND BLANK will add a new __record = 6 since it doesn't exist now and everything is OK
* --- PASS 2 END


* "--- alternative for PASS 2
* RENUMBER__record(cTablename) // here
* DBREFRESH()
* oBrowse:Refreshall()
* "--- alternative for PASS 2

*------------------------------------------------------------------------
* --- AFTER DELETING ROW FOLLOWS RESET COLUMN __record -------------- END
*------------------------------------------------------------------------



// rekonstrukcija formirane prazne i/ili napunjene table:
```

```
  cSQL := "VACUUM FULL "+cTablename+";"
  oStmt := DacSqlStatement():fromChar(cSQL)
  oStmt:build():execute()


   SELECT(radno) // SELECT "ART"
   DbRefresh()
   oBrowse:refreshall()
   GO TOP

* ---
IF nn=1
c__poruka("END RENUMBERATION","END",,"G3")
ENDIF
RETURN NIL



*****************************************************************************
FUNCTION RECONSTRUCTION(ccTablename,oBrowse,nn)
*****************************************************************************
* ccTablename = 'tablename' or 'schemename.tablename'
* Može se koristiti za bilo koju upsize tablu koja ima __record kolonu
* Can be used for any upsize table that has a __record column

LOCAL radno:=SELECT()
PRIVATE cTablename := ccTablename

IF nn=1
c__poruka("START RECONSTRUCTION","START",,"G3")
ENDIF


* ---
  * specijalna operacija:
  * APPEND BLANK kad se prekine sa error u tablu doda jedan prazan red
  * koji nema šifru artikla i koji treba obrisati.
  * Biće obrisani svi redovi koji imaju robs_ IS NULL (robs_="     ")
  * special operation:
  * APPEND BLANK, when terminated with an error, adds one blank row to the table
  * which does not have an article code and which should be deleted
  * All rows that have robs_ IS NULL (robs_="     ") will be deleted

  cSQL := "DELETE FROM ONLY "+cTablename+" WHERE robs_ IS NULL;"
  oStmt := DacSqlStatement():fromChar(cSQL)
  oStmt:build():execute()
  SELECT(radno)
  //DBREFRESH()
* ---


* --- PASS 1 START
// Napravi se lista brojeva upisanih u __record za svaki red table
// Create a list of numbers written in __record for each row of the table
* ---
cSQL := "SELECT __record FROM "+cTablename+";"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():query(, @xResult)
SELECT QUERY
aRow:={}
```

```
DO WHILE .NOT. EOF()
  AADD(aRow,__record)
SKIP
ENDDO
CLOSE QUERY
SELECT(radno) // SELECT "ART"
* ---
// Izvrši se renumeracija svih redova u tabli u koloni __record Public Key
// All rows in the table in the __record Public Key column are renumbered
* ---
cSQL:=""; ni:=0
FOR ni=1 TO len(aRow)

 alt_No__record := var2char(aRow[ni])  // -1,-2,-3,-5,-6 state after accident
 new_No__record := var2char(-aRow[ni]-100000)
 // -100001,-200002,-300003,-500005,-600006  state after reconstructions pass 1

TEXT INTO txt WRAP
UPDATE &cTablename SET __record = &new_No__record WHERE __record = &alt_No__record;
ENDTEXT

cSQL:=cSQL+txt
NEXT ni
* ---
* dc_memoedit(cSQL) // test
// sve se to upiše u sve redove table
// everything is written in all the rows of the board
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
//DBREFRESH()
* --- PASS 1 END


* --- PASS 2 START
// ponovo se učitaju svi ažurirani redovi table sa negativnim __record
// reload all updated table rows with negative __record
cSQL := "SELECT __record FROM "+cTablename+";"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():query(, @xResult)
SELECT QUERY
aRow:={}
DO WHILE .NOT. EOF()
  AADD(aRow,__record)
SKIP
ENDDO
CLOSE QUERY
SELECT(radno) // SELECT "ART"
* ---
* ---
cSQL:=""; ni:=0
FOR ni=1 TO len(aRow)


 alt_No__record := var2char(aRow[ni])
 // -100001,-200002,-300003,x,-500005,-600006 state after reconstructions pass 1
 new_No__record := var2char(ni) // 1,2,3,4,5  state after reconstructions pass 2


TEXT INTO txt WRAP
UPDATE &cTablename SET __record = &new_No__record WHERE __record = &alt_No__record;
```

```
ENDTEXT

cSQL:=cSQL+txt
NEXT ni
* ---
* dc_memoedit(cSQL) // test
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
// DBREFRESH()
// APPEND BLANK će dodati novi __record = 6 pošto on sada ne postoji i sve je OK
// APPEND BLANK will add a new __record = 6 since it doesn't exist now and everything is OK
* --- SECTION 2 END

*------------------------------------------------------------------------
* --- AFTER DELETING ROW FOLLOWS RESET COLUMN __record -------------- END
*------------------------------------------------------------------------


// rekonstrukcija formirane prazne i/ili napunjene table:
  cSQL := "VACUUM FULL "+cTablename+";"
  oStmt := DacSqlStatement():fromChar(cSQL)
  oStmt:build():execute()


SELECT(radno) // SELECT "ART"
DbRefresh()
oBrowse:refreshall()
setappfocus(oBrowse)
GO TOP  // ubije se program ???


* ---
IF nn=1
c__poruka("END RECONSTRUCTION","END",,"G3")
ENDIF


RETURN NIL



*************************************************************************
FUNCTION RENUMBER__record(cTablename)
*************************************************************************
* ccTablename = 'tablename' or 'schemename.tablename'
* Može se koristiti za bilo koju upsize tablu koja ima __record kolonu
* Can be used for any upsize table that has a __record column

LOCAL radno:=SELECT()
LOCAL cSQL,oStmt,nRows
PRIVATE xTable := cTablename
*------------------------------------------------------------------- START
* RENUMERACIJA kolone __record brojevima od 1 do n
* RENUMBER of the __record column with numbers from 1 to n
*------------------------------------------------------------------------
* https://stackoverflow.com/questions/41346345/
* how-can-i-update-all-rows-in-postgresql-from-1-to-n-where-n-number-of-rows
* Example:
* update kroba_11
*   set __record = t.rn
```

```
* from (
*   select __record,
*          row_number() over (order by __record) as rn
*   from kroba_11
* ) t
* where t.__record = kroba_11.__record;
* ---

PRIVATE xTableRecord := xTable+".__record"

TEXT INTO cSQL WRAP
update &xTable
  set __record = t.rn
from (
  select __record,
         row_number() over (order by __record) as rn
  from &xTable
) t
where t.__record = &xTablerecord;
ENDTEXT

* dc_memoedit("2"+chr(13)+chr(10)+cSQL) // test

oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
* ---

* ---
// Brojanje redova u tabli
// Counting rows in the table
cSQL:="SELECT COUNT(*) FROM "+xTable+";"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():query()
SELECT QUERY
nRows:=count
CLOSE QUERY
c_poruka("NUMBER OF ROWS RENUMBERED "+var2char(nRows))
*------------------------------------------------------------------------
* RENUMERACIJA kolone __record brojevima od 1 do n
* RENUMBER of the __record column with numbers from 1 to n
*------------------------------------------------------------------------END

SELECT(radno) // SELECT "ART"
DbRefresh()

RETURN NIL
```

__ARTICLES_DESCRIPTION.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES_DESCRIPTION.PRG                          SQL          //
//                                                                    //
//   01-11-2023                                                       //
//                                                                    //
//   www.cobasystems.com --- COBA Systems --- Slobodan Stanojević Coba //
//   Open Source Project BAST Business Account Software Technology    //
//   www.Alaska-Software.com --- Alaska Xbase++ version 2.0.1503       //
//   www.Donnay-software.com --- eXpress++     version 2.0.268         //
//   Sergej Spirin --- FastReport for Xbase++  version 27.03.2015      //
//                                                                    //
//   Database Server PostgreSQL                version 9.4.4.         //
//                                                                    //
//                                                                    //
////////////////////////////////////////////////////////////////////////

/*

-------------------------------
  PACKAGE:
  __ARTICLES__MAIN.PRG
  __ARTICLES_CODEBOOK.PRG
  __ARTICLES_SELECTION.PRG
  __ARTICLES_DESCRIPTION.PRG
  __ARTICLES_REGISTERCARD.PRG
-------------------------------

  EDITOR TEKST STRINGA - MEMO POLJA  - U DBF/DBT FAJLU
  EDITOR TEKST STRINGA - TXT  KOLONE - U POSTGRESQL TABLI

  TEXT STRING EDITOR - MEMO FIELDS - IN DBF/DBT FILE
  TEXT STRING EDITOR - TXT COLUMNS - IN POSTGRESQL TABLE

-------------------------------
 FUNCTION ARTICLES_DESCRIPTION(oBrowse)
 FUNCTION _Editor_()
 STATIC FUNCTION mehlp()
 STATIC FUNCTION xxx(oDlg,x,y)  // test window position

*/

#include "Xbp.ch"
#include "AppEvent.ch"
#include "Common.ch"

#include "xbtsys.ch"
#include "dcdialog.ch"

#include "pgdbe.ch"

#include "std.ch"
#include "sql.ch"
```

```
#include "dac.ch"
#pragma library("adac20b.lib")

* koristi se za testiranje programa
*  #include "Appbrow.ch"
*  MEMVAR appObject




*****************************************************************************

FUNCTION ARTICLES_DESCRIPTION(oBrowse)  // Obrada text polja OPIS_ iz XbpMLE editora

*****************************************************************************
LOCAL radno := SELECT()  // SELECT "ART"

rec:=recno() // this works

PRIVATE ;
      sifra := ROBS_,; // string from table
      naziv := ROBN_,; // string from table
      cTxt  := ""   ,; // string
      txt   := gde_exe()+"\t.txt" // txt file

      // POSTUPAK:
      // UČITAJ IZ TABLE TEKST OPIS_ U STRING cTxt
      // OBRISI TXT FAJL txt AKO POSTOJI
      // ZAPISI STRING cTxt U TXT FAJL txt
      // IZ EDITORA MLE UČITAJ TEKST cTxt IZ TXT FAJLA txt
      // IZ EDITORA MLE EDITUJ TEKST cTxt
      // IZ EDITORA MLE ZAPIŠI TEKST cTxt U TXT FAJL txt
      // UČITAJ IZ TXT FAJLA txt STRING cTxt
      // ZAPIŠI STRING cTxt U TABLU U OPIS_

       // PROCEDURE:
       // LOAD FROM TABLE TEXT OPIS_ INTO STRING cTxt
       // DELETE TXT FILE txt IF EXISTS
       // WRITE STRING cTxt TO TXT FILE txt
       // FROM MLE EDITOR LOAD TEXT cTxt FROM TXT FILE txt
       // FROM EDITOR MLE EDIT TEXT cTxt
       // FROM THE MLE EDITOR WRITE THE TEXT cTxt TO THE TXT FILE txt
       // LOAD FROM TXT FILE txt STRING cTxt
       // WRITE STRING cTxt IN TABLE IN OPIS_


      cTxt := OPIS_    // string from table - string from memo fields
      DELETE FILE (txt)
      MemoWrit(txt,cTxt)



 ****************************************************************
  IZMENA := ;
  _Editor_(txt,67,8,"centar","ARTICLE DESCRIPTION:",sifra+"  "+naziv)
 ****************************************************************
```

```
 IF IZMENA==.T. // samo ako je bilo izmena u tekstu
                // only if there were changes in the text


    // UCITAJ TEKST IZ TXT FAJLA txt U STRING
    // LOAD TEXT FROM TXT FILE txt INTO STRING
     cTxt := ALLTRIM( MemoRead(txt) )


    ************************************************************

     PRIVATE ztable:=cschemename+"."+ctablename
     // ne može: &cschemename.&ctablename // tačka se ne prikazuje u TEXT INTO
     // can't: &cschemename.&ctablename // dot doesn't show in TEXT INTO

     PRIVATE cSQL, cStmt
     TEXT INTO cSQL WRAP
     UPDATE &ztable SET OPIS_ = '&cTxt' WHERE ROBS_ = '&sifra'
     ENDTEXT
     // dc_memoedit(cSQL)

     oStmt := DacSqlStatement():fromChar(cSQL)
     oStmt:build():execute()


     SELECT (radno)// ovo je obavezno | this is mandatory
     DBREFRESH()
     oBrowse:refreshall() // ova komanda osvežava browser sa novim podatkom
                          // this command refreshes the browser with new data
    ************************************************************


    ************************************************************
    * SELECT(radno)
    *
    *  IF DBRLOCK() // TABLE IS USE EXCLUSIVE
    *    REPLACE OPIS_ WITH cTxt  // this doesn't work
    *    * for table is:
    *    * "SELECT * FROM "+cschemename+"."+ctablename+" ORDER BY robn_ ;"
    *    * it is obtained:
    *    * error: * file is opened in read-only mode
    *  DBRUNLOCK()
    *  ENDIF
    *
    * DBREFRESH()
    * oBrowse:refreshall() // ova komanda osvežava browser sa novim podatkom
    ************************************************************


 ENDIF // IF IZMENA==.T. // samo ako je bilo izmena u tekstu
                         // only if there were changes in the text
SELECT(radno)
RETURN(nil)



**********************************************************************
* Example: use CRT X=67,Y=8 not PIXEL X=600, Y=300
* eEditor1( txtfajl,67,8,"centar",f_i,P_NAZIV,;
```

```
*           "8.Helv Bold",GRA_CLR_DARKRED,"8.Helv",GRA_CLR_BLACK)

FUNCTION _Editor_( cFileName,x,y,pozicija,cnaslov1,cnaslov2,;
                   cFontnaslov2,cBojanaslov2,cFontTekst,cBojaTekst )

*****************************************************************************
LOCAL GetList := {}, getoptions, oDlg,;
      oNaziv, size1, sTxt ,oEdit , lOpis := .F.

DEFAULT cFileName TO Gde_Exe()+"\t.txt"    ,;
        x         TO 67                    ,;
        y         TO 8                     ,;
        pozicija     TO "centar"           ,;
        cnaslov1     TO "COBA Systems ®"   ,;
        cnaslov2     TO "BELEŠKA"          ,;
        cFontNaslov2 TO "12.Consolas Bold" ,;
        cBojaNaslov2 TO GRA_CLR_DARKRED    ,;
        cFontTekst   TO "11.Consolas"      ,;
        cBojaTekst   TO GRA_CLR_DARKBLUE


      IF File(cFileName) = .F.
         MemoWrit(cFileName,cnaslov1)
      ENDIF


      aCUR      := {"USER32.DLL",114,XBPWINDOW_POINTERTYPE_POINTER}



                   BMP_DOC := XbpBitmap():new():create()
                   BMP_DOC:load( "BAZNE.DLL", 11021 )
                   BMP_DOC:transparentClr := BMP_DOC:getDefaultBgColor()

                   BMP_OK := XbpBitmap():new():create()
                   BMP_OK:load( "BAZNE.DLL", 11041 )
                   BMP_OK:transparentClr := BMP_OK:getDefaultBgColor()

                   BMP_ESC := XbpBitmap():new():create()
                   BMP_ESC:load("BAZNE.DLL",11042 )
                   BMP_ESC:transparentClr := BMP_ESC:getDefaultBgColor()

                   BMP_HLP := XbpBitmap():new():create()
                   BMP_HLP:load( "BAZNE.DLL", 11043 )
                   BMP_HLP:transparentClr := BMP_HLP:getDefaultBgColor()



   // --------------------- dialog top right
      aRefSize := SetAppWindow():currentSize()
      aRefPos  := SetAppWindow():currentPos()

         aSize := { x * 7.3, y * 33.4 } // prevođenje BAZNIH CRT u PIXELE
         aPos  := {0,0}

      IF pozicija == "goredesno"
         aPos  := GoreDesnoPos( aSize, aRefSize, aRefPos )   // Bazne.dll
      ENDIF
      IF pozicija == "gorelevo"
```

```
         aPos  := GoreLevoPos( aSize, aRefSize, aRefPos )   // Bazne.dll
      ENDIF
      IF pozicija == "doledesno"
         aPos  := DoleDesnoPos( aSize, aRefSize, aRefPos )   // Bazne.dll
      ENDIF
      IF pozicija == "dolelevo"
         aPos  := DoleLevoPos( aSize, aRefSize, aRefPos )   // Bazne.dll
      ENDIF

      IF pozicija == "centar"
         aRefSize := AppDesktop():currentSize()
         aPos  := CenterPos( aSize, aRefSize )
      ENDIF
   // --------------------- dialog top right


         sTxt := MemoRead(cFileName)
         sTxt := HardCR(sTxt)


   @ 0,0  DCSAY cnaslov2  SAYFONT cFontnaslov2 SAYCOLOR cBojanaslov2 SAYSIZE 0


   @ 2,0   DCMULTILINE sTxt ;
           SIZE x,y OBJECT oEdit ;
           FONT cFontTekst ;
           COLOR cBojaTekst, GRA_CLR_WHITE ;
           NOWORDWRAP

   ox := 2+y+1

tbr := 67
btt := tbr/3
dd := 7

@ ox,0 DCTOOLBAR oToolBar SIZE tbr,3 BUTTONSIZE btt,3

   *************
   IZMENA := .F.
   *************
   DCADDBUTTON CAPTION BMP_OK PARENT oToolBar SIZE dd WHEN {||.F.}
   DCADDBUTTON CAPTION "Save-Exit" PARENT oToolBar ;
   ACCELKEY { xbeK_TAB } ;
   TOOLTIP " [TAB] Snimi izmene i izlaz | Save changes and exit " ;
   ACTION {|| IZMENA:=.T. ,;
              MemoWrit( cFileName, sTxt ), tone(100),;
              c_poruka("SNIMLJENO | RECORDED"),;
              DC_GetRefresh(Getlist), SetAppFocus(oEdit),;
              DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList) } ;
   CURSOR aCUR SIZE btt-dd


   DCADDBUTTON CAPTION BMP_HLP PARENT oToolBar SIZE dd WHEN {||.F.}
   DCADDBUTTON CAPTION "Help" PARENT oToolBar ;
   ACCELKEY { xbeK_F1 } ;
   TOOLTIP " [F1] Help " ;
   ACTION {|| mehlp(), DC_GetRefresh(Getlist), SetAppFocus(oEdit) } ;
   CURSOR aCUR SIZE btt-dd

   DCADDBUTTON CAPTION BMP_ESC PARENT oToolBar SIZE dd WHEN {||.F.}
```

```
   DCADDBUTTON CAPTION "Exit" PARENT oToolBar ;
   ACCELKEY { xbeK_ESC } ;
   TOOLTIP " [Esc] Izlaz " ;
   ACTION {|| DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList) }  ;
   CURSOR aCUR SIZE btt-dd

   DCGETOPTIONS WINDOWROW aPos[2] WINDOWCOL aPos[1];
   NOESCAPEKEY NOMAXBUTTON NOMINBUTTON NOTASKLIST FONT "10.Arial Bold"

   DCREAD GUI FIT;
      TITLE cnaslov1 ;
      OPTIONS GetOptions ;
              PARENT @oDlg ;
              EVAL {||SetAppWindow(oDlg)} ;
              MODAL

RETURN IZMENA



*************************************************************************
STATIC FUNCTION mehlp()
*************************************************************************
c__poruka("Detaljniji opis artikla koji se uvek može" + chr(59)+;
          "videti uz artikal, a štampa se na fakturi" + chr(59)+;
          " "                                         + chr(59)+;
          "Detailed description of article that can always"  + chr(59)+;
          "see with article, and it is printed on invoice."  + chr(59)+;
          "","Detaljni Opis artikla | Detailed description of article",,"G1")

RETURN NIL


STATIC FUNCTION xxx(oDlg,x,y)
 as := oDlg:currentsize()
 ap := odlg:currentpos()
  msgbox(var2char(as[1])+"x"+var2char(as[2])+chr(13)+;
  var2char(ap[1])+"x"+var2char(ap[2]),"size-pos: "+alltrim(str(x))+" "+alltrim(str(y)))
RETURN NIL
```

__ARTICLES_REGISTERCARD.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES_REGISTERCARD.PRG                          SQL         //
//                                                                    //
//   01-11-2023                                                       //
//                                                                    //
//   www.cobasystems.com --- COBA Systems --- Slobodan Stanojević Coba //
//   Open Source Project BAST Business Account Software Technology    //
//   www.Alaska-Software.com --- Alaska Xbase++ version 2.0.1503       //
//   www.Donnay-software.com --- eXpress++      version 2.0.268        //
//   Sergej Spirin --- FastReport for Xbase++   version 27.03.2015     //
//                                                                    //
//   Database Server PostgreSQL                 version 9.4.4.        //
//                                                                    //
//                                                                    //
////////////////////////////////////////////////////////////////////////

/*
-------------------------------
  PACKAGE:
  __ARTICLES__MAIN.PRG
  __ARTICLES_CODEBOOK.PRG
  __ARTICLES_SELECTION.PRG
  __ARTICLES_DESCRIPTION.PRG
  __ARTICLES_REGISTERCARD.PRG
-------------------------------

REGISTAR KARTICA ARTIKLA SA SVIM PODACIMA O ARTIKLU
PREGLED I EDITOVANJE UZ OVLAŠĆENJE

ARTICLE CARD REGISTER WITH ALL INFORMATION ABOUT ARTICEL
VIEWING AND EDITING WITH AUTHORIZATION

-------------------------------
FUNCTION ARTICLES_REGISTERCARD()
 FUNCTION artikal_CFG_upisi1()
 FUNCTION artikal_CFG_citaj1()
 STATIC FUNCTION XGRUPA()
 STATIC FUNCTION Artikli_Maticni_Podaci_Pisi(oBrowse)
 STATIC FUNCTION Artikli_Maticni_Podaci_Public()
 STATIC FUNCTION Artikli_Maticni_Podaci_Release()
 STATIC FUNCTION Artikli_Maticni_Podaci_Citaj()
 STATIC FUNCTION help11()


*/



#include "Xbtsys.ch"   // XbttoolsIII
#include "Appevent.ch"
```

```
#include "Common.ch"
#include "Xbp.ch"
#include "dll.ch"
#include "dcdialog.ch"




 // UPIS SIFRE ARTIKLA U ARTIKAL.CFG
 // WRITE ARTICLE CODE IN ARTIKAL.CFG
   ************************************************************
   FUNCTION artikal_CFG_upisi1( cfg_asifra, cfg_abarcod )
   ************************************************************
      LOCAL cUser := alltrim(User_Name())
      LOCAL ARTIKALLOK := Gde_Exe()+"\"+cUser, ;
            ARTIKALCFG := Gde_Exe()+"\"+cUser+"\ARTIKAL.CFG",;
            rezultat

      DEFAULT cfg_asifra  TO space(5),;  // mogu da budu brojevi, slova, znaci, prazno
              cfg_abarcod TO space(25)   // mogu da budu brojevi, slova, znaci, prazno


      IF FILE(ARTIKALLOK,"D")=.F.
        rezultat = DirMake(ARTIKALLOK)
        IF rezultat <> NO_DISK_ERR
         MsgBox("Nemoguc pristup području:"+chr(13)+;
                 ARTIKALLOK,"STOP")
         QUIT_QUIT() // bazne.dll
        ENDIF
      ENDIF // IF FILE(ARTIKALLOK,"D")=.F.

      MemoWrit( ARTIKALCFG, ALLTRIM(cfg_asifra)+"/"+ALLTRIM(cfg_abarcod) )

     RETURN(nil)


  // CITANJE SIFRE ARTIKLA IZ ARTIKAL.CFG
  // READ ARTICLE CODE FROM ARTIKAL.CFG
   ************************************************************
   FUNCTION artikal_CFG_citaj1()
   ************************************************************

   LOCAL cUser := alltrim(User_Name())  // bazne.dll
   LOCAL ARTIKALCFG := Gde_Exe()+"\"+cUser+"\ARTIKAL.CFG",;
         string := ""

   PUBLIC cfg_asifra := space(5)   ,; // mogu da budu brojevi, slova, znaci, prazno
          cfg_abarcod := space(25)    // // can be numbers, letters, signs, blank

       IF FILE( ARTIKALCFG ) = .F.
            artikal_CFG_upisi1()  // otvori prazan fajl ARTIKAL.CFG
       ENDIF                      // open empty file ARTIKAL.CFG

       string := ALLTRIM(MemoRead( ARTIKALCFG ))

       cfg_asifra  := ALLTRIM(TOKEN(string,"/",1))
       cfg_abarcod := ALLTRIM(TOKEN(string,"/",2))

   RETURN(cfg_asifra)
```

```
*********************************************************************
FUNCTION ARTICLES_REGISTERCARD(lEdit,GetList1,oBrowse,GRUPA)
*********************************************************************
* lEdit:=.T. editovanje ON  EDITPROTECT OFF :=.F. (!lEdit)
* lEdit:=.F. editovanje OFF EDITPROTECT ON  :=.T. (!lEdit)

************************
LOCAL radno := SELECT() // SELECT "ART"
************************

 LOCAL GetList := {}, GetOptions, boja, xx:=0, zz:=1+0.2
 LOCAL aCUR := {"user32.dll",114}
 LOCAL oCUR := {"user32.dll",112}

 DEFAULT lEdit TO .F. // NO EDIT NO UPDATE

                    BMP_DOC := XbpBitmap():new():create()
                    BMP_DOC:load( "BAZNE.DLL", 11021 )
                    BMP_DOC:transparentClr := BMP_DOC:getDefaultBgColor()

                    BMP_OK := XbpBitmap():new():create()
                    BMP_OK:load( "BAZNE.DLL", 11041 )
                    BMP_OK:transparentClr := BMP_OK:getDefaultBgColor()

                    BMP_ESC := XbpBitmap():new():create()
                    BMP_ESC:load("BAZNE.DLL",11042 )
                    BMP_ESC:transparentClr := BMP_ESC:getDefaultBgColor()

                    BMP_HLP := XbpBitmap():new():create()
                    BMP_HLP:load( "BAZNE.DLL", 11043 )
                    BMP_HLP:transparentClr := BMP_HLP:getDefaultBgColor()

                    BMP_EDIT := XbpBitmap():new():create()
                    BMP_EDIT:load( "BAZNE.DLL", 3500 )
                    BMP_EDIT:transparentClr := BMP_EDIT:getDefaultBgColor()


 ROBS := ROBS_

 Artikli_Maticni_Podaci_Public()
 Artikli_Maticni_Podaci_Citaj(ROBS)

 boja := { GRA_CLR_BLACK, GRA_CLR_WHITE }
 SET DATE FORMAT TO 'dd.mm.yyyy'



@  0,0 DCSTATIC TYPE XBPSTATIC_TYPE_BITMAP PIXEL SIZE 33,33 CAPTION BMP_EDIT // BAZNE.DLL


xx:=2

xx:=xx+zz
   @ xx, 1 DCSAY 'Article Code | Šifra artikla.................' GET ROBS EDITPROTECT {||.T.} ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Interna Šifra artikla | "+;
```

```
   "Internal Article code "
xx:=xx+zz
   @ xx, 1 DCSAY 'Article Type | Tip artikla..................' GET GRUP ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Grupa Hrana = slovo H, Grupa Piće = slovo P | "+;
   "Group Food = letter H, Group Beverage = letter P"
xx:=xx+zz
   @ xx, 1 DCSAY 'Article Name | Naziv artikla.................' GET ROBN ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Skraćeni naziv artikla | "+;
   "Short name of the Article "

xx:=xx+zz
   @ xx, 1 DCSAY 'Long Article name | Dugački Naziv artikla:' ;
   SAYCOLOR {||boja} CURSOR oCUR MESSAGE ;
   "Pun neskraćeni naziv artikla | "+;
   "Full unabbreviated Article name "


xx:=xx+zz
   @ xx, 1 DCGET NAME PICTURE "@S52" // +replicate("x",200)


xx:=xx+zz+1
   @ xx, 1 DCSAY 'BarCode | Barkod............................' GET ROBK;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Interni ili stvarni barkod artikla | "+;
   "Internal or actual article barcode"
xx:=xx+zz
   @ xx, 1 DCSAY 'Catalog Number | Kataloški broj.............' GET DOBK;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Kataloški broj artikla ili drugi podatak | "+;
   "Article catalog number or other information"
xx:=xx+zz
   @ xx, 1 DCSAY 'Reference for Link | Referenca na vezu.......' GET VEZA;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE;
   "Oznaka reference za vezu sa artiklom | "+;
   "Reference mark for link to the article"
xx:=xx+zz
   @ xx, 1 DCSAY 'Group Article 1-16 | Grupa artikala 1-16.....' GET DOBS;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE;
   "Razvrstavanje artikla u 16 grupa od 1 do 16 | "+;
   "Classification of the article into 16 groups from 1 to 16"
xx:=xx+zz
   @ xx, 1 DCSAY 'Article Measurment Unit | Jedinica mere......' GET MERA;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Oznaka za jedinicu mere artikla | "+;
   "Designation for the unit of measure of the article"
xx:=xx+zz
   @ xx, 1 DCSAY 'Tarifa (internal) | Tarifa (interno).........' GET TARI;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Interna oznaka (ugrađuje se u broj konta) | "+;
   "Internal code (built into the account number)"
xx:=xx+zz
   @ xx,1 DCSAY 'VAT rate in % | Stopa poreza u %.............' GET PPUP;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "PDV stopa artikla u %  |  "+;
   "VAT rate of article in %"
xx:=xx+zz
xx:=xx+zz
```

```
   @ xx,1 DCSAY 'Sort article | Vrsta artikla | R,P,M,U,K......' GET STATx;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "R-roba, P-proizvod, M-materijal, U-usluga, K-komision | "+;
   "R-goods, P-product, M-material, U-service, K-commission"
xx:=xx+zz
   @ xx,1 DCSAY 'Sales Price incl.VAT | Cena maloprodajna......' GET CENA_MPR GETSIZE 20 ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Prodajna cena u kojoj se sadrži i PDV | "+;
   "Sales price including VAT"
xx:=xx+zz
   @ xx,1 DCSAY 'Foreign currency Price | Cena devizna.........' GET CENA_DEV GETSIZE 20 ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Devizna Prodajna cena u kojoj se sadrži i PDV | "+;
   "Foreign currency Sales price including VAT"
xx:=xx+zz
   @ xx,1 DCSAY 'Quantity of article in stock | Zalihe.........' GET ZALI     GETSIZE 20 ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Trenutna količina artikla na zalihama | "+;
   "Current quantity of article in stock"

//-------------------------------------------------------
xx:=xx+zz
   @ xx,1 DCSAY 'Rok trajanja | expiration date.........' GET DAT0     GETSIZE 20 ;
   SAYCOLOR {||boja} SAYCURSOR oCUR MESSAGE ;
   "Rok trajanja ili drugi datum | "+;
   "expiration date or other date"
//-------------------------------------------------------


xx:=xx+zz

   of4 := Font_codepage(10,"Werdana",238,.t.)
   @ xx,1 DCMESSAGEBOX OBJECT oMsgBox SIZE 60.2,2 ;
          TEXTOBJECT MESSAGE OPTIONS XBPSTATIC_TEXT_WORDBREAK;
          COLOR GRA_CLR_DARKRED EVAL {|o| o:setfont(of4) }

xx:=xx+zz+2


//-----------------------------------------------------------------------

tbr := 60
btt := tbr/4
dd := 7

@ xx,1 DCTOOLBAR oToolBar SIZE tbr,3 BUTTONSIZE btt,3 FONT of3

DCADDBUTTON CAPTION BMP_OK PARENT oToolBar SIZE dd WHEN {||.F.}
DCADDBUTTON CAPTION "OK" PARENT oToolBar ;
   ACTION {|| sifra_artikla := ROBS_ ,;
   DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList)} ;
   TOOLTIP " Potvrdi izmenu - Confirm the change " ;
   CURSOR aCUR SIZE btt-dd

DCADDBUTTON CAPTION BMP_EDIT PARENT oToolBar SIZE dd WHEN {||.F.}
DCADDBUTTON CAPTION "Edit" PARENT oToolBar ;
   ACTION {|| lEdit:=.T. , c_poruka("EDIT ON"),;
   DC_GetRefresh(GetList)} ;
   TOOLTIP " Edituj podatke - Edit data " ;
   CURSOR aCUR SIZE btt-dd
```

```
   // ; WHEN {|| XGRUPA(GRUPA) } // button ON/OFF

DCADDBUTTON CAPTION BMP_HLP PARENT oToolBar SIZE dd WHEN {||.F.}
DCADDBUTTON CAPTION "Help" PARENT oToolBar ACCELKEY xbeK_F1 ;
   ACTION {|| help11(),;
   SetAppFocus(oBrowse)} ;
   TOOLTIP " [F1] Pomoć-Help " ;
   CURSOR aCUR SIZE btt-dd

DCADDBUTTON CAPTION BMP_ESC PARENT oToolBar SIZE dd WHEN {||.F.}
DCADDBUTTON CAPTION "Exit" PARENT oToolBar ACCELKEY xbeK_ESC;
   OBJECT oExit;
   ACTION {|| sifra_artikla := "",;
   DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList)} ;
   TOOLTIP " [Esc] izlaz-exit " ;
   CURSOR aCUR  SIZE btt-dd

//----------------------------------------------------------------------

   DCGETOPTIONS ICON 1;
      NOESCAPEKEY NOMINBUTTON NOMAXBUTTON;
      FONT "10.Archivo Narrow" GETFONT "11.Consolas Bold";
      EDITPROTECT {|| !lEdit } ;  // dcget blocade
      SAYWIDTH 200 ;
      COLORGETS { { GRA_CLR_DEFAULT, GRA_CLR_WHITE },{ GRA_CLR_DEFAULT,;
       GraMakeRGBColor({235,232,247}) } }

   DCREAD GUI ;
      OPTIONS GetOptions ;
      FIT;
      TITLE "ARTICLE REGISTER CARD" ;
      PARENT @oDlg ;
      SETFOCUS @oexit ;
      MODAL ;
      EVAL {|o|SetAppWindow(o)}




      IF lEdit==.T. // EDIT AND UPDATE

         * SELECT "ART"
         SELECT(radno)

         Artikli_Maticni_Podaci_Pisi(ROBS,oBrowse)    // WRITE DATA

         * SELECT "ART"
         SELECT(radno)
         dc_getrefresh(GetList)

         GO TOP
         LOCATE FOR ROBS_ == ROBS
         oBrowse:Refreshall()

         Artikli_Maticni_Podaci_Release()
         CLEAR TYPEAHEAD  // vraca fokus na prethodni objekat
                          // returns focus to the previous object
      ENDIF
```

```
      Artikli_Maticni_Podaci_Release()
      CLEAR TYPEAHEAD  // vraca fokus na prethodni objekat
                       // returns focus to the previous object
RETURN nil

************************************************************
STATIC FUNCTION XGRUPA(GRUPA)
************************************************************
IF EMPTY(GRUPA)
   RETURN .T.
ENDIF
// C__GRESKA("NO EDIT","STOP",,"G3")
RETURN .F.




************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Pisi(ROBS,oBrowse)
************************************************************
LOCAL radno := SELECT()

PRIVATE ztable := _she_ + "." + ctablename
// ne može: &cschemename.&ctablename - tačka se ne prikazuje u TEXT INTO
// can't: &cschemename.&ctablename - dot doesn't show in TEXT INTO

PRIVATE cSQL, cStmt

PRIVATE xROBS := ROBS
PRIVATE xDAT0 := DAT0  // Xbase++ DBF date format 20231122 (var2char)
IF EMPTY(xDAT0)
   xDAT0:=DATE()       // Xbase++ DBF date format 20231122 (var2char)
ENDIF


* ---
* SQL command UPDATE
* "UPDATE kroba_11 SET dat0_ = '2023-11-22' WHERE robs_='12345';"
* ---
*   xDAT0=20231122 // Xbase++ DBF date format
* potrebno je za upis u postgreSQL tablu
* it is necessary to write in the sql table
*   xDAT0="2023-11-22" // string
* ---

 xDAT0 := DTOS(xDAT0) // ---> string: "20231122"
 //STOP(2,xDAT0,1)      // ---> "20231122"
 xDAT0:=SUBSTR(xDAT0,1,4)+"-"+SUBSTR(xDAT0,5,2)+"-"+SUBSTR(xDAT0,7,2)
 //STOP(3,xDAT0,1)      // ---> "2023-11-22"

* ---
* numerike prevedi u stringove zbog TEXT INTO cSQL WRAP
* numeric translate in strings bye TEXT INTO cSQL WRAP

 PPUP     := var2char(PPUP_)       // numeric
 PPAP     := var2char(PPAP_)       // numeric
 OSNO_AKC := var2char(OSNO_AKC_)   // numeric
 CENA_PRO := var2char(CENA_PRO_)   // numeric
```

```
 CENA_DEV := var2char(CENA_DEV_)   // numeric
 ZALI     := var2char(ZALI_)       // numeric
 ROK      := var2char(ROK_)        // integer
 CENA_FAK := var2char(CENA_FAK_)   // numeric
 CENA_NAB := var2char(CENA_NAB_)   // numeric
 CENA_VPR := var2char(CENA_VPR_)   // numeric
 CENA_MPR := var2char(CENA_MPR_)   // numeric
* ---

TEXT INTO cSQL WRAP
UPDATE &ztable
SET
DAT0_ = '&xDAT0',
GRUP_ = '&GRUP',
ROBN_ = '&ROBN',
ROBK_ = '&ROBK',
DOBK_ = '&DOBK',
VEZA_ = '&VEZA',
DOBS_ = '&DOBS',
MERA_ = '&MERA',
TARI_ = '&TARI',
PPUP_ =  &PPUP,
PPAP_ =  &PPAP,
OSNO_AKC_ = &OSNO_AKC,
ROK_  = &ROK,
CENA_FAK_ = &CENA_FAK,
CENA_NAB_ = &CENA_NAB,
CENA_VPR_ = &CENA_VPR,
CENA_MPR_ = &CENA_MPR,
CENA_PRO_ = &CENA_PRO,
CENA_DEV_ = &CENA_DEV,
ZALI_ = &ZALI,
STAT_ = '&STATx',
FLAG_ = '&FLAG',
ZNAK_ = '&ZNAK',
NAME_ = '&NAME',
OPIS_ = '&OPIS'
WHERE
ROBS_ = '&xROBS'
ENDTEXT
// dc_memoedit(cSQL,1) ; __pg__AppQuit() // test

     oStmt := DacSqlStatement():fromChar(cSQL)
     oStmt:build():execute()
     ********************
     * SELECT "ART"
     SELECT(radno)
     ********************
     SetAppFocus(oBrowse)
     DBREFRESH()
     oBrowse:refreshall()
     dc_getrefresh(GetList)

RETURN NIL



************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Public()
```

```
************************************************************
 PUBLIC ;
 ROBS ,;
 DAT0 ,;
 GRUP ,;
 ROBN ,;
 ROBK ,;
 DOBK ,;
 VEZA ,;
 DOBS ,;
 MERA ,;
 TARI ,;
 PPUP ,;
 PPAP ,;
 OSNO_AKC ,;
 CENA_PRO ,;
 CENA_DEV ,;
 ZALI ,;
 ROK  ,;
 CENA_FAK ,;
 CENA_NAB ,;
 CENA_VPR ,;
 CENA_MPR ,;
 STATx    ,;
 FLAG    ,;
 ZNAK    ,;
 NAME    ,;
 OPIS
RETURN NIL
************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Release()
************************************************************
 RELEASE ;
 ROBS ,;
 DAT0 ,;
 GRUP ,;
 ROBN ,;
 ROBK ,;
 DOBK ,;
 VEZA ,;
 DOBS ,;
 MERA ,;
 TARI ,;
 PPUP ,;
 PPAP ,;
 OSNO_AKC ,;
 ROK  ,;
 CENA_FAK ,;
 CENA_NAB ,;
 CENA_VPR ,;
 CENA_MPR ,;
 CENA_PRO ,;
 CENA_DEV ,;
 ZALI     ,;
 STATx    ,;
 FLAG    ,;
 ZNAK    ,;
 NAME    ,;
 OPIS
RETURN NIL
```

```
*************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Citaj(ROBS)
*************************************************************

 ROBS := ROBS_ // string
 DAT0 := DAT0_ // date
 GRUP := GRUP_ // string
 ROBN := ROBN_ // string
 ROBK := ROBK_ // string
 DOBK := DOBK_ // string
 VEZA := VEZA_ // string
 DOBS := DOBS_ // string
 MERA := MERA_ // string
 TARI := TARI_ // string

 PPUP     := PPUP_       // numeric
 PPAP     := PPAP_       // numeric
 OSNO_AKC := OSNO_AKC_   // numeric
 CENA_PRO := CENA_PRO_   // numeric
 CENA_DEV := CENA_DEV_   // numeric
 ZALI     := ZALI_       // numeric
 ROK      := ROK_        // integer
 CENA_FAK := CENA_FAK_   // numeric
 CENA_NAB := CENA_NAB_   // numeric
 CENA_VPR := CENA_VPR_   // numeric
 CENA_MPR := CENA_MPR_   // numeric

 STATx := STAT_ // string
 FLAG  := FLAG_ // string
 ZNAK  := ZNAK_ // string
 NAME  := NAME_ // string
 OPIS  := OPIS_ // string

RETURN NIL


*************************************************************************
STATIC FUNCTION help11()
*************************************************************************
LOCAL txt, cr := chr(59)
txt := ;
"Ovo je registar kartica artikla (roba, materijal, proizvod, usluga)."    +cr+;
"Artikal se nalazi u spisku artikala prodajnog objekta ili restorana."    +cr+;
"Kartica mora postojati sa ispravnim i potpunim podacima, da bi se"       +cr+;
"podaci o artiklu mogli poslati u druga dokumenta u ovoj aplikaciji."     +cr+;
"Podaci u kartici mogu se menjati i korigovati od strane korisnika. "     +cr+;
" "                                                                       +cr+;
"This is an article card register (goods, material, product, service)."   +cr+;
"The article is in the articles list of sales facility or restaurant."    +cr+;
"The card must exist with correct and complete data, in order to"         +cr+;
"article data could be sent to other documents in this application."      +cr+;
"Data in the card can be changed and corrected by the user. "             +cr+;
""

c__procitaj(txt,"REGISTAR KARTICA ARTIKLA",,"G1")

RETURN(nil)
```

__ARTICLES_SELECTION.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES_SELECTION.PRG                                SQL      //
//                                                                    //
//   01-11-2023                                                       //
//                                                                    //
//   www.cobasystems.com --- COBA Systems --- Slobodan Stanojević Coba //
//   Open Source Project BAST Business Account Software Technology    //
//   www.Alaska-Software.com --- Alaska Xbase++ version 2.0.1503       //
//   www.Donnay-software.com --- eXpress++     version 2.0.268         //
//   Sergej Spirin --- FastReport for Xbase++  version 27.03.2015      //
//                                                                    //
//   Database Server PostgreSQL                version 9.4.4.         //
//                                                                    //
//                                                                    //
////////////////////////////////////////////////////////////////////////


/*

-------------------------------
  PACKAGE:
  __ARTICLES__MAIN.PRG
  __ARTICLES_CODEBOOK.PRG
  __ARTICLES_SELECTION.PRG
  __ARTICLES_DESCRIPTION.PRG
  __ARTICLES_REGISTERCARD.PRG
-------------------------------

IZDVAJANJE-SELEKCIJA ARTIKALA IZ SPISKA ARTIKALA NA POSEBNU FILTER LISTU
SISTEM SA OTVARANJEM OTVORENOG DBF SPISKA ARTIKLA U DRUGOJ INSTANCI
I SA FILTRIRANJEM SPISKA U DRUGOJ INSTANCI DA SE DOBIJE FILTER LISTA
SVE SE DESAVA DOK JE OTVOREN I AKTIVAN GLAVNI SPISAK ARTIKALA OBJEKTA

SELECTION-SELECTION OF ARTICLES FROM LIST OF ITEMS TO A SPECIAL FILTER LIST
SYSTEM WITH OPENING OF THE OPEN DBF LIST OF ARTICLES IN THE SECOND INSTANCE
AND WITH FILTERING THE LIST IN THE SECOND INSTANCE TO GET THE FILTER LIST
EVERYTHING HAPPENS WHILE THE OBJECT'S MAIN ITEM LIST IS OPEN AND ACTIVE

-------------------------------
FUNCTION ARTICLES_SELECTION( selekcija, vrsta, oDlgMain, xxsdbf, grupa )
STATIC FUNCTION si(oDlg)
STATIC FUNCTION selekcija_help()

*/


#include "Appevent.ch"
#include "Xbp.ch"
#include "common.ch"
#include "xbtsys.ch"
#include "dcdialog.ch"
```

```
*********************************************************************************
FUNCTION ARTICLES_SELECTION( selekcija, vrsta, oDlgMain, xxsdbf, grupa )
*********************************************************************************
* selekcija = string: "BEER"
* vrsta     = string: "ROBN_", "DOBS_", "VEZA_"
* oDlgMain  = object: setappwindow()
* xxsdbf    = string: 'tablename' or 'schemename.tablename'
*                     tabla iz koje se uzimaju podaci
*                     table from which the data is taken
* grupa     = string: group of articles "1","2","3"... "99"

LOCAL radno := SELECT()
LOCAL GetList := {}, oBrowse, oBrowBox , oToolBar, bTitle
LOCAL GetOptions , oDlg

LOCAL rek := RecNo(), xsvega  // aktuelni slog spiska artikala

LOCAL BMP_DOC,BMP_OK,BMP_ESC,BMP_HLP
LOCAL aCUR := {"user32.dll",114}
LOCAL oCUR := {"user32.dll",112}

DEFAULT selekcija TO "A", ;
        vrsta     TO "PARTOFNAME",;
        oDlgMain  TO SetAppWindow(),;
        xxsdbf    TO "",;
        grupa     TO ""

// stop(selekcija,vrsta,oDlgMain,xxsdbf,grupa,1)



        BMP_DOC := XbpBitmap():new():create()
        BMP_DOC:load( "BAZNE.DLL", 11021 )
        BMP_DOC:transparentClr := BMP_DOC:getDefaultBgColor()

        BMP_OK := XbpBitmap():new():create()
        BMP_OK:load( "BAZNE.DLL", 11041 )
        BMP_OK:transparentClr := BMP_OK:getDefaultBgColor()

        BMP_ESC := XbpBitmap():new():create()
        BMP_ESC:load("BAZNE.DLL",11042 )
        BMP_ESC:transparentClr := BMP_ESC:getDefaultBgColor()

        BMP_HLP := XbpBitmap():new():create()
        BMP_HLP:load( "BAZNE.DLL", 11043 )
        BMP_HLP:transparentClr := BMP_HLP:getDefaultBgColor()



PRIVATE title_selekcije := "ARTICLE"
IF vrsta = "ROBN_"
   title_selekcije := "ARTICLE NAME"
ENDIF
IF vrsta = "DOBS_"
   title_selekcije := "SUPPLIER CODE"
ENDIF
IF vrsta = "VEZA_"
   title_selekcije := "ARTICLE LINK"
ENDIF
```

```
PRIVATE opis_selekcije := 'contain word = "'+selekcija+'"'

PRIVATE cgrupa := grupa      // cgrupa NO LOCAL
PRIVATE txt    := selekcija  // ctxt NO LOCAL

*"----------------------------- start
*"table or schemename.tablename
*"-----------------------------
// varijabla ctable mora biti PRIVATE zbog komande TEXT INTO cSQL
// u kojoj se koristi kao &ctable
// the ctable variable must be PRIVATE because of the TEXT INTO cSQL
// command in which it is used as &ctable
PRIVATE ctable := xxsdbf  // tabla iz koje se uzimaju podaci
*                         // the table from which the data is taken
*       ctable := "public.kroba_11"
*       cTABLE := "kroba_11"
* or
*
PRIVATE ctable := cschemename+"."+ctablename
// command TEXT INTO cSQL
// ne sme da sadrži - must not contain:
//     &cschemename.&ctablename
// tačka se ne štampa u TEXT INTO - dot is not printed in TEXT INTO
*"-----------------------------
*"table or schemename.tablename
*"----------------------------- end

PRIVATE xAlias, cSQL, cStmt, LISTA




* -----------------
IF vrsta = "ROBN_"  // selekcija je string zadat od strane usera koji se traži u polju robn_ u
tabli
* -----------------

  // daje sve nazive koji počinju sa reči: selekcija
  *  cSQL  := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE robn_ LIKE '"+selekcija+"%'
ORDER BY robn_ ;"

  // https://www.postgresql.org/docs/9.4/functions-string.html
  // pronaći funkciju koja daje sve nazive koji sadrže reč: selekcija
  // strpos('COBA','OB')=2
  *  cSQL  := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE strpos(robn_,'KOBAS')>0 ;"


    IF cgrupa == ""
       TEXT INTO cSQL WRAP
       SELECT * FROM &ctable WHERE strpos( upper(robn_) , upper('&txt') ) > 0 ;
       ENDTEXT
    ELSE
       TEXT INTO cSQL WRAP
       SELECT * FROM &ctable WHERE strpos( upper(robn_) , upper('&txt') ) > 0 AND dobs_ =
'&cgrupa';
       ENDTEXT
    ENDIF

    oStmt := DacSqlStatement():fromChar(cSQL)
    xAlias  := oStmt:build():query(,"LISTA")
```

```
    LISTA:=xAlias
    SELECT "LISTA"
    xsvega:=reccount() // radi ispravno


* -----------------
ENDIF
* -----------------

* -----------------
IF vrsta = "VEZA_"  // selekcija je string preuzet iz polja veza_ u tabli koji se traži u polju
veza_ u tabli
* -----------------

    cSQL  := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE veza_= '"+selekcija+"' ORDER
BY robn_ ;"
    oStmt := DacSqlStatement():fromChar(cSQL)
    xAlias  := oStmt:build():query(,"LISTA")

    LISTA:=xAlias
    SELECT "LISTA"
    xsvega:=reccount() // radi ispravno
* -----------------
ENDIF
* -----------------

* -----------------
IF vrsta = "DOBS_"  // selekcija je string preuzet iz polja dobs_ u tabli  koji se traži u
polju dobs_ u tabli
* -----------------

    cSQL  := "SELECT * FROM "+cschemename+"."+ctablename+" WHERE dobs_='"+selekcija+"' ORDER BY
robn_ ;"
    oStmt := DacSqlStatement():fromChar(cSQL)
    xAlias  := oStmt:build():query(,"LISTA")

    LISTA:=xAlias
    SELECT "LISTA"
    xsvega:=reccount() // radi ispravno
* -----------------
ENDIF
* -----------------


    ****************************
     PRIVATE sifra_artikla := ""
     // ovde se smešta sifra izabranog artikla
     // here is the code of the selected article
    ****************************


@  0,0 DCSTATIC TYPE XBPSTATIC_TYPE_BITMAP PIXEL SIZE 22,22 CAPTION BMP_DOC PARENT BrowBox
@  0,6 DCSAY title_selekcije SAYSIZE 0 PARENT BrowBox
@  1,6 DCSAY opis_selekcije+" = "+var2char(xsvega) SAYSIZE 0 PARENT BrowBox


@  2,0             DCSTATIC TYPE XBPSTATIC_TYPE_RECESSEDBOX SIZE 45,11.2 OBJECT oBrowBox
@  0+0.2,0+0.2     DCBROWSE oBrowse PARENT oBrowBox ALIAS "LISTA" ;
                   SIZE 45-0.4,11.2-0.4 ;
                   CURSORMODE XBPBRW_CURSOR_ROW ;
```

```
                  ITEMSELECTED {|| sifra_artikla := ROBS_ ,;
                  DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList) }; // FONT "6.Helv"


   DCBROWSECOL FIELD LISTA->ROBN_     WIDTH 20 HEADER "ARTICLE NAME" PARENT oBrowse
   DCBROWSECOL FIELD LISTA->ROBS_     WIDTH 3  HEADER "CODE"         PARENT oBrowse

   DCBROWSECOL FIELD LISTA->CENA_MPR_ WIDTH 6  HEADER "PRICE"        PARENT oBrowse

   DCBROWSECOL FIELD LISTA->ROBK_     WIDTH 8  HEADER "BARCOD"       PARENT oBrowse
   DCBROWSECOL FIELD LISTA->MERA_     WIDTH 2  HEADER "UNIT"         PARENT oBrowse
                                               * MEASUREMENT UNIT
   DCBROWSECOL FIELD LISTA->DOBS_     WIDTH 4  HEADER "SUPP"         PARENT oBrowse
   DCBROWSECOL FIELD LISTA->VEZA_     WIDTH 15 HEADER "LINK"         PARENT oBrowse


tbr := 45
btt := tbr/3

@ 13.6,0 DCTOOLBAR oToolBar SIZE tbr,3 BUTTONSIZE btt,3

//-----------------------------------------------------------------------------
DCADDBUTTON CAPTION BMP_OK PARENT oToolBar SIZE 5 WHEN {||.F.}
DCADDBUTTON CAPTION "&Take it" PARENT oToolBar ACCELKEY xbeK_ENTER ;
   ACTION {|| sifra_artikla := ROBS_ ,DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList)} ;
   TOOLTIP "[Enter] Uzmi artikal - Take it article" ;
   CURSOR aCUR SIZE btt-5

DCADDBUTTON CAPTION BMP_HLP PARENT oToolBar SIZE 5 WHEN {||.F.}
DCADDBUTTON CAPTION "&Help" PARENT oToolBar ACCELKEY xbeK_F1 ;
   ACTION {|| selekcija_help(),  DC_GetRefresh(GetList), SetAppFocus(oBrowse)} ;
   TOOLTIP "[F1] Help" ;
   CURSOR aCUR SIZE btt-5

DCADDBUTTON CAPTION BMP_ESC PARENT oToolBar SIZE 5 WHEN {||.F.}
DCADDBUTTON CAPTION "&Exit" PARENT oToolBar ACCELKEY xbeK_ESC;
   ACTION {|| sifra_artikla := "", DC_ReadGuiEvent(DCGUI_EXIT_OK,GetList)} ;
   TOOLTIP "[Esc] Odustani od uzimanja - Stop taking it" ;
   CURSOR aCUR  SIZE btt-5

//-----------------------------------------------------------------------------


   // ---------------------- Prozor gore desno u AppWindow() -----------------

   // Funkcija si(oDlg)
   // daje aSize aktuelnog oDlg prozora, posle njegovog formiranja

      PRIVATE aSize    := {345,354} // aktuelni oDlg GUI prozor ovog programa
      PRIVATE aRefSize := SetAppWindow():currentSize()
      PRIVATE aRefPos  := SetAppWindow():currentPos()
      PRIVATE aPos     := GoreDesnoPos( aSize, aRefSize, aRefPos )   // xBazne.prg

         //MsgBox(str(aRefPos[1])+" "+str(aRefPos[2])+chr(13)+;
         //       str(aPos[1])   +" "+str(aPos[2]),"Pos oDlg")

   // ---------------------- Prozor gore desno u AppWindow() -----------------


DCGETOPTIONS ICON 1 ;
```

```
   AUTOFOCUS ;
   WINDOWCOL  aPos[1]  WINDOWROW  aPos[2] ;
   NORESIZE ;
   NOMAXBUTTON ;
   NOMINBUTTON ;
   NOESCAPEKEY ;
   FONT "11.Archivo Narrow Bold"


 DCREAD GUI TITLE "LIST FROM OBJECT ("+MAG+")" ;
            OPTIONS GetOptions ;
            FIT ;
            SETFOCUS @oBrowse ;
              PARENT @oDlg ;
              EVAL {||SetAppWindow(oDlg)} ;
              MODAL

SELECT "LISTA"
USE
SELECT(radno)


IF EMPTY(sifra_artikla)=.F.  // samo ako je sifra artikla preuzeta iz liste
   GO TOP
   LOCATE FOR alltrim(ROBS_) == alltrim(sifra_artikla)
ELSE
   // GO TOP --> NO
   // vrati se na aktuelni record spiska artikala
   // return to the current item list record
ENDIF

CLEAR TYPEAHEAD  // vraća fokus na objekat sa koga je fokus prenet ovde
RETURN NIL

*********************************************************************************
STATIC FUNCTION si(oDlg)
*********************************************************************************
// Pomocna funkcija - alat za programera
// Da bi se utvrdile dimenzije FIT aktivnog oDlg prozora, mora se, tek posle
// njegovog kreiranja sa DCREAD GUI FIT, pozvati ova funkcija da ga "izmeri".
// ove dimenzije se koriste kao gotove fiksne vrednosti npr. aSize := {300,400}
// u funkciji:
//   aPos := GoreDesnoPos( aSize, aRefSize, aRefPos )   // xBazne.prg
     LOCAL aSize  :=  oDlg:currentSize()                // dimenzije prozora
     MsgBox(str(aSize[1])+" "+str(aSize[2]),"oDlg Prozor")

RETURN(nil)
*********************************************************************************
STATIC FUNCTION selekcija_help()
*********************************************************************************
LOCAL cr := chr(59), cTxt := ;
"Izabere se artikal na filter listi artikala:"+cr+;
" "                                           +cr+;
" - Dupli klik mišem na izabrani artikal "    +cr+;
" - ENTER na izabrani artikal"                +cr+;
" - Klik na command button [ Take it ]"       +cr+;
" "                                           +cr+;
"i izabrani artikal biće pronađen u šifarniku"+cr+;
"artikala(select za preuzimanje u aplikaciju)"+cr+;
" "                                           +cr+;
```

```
"An article is selected in item filter list:" +cr+;
" "                                           +cr+;
" - Double mouse click on the selected item " +cr+;
" - ENTER to selected item"                   +cr+;
" - Click on the command button [ Take it ]"  +cr+;
" "                                           +cr+;
"and selected item will be found in codebook" +cr+;
"article (select to download to application)" +cr+;
" "
c__procitaj(cTxt,"POSEBNA SELEKCIJA | SPECIAL SELECTION",,"H")
CLEAR TYPEAHEAD// vraća fokus na objekat sa koga je fokus prenet ovde
RETURN(nil)
```

DODATAK – INSTALACIJA PROJEKTA
APPENDIX - PROJECT INSTALLATION

Aplikacije ARTICLES_CODEBOOK2.EXE i ARTICLES_CODEBOOK3.EXE imaju PostgreSQL bazu podataka. Ova baza podataka se formira na sledeći način:
The applications ARTICLES_CODEBOOK2.EXE and ARTICLES_CODEBOOK3.EXE have a PostgreSQL database. This database is formed as follows:

1. Instalira se u računar PostgreSQL server (verzija 9.4 ili novija). Za ove programe koristi se instalacija u kojoj se zada da je:

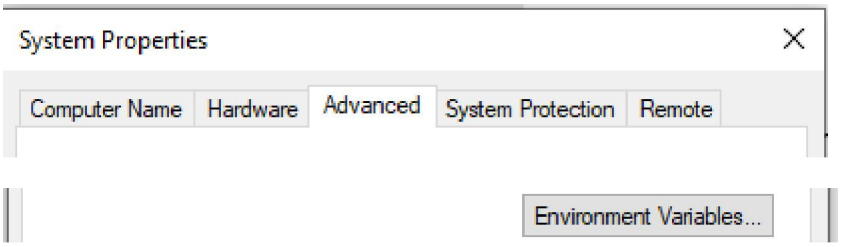
1. PostgreSQL server (version 9.4 or later) is installed in the computer. For these programs, an installation is used, in which it is assumed that:

```
server hostname     = 'Localhost'
port                = 5432
username            = 'postgres'
password            = 'coba#1949'
```

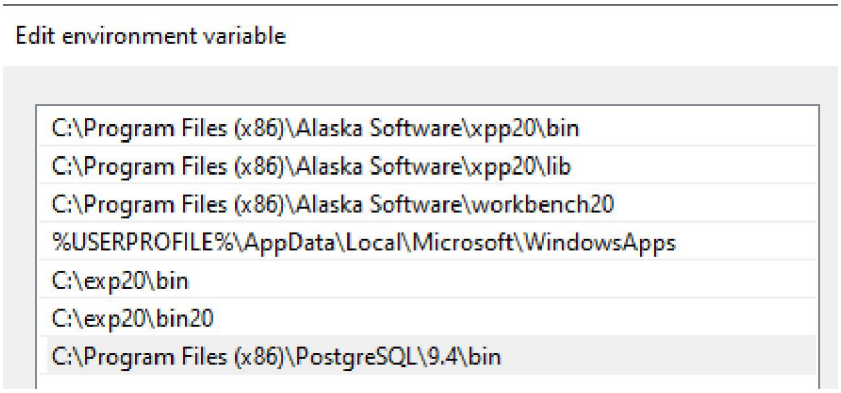
2. Instalira se u računar Alaska Xbase++ 2.0.1503 ili novija i odgovarajući eXpress++, tako da varijable okruženja budu u PATH-u

2. Installs in Alaska Xbase++ 2.0.1503 or later and corresponding eXpress++ computer, so environment variables are in PATH

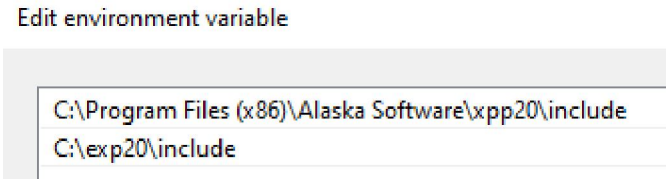
Evironment Variables moraju da postoje:
Environment Variables must exist:



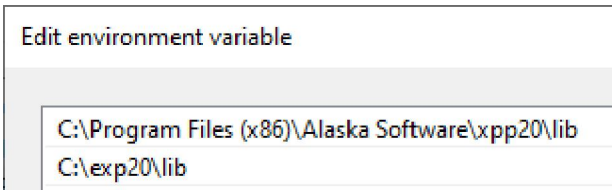
PATH



INCLUDE



LIB



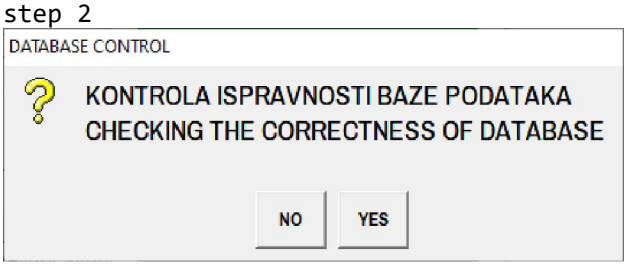
Folder CODEBOOK sa svim projektima u njemu postavi se na disk C: ili D: ili bilo koji, na primer C:\CODEBOOK

3. Startuje se servisna aplikacija C-POSTGRESQL-DATABASE.EXE i operacija

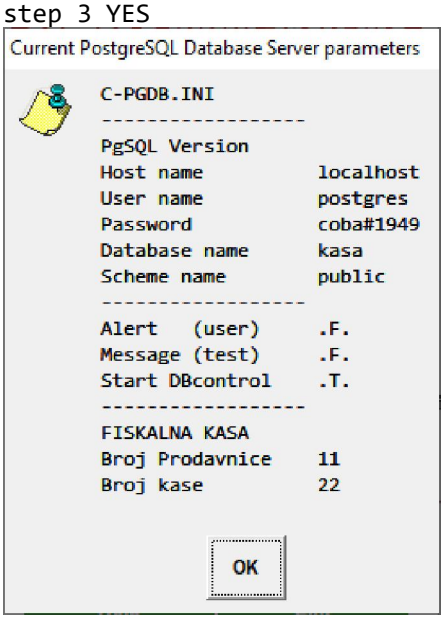
3. Service application C-POSTGRESQL-DATABASE.EXE is started and the operation is started

step 1 click

step 2

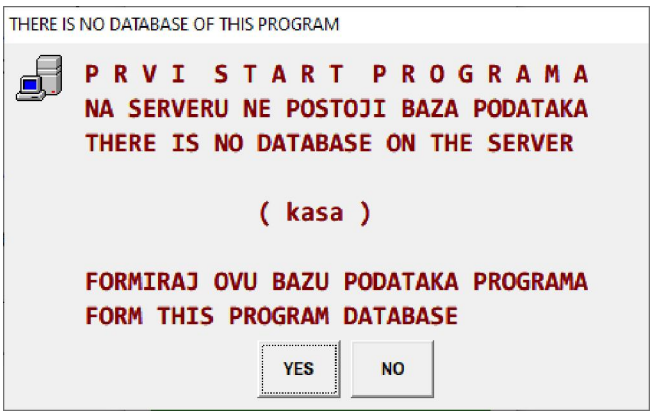


step 3 YES

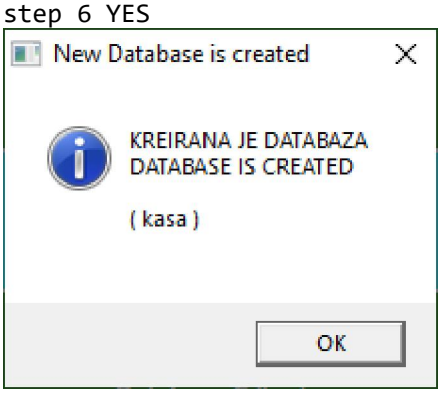


step 4 OK

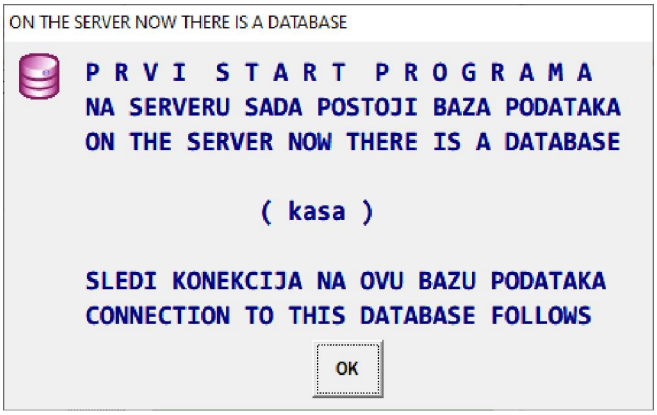
step 5 YES



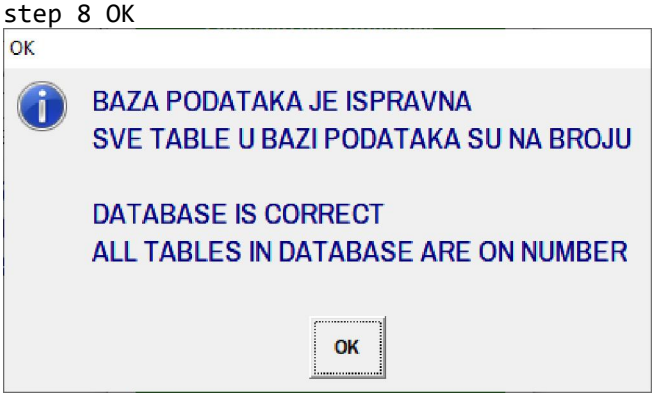
step 6 YES



step 7 OK

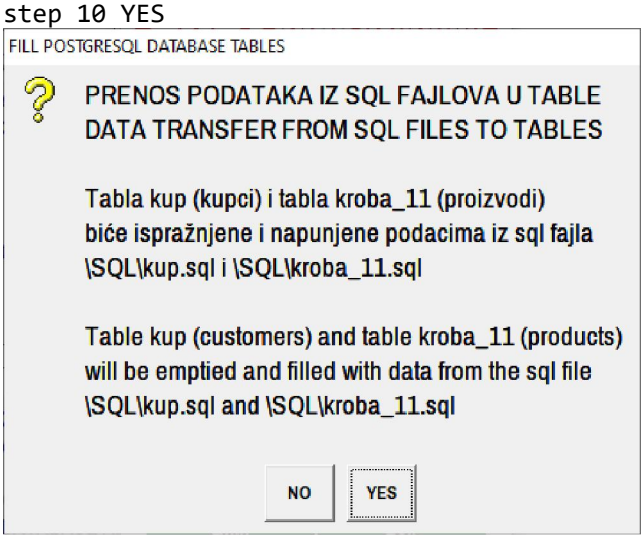


step 8 OK

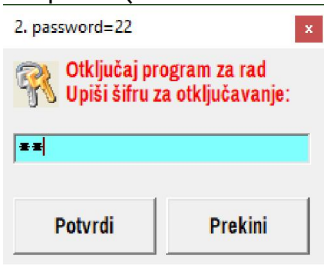


step 9 click

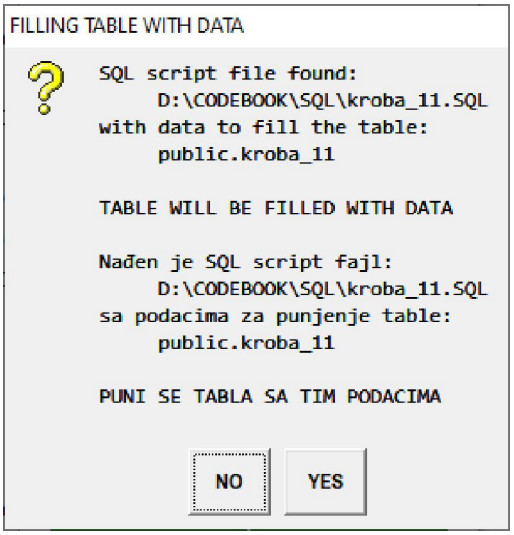
step 10 YES



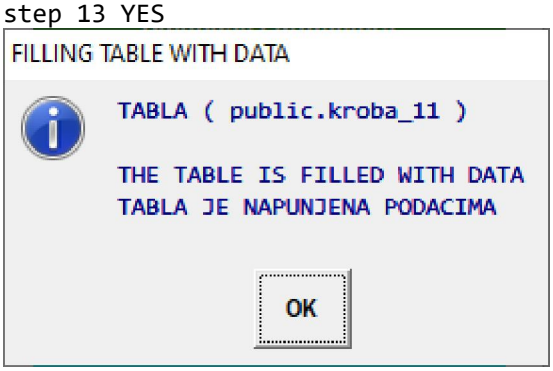
step 11 (number 22 -> ENTER or Potvrdi)



step 12



step 13 YES



step 14 OK

step 15
Izađe se iz aplikacije C-POSTGRESQL-DATABASE.EXE i startuje se jedna od dve aplikacije: aplikacija ARTICLES_CODEBOOK2.EXE ili ARTICLES_CODEBOOK3.EXE.
Ove dve aplikacije sada imaju svoju postgreSQL bazu podataka sa nazivom 'kasa' koja u šemi 'public' ima formirane sve potrebne upsize sql table, a jedna od njih je i tabla 'kroba_11' sa kojom rade ove dve aplikacije.

step 15
Exit the C-POSTGRESQL-DATABASE.EXE application and start one of two applications: the ARTICLES_CODEBOOK2.EXE or ARTICLES_CODEBOOK3.EXE application.
These two applications now have their own postgreSQL database with the name 'kasa', which has all the necessary upsize sql tables created in the 'public' schema, and one of them is the table 'kroba_11' with which these two applications work.

4. U opciji kada se ne koristi servisni program C-POSTGRESQL-DATABASE.EXE za postgreSQL bazu podataka aplikacije – tačka 3, već se jedna od dve aplikacije ARTICLES_CODEBOOK2.EXE ili ARTICLES_CODEBOOK3.EXE startuje odmah posle tačka 2, desiće se sledeće:
- sama aplikacija će izvršiti korake od step 2 do step 8 na isti način kao i servisna aplikacija za bazu podataka
- formiraće se postgreSQL baza podataka aplikacije kao prazna baza podataka – bez podataka u postgreSQL tablama
- Ako treba u table upisati demo podatke mora se to uraditi iz servisne aplikacije za bazu podataka C-POSTGRESQL-DATABASE.EXE, odnosno moraju se izvršiti koraci od step 9 do step 15.
iz opcije:

4. In the option when the service program C-POSTGRESQL-DATABASE.EXE is not used for the postgreSQL database of the application - point 3, but one of the two applications

ARTICLES_CODEBOOK2.EXE or ARTICLES_CODEBOOK3.EXE is started immediately after point 2, the following will happen:
- the application itself will perform steps from step 2 to step 8 in the same way as the service application for the database
- the application's postgreSQL database will be created as an empty database - without data in postgreSQL tables
- If you need to enter demo data in the tables, it must be done from the service application for the database C-POSTGRESQL-DATABASE.EXE, that is, steps from step 9 to step 15 must be performed from the option:

Dodatno uz tačku 4.
In addition to point 4.

Da bi aplikacija izvršila operacije navedene u tački 4. u aplikacije ARTICLES_CODEBOOK2.EXE i ARTICLES_CODEBOOK3.EXE mora se linkovati program: **__tab__create__app.prg**
a fajl C-PGDB.INI mora imati upisan parametar: _lDBcontrol_=1

Ovaj program sadrži tehnologiju za kreiranje i kontrolu postgreSQL tabli za bazu podataka aplikacije na osnovu podataka koje preuzima iz externog SQL fajla sa podacima. Pošto svaka aplikacija ima drugačiju bazu podataka i ovaj prg fajl za svaku aplikaciju ima drugačiji kod, a takođe i eksterni SQL fajl ima drugačiji sadržaj.

Samo ako je ovaj prg linkovan sa aplikacijom može se koristiti kontrola baze podataka upisana u main() proceduri aplikacije (slika control):

In order for the application to perform the operations specified in point 4, the program: **__tab__create__app.prg** must be linked to the ARTICLES_CODEBOOK2.EXE and ARTICLES_CODEBOOK3.EXE applications, and the C-PGDB.INI file must have the following parameter: _lDBcontrol_=1

This program contains technology for creating and controlling postgreSQL tables for the application's database based on the data it retrieves from an external SQL data file. Since each application has a different database, this prg file for each application has a different code, and also the external SQL file has a different content.

Only if this prg is linked to the application can the database control written in the main() procedure of the application be used (picture control):

Picture control

```
PROCEDURE MAIN()
  *====================================================
  * CONTROL APP DATABASE - CONNECT-CONTROL-DISCONNECT
  __pg__ini()  // C-PGDB.DLL
  *  databases PUBLIC variables
  * _server_,_uid_,_pwd_,_db_,_she_,_tab_,_oSession_,
  * _dblist_,_shelist_,_tablist_,lAlert,lMsg,_lDBcontrol_
  * _MAG_,_KAS_,
  * _version_, _cConnStr_, cConnStrTest_ , _cdbc_
    IF _lDBcontrol_=.T. // C-PGDB.INI
      // vrši se kontrola PgSQL database
      // PgSQL database control is performed
```

```
      // control database and create database if not exists
        qq := __pg__connect_app() // C-PGDB.DLL
        * qq=.T. or qq=.F.
    ELSE
      // ne vrši se kontrola PgSQL database
      // PgSQL database control is not performed
      // Aplikacija nastavlja rad bez kontrole databaze
      // Application continues to operate without database control
        qq:=.T.
        *" N A S T A V A K  P R O G R A M A
        *" DATABAZA JE OK SVE TABLE SU OK
        *" CONTINUE PROGRAM
        *" DATABASE IS OK ALL TABLES IS OK
    ENDIF
    IF qq==.F.
       __pg__appquit() // C-PGDB.DLL
       *" P R E K I D  P R O G R A M A
       *" NEDOSTAJU TABLE U DATABAZI
       *" STOP PROGRAM
       *" TABLES IN DATABASE ARE MISSING
    ENDIF
  *=================================================
```

SERVISNA APLIKACIJA ZA POSTGRESQL BAZU PODATAKA

Servisna aplikacija za postgreSQL bazu podataka C-POSTGRESQL-DATABASE.EXE ima na svom startu ovaj kod i u tu aplikaciju je linkovan i program **__tab__create__app.prg.**

Servisna aplikacija je dobra praksa. Ova aplikacija se posebno pravi i isporučuje za svaku poslovnu aplikaciju čime se rasterećuje i uprošćava kod poslovne aplikacije i iz izbora rada koji se daje korisniku uklanjaju se opcije vezane za bazu podataka koje se vrše vrlo retko i to od strane administratora. Dobra praksa je da za aplikaciju CUSTOMERS.EXE naziv servisne aplikacije bude CUSTOMERS-DATABASE.EXE. Ovde je dat pun kod servisne aplikacije C-POSTGRESQL-DATABASE.EXE koji je isti za bilo koju servisnu aplikaciju – različiti su samo programi **__tab__create__app.prg**.

SERVICE APPLICATION FOR POSTGRESQL DATABASE

The service application for the postgreSQL database C-POSTGRESQL-DATABASE.EXE has this code at its start, and the program **__tab__create__app.prg** is also linked to that application.

A service application is a good practice. This application is specially made and delivered for each business application, which relieves and simplifies the business application, and from the choice of work given to the user, options related to the database are removed, which are performed very rarely by the administrator. It is good practice for the CUSTOMERS.EXE application to name the service application CUSTOMERS-DATABASE.EXE. Here is the full code of the service application C-POSTGRESQL-DATABASE.EXE which is the same for any service application - only the **__tab__create__app.prg** programs are different.

ZAKLJUČAK - CONCLUSION

1.
PGDBE tehnologija napravljena je da programski jezik Alaska Xbase++ može da kao bazu podataka koristi postgreSQL bazu podataka kroz PGDBE SQL komande i funkcije i kroz PGDBE ISAM simulirane komande i funkcije.
1.
PGDBE technology is designed so that the Alaska Xbase++ programming language can use the postgreSQL database as a database through PGDBE SQL commands and functions and through PGDBE ISAM simulated commands and functions.

2.
U PGDBE je ugrađena i Dodatna UPSIZE tehnologija. UPSIZE tehnologija napravljena je za jednokratnu upotrebu. Cilj te tehnologije je da se stare postojeće Alaska Xbase++ aplikacije sa ISAM (DBF-DBT-NTX-FPT-CDX) bazom podataka, UZ MALE IZMENE U KODU prenesu na PostgreSQL bazu podataka. Jasno je da će se to uraditi samo jednom i nikad više.
2.
Additional UPSIZE technology is built into PGDBE. UPSIZE technology is made for single use. The goal of this technology is to transfer old existing Alaska Xbase++ applications with ISAM (DBF-DBT-NTX-FPT-CDX) database to the PostgreSQL database, WITH SMALL CHANGES IN THE CODE. It is clear that this will only be done once and never again.

3.
Još jedan Cilj UPSIZE tehnologije je i da jedna Xbase++ aplikacija može istovremeno da radi sa fajlovima iz ISAM DBF baze podataka i sa tablama iz postgreSQL baze podataka i da te dve baze podataka mogu međusobno da razmenjuju podatke.
3.
Another goal of UPSIZE technology is that one Xbase++ application can simultaneously work with files from the ISAM DBF database and with tables from the PostgreSQL database and that these two databases can exchange data with each other.

4.
Aplikacija ARTICLES-CODEBOOK3.EXE napravljena je prema postavci iz tačke 1.
4.
The ARTICLES-CODEBOOK3.EXE application was created according to the setting from point 1.

5.
Poređenjem posla na pravljenju aplikacije prema postavci iz tačke 2. (ARTICLES-CODEBOOK2.EXE) i prema postavci iz tačke 1. (ARTICLES-CODEBOOK3.EXE) došao sam do sledećeg zaključka (a ne mora da znači da će to biti i vaš zaključak):

- Aplikacija pravljena po tački 1. je mnogo bolja. Dobija se mnogo bolja i brža aplikacija, sigurnija u radu sa PgSQL bazom podataka, otvorena za upotrebu moćnih stvari koje pruža SQL jezik i SQL baza podataka.

- Aplikacija pravljena po tački 1. ima mnogo više izmena od aplikacije pravljene po tački 2. ali prednosti koje se dobijaju aplikacijom iz tačke 1. opravdavaju taj posao.

- Interfejs stare Alaska Xbase++ DBFNTX aplikacije ostaje isti i funkcionalan je i u aplikaciji iz tačke 1. i u aplikaciji iz tačke 2.

- U aplikaciji iz tačke 1. mora se odustati od indeksa na ISAM način, mora se odustati od filtera na ISAM način i moraju se databaze CRUD (CREATE, READ, UPDATE, DELETE) komande izvoditi preko SQL jezika. Dakle, umesto APPEND BLANK, REPLACE, DELETE, PACK, DCGET, DCSAY (komande koje rade sa poljima u DBF tabli) moraju se koristiti SQL komande. Ova lista nije konačna i možda je u nekim situacijama netačna, ali ovo je ono što je osnovno. Tek daljim testiranjem imaću nove informacije. Za sada je i ovo dovoljno, jer se dobija visoko

profesionalna poslovna aplikacija koja koristi PostgreSQL bazu podataka. Pošto mi je to bila stara želja, smatram da sam je sada potpuno ispunio.

5.
By comparing the work of creating the application according to the setting from point 2. (ARTICLES-CODEBOOK2.EXE) and according to the setting from point 1. (ARTICLES-CODEBOOK3.EXE) I came to the following conclusion (and it does not necessarily mean that this will be your conclusion as well):

- The application made according to point 1 is much better. The result is a much better and faster application, more secure in working with the PgSQL database, open to use the powerful things provided by the SQL language and the SQL database.

- The application made according to point 1. has many more changes than the application made according to point 2. but the advantages obtained by the application from point 1. justify the work.

- The interface of the old Alaska Xbase++ DBFNTX application remains the same and is functional in both the application from point 1 and the application from point 2.

- In the application from point 1, indexes must be dropped in the ISAM way, filters must be dropped in the ISAM way, and database CRUD (CREATE, READ, UPDATE, DELETE) commands must be executed via the SQL language. Therefore, instead of APPEND BLANK, REPLACE, DELETE, PACK, DCGET, DCSAY (commands that work with fields in the DBF table), SQL commands must be used. This list is not exhaustive and may be incorrect in some situations, but this is the gist. Only with further testing will I have new information. For now, this is enough, because a highly professional business application that uses the PostgreSQL database is obtained. Since it was an old wish of mine, I consider that I have now completely fulfilled it.

NOTE

Na kraju ovog Dela 5b, uz ovu knjigu prilažem za download izvorni kod projekata: ARTICLES-CODEBOOK1.EXE, ARTICLES-CODEBOOK2.EXE, ARTICLES-KODEBOOK3.EXE u fajlu:
Alaska Xbase++ To PostgreSQL (Part 5a-5b).zip

Zadnji Deo 5c ove knjige
Prikazuje aplikaciju KASA-STOLOVI.EXE koja u sebi sadrži i aplikaciju ARTICLES-CODEBOOK.EXE urađenu po tehnologiji iz tačke 2 (ARTICLES-CODEBOOK2.EXE)
Na kraju ovog Dela 5c, uz ovu knjigu prilažem za download izvorni kod projekata: KASA-STOLOVI.EXE, u fajlu:
Alaska Xbase++ To PostgreSQL (Part 5c).zip

At the end of this Part 5b, along with this book, I attach for download the source code of the projects: ARTICLES-CODEBOOK1.EXE, ARTICLES-CODEBOOK2.EXE, ARTICLES-KODEBOOK3.EXE in the file:
Alaska Xbase++ To PostgreSQL (Part 5a-5b).zip

Last Part 5c of this book
It shows the KASA-STOLOVI.EXE application, which also contains the ARTICLES-CODEBOOK.EXE application made according to the technology from point 2 (ARTICLES-CODEBOOK2.EXE).
At the end of this Part 5c, along with this book I attach the source code of the project: KASA-STOLOVI.EXE, in the file:
Alaska Xbase++ To PostgreSQL (Part 5c).zip