CSYSTEMS™

PROGRAMSKI PAKET ZA KNJIGOVODSTVO

COBA Systems

Alaska Xbase++
DBF to PostgreSQL database application

(PART 5)
ARTICLES-CODEBOOK1.EXE
ARTICLES-CODEBOOK2.EXE
ARTICLES-CODEBOOK3.EXE

01.11.2023

UVOD
INTODUCTION

U delu 1 ove knjige opisan je PGDBE (PostGreDatabaseEngine) za rad Alaska Xbase++ aplikacija sa PostgerSQL databazom. Opisana je i UPSIZE tehnologija koja se može primeniti uz PGDBE, tako da se sa PostgreSQL databazom može komunicirati putem ISAM komandi i funkcija i putem SQL komandi i funkcija. Navedena su 3 programa koji se koriste u COBA Systems poslovnim aplikacijama za rad sa PostgreSQL bazom podataka.

Part 1 of this book describes PGDBE (PostGreDatabaseEngine) for the operation of Alaska Xbase++ applications with the PostgerSQL database. UPSIZE technology, which can be applied with PGDBE, is also described, so that it is possible to communicate with the PostgreSQL database through ISAM commands and functions and through SQL commands and functions. There are 3 programs used in COBA Systems business applications for working with the PostgreSQL database.

U delu 2 ove knjige je opisan i dokumentovan prvi od ta tri programa:
In part 2 of this book, the first of those three programs is described and documented:

C-DBF2SQLFILE.EXE (developer service program)

U delu 3 ove knjige opisan je i dokumentovan je program
Part 3 of this book describes and documents the program

C-POSTRGRESQL-DATABASE.EXE (supplementary program with each EXE application)

U delu 4 ove knjige opisana je i dokumentovana je bublioteka funkcija za komunikaciju sa Alaska Xbase++ PostgreSQL upsize bazom podataka.
Part 4 of this book describes and documents the library of functions for communicating with the Alaska Xbase++ PostgreSQL upsize database.

C-PGDB.DLL (common procedures and functions for applications)

Ovaj program se sastoji od skupa common funkcija i isporučuje se uz postgreSQL bazu podataka svake poslovne EXE aplikacije. Služi za komunikaciju sa bazom podataka (sa bilo kojom PostgreSQL bazom podataka).
This program consists of a set of common functions and is supplied with the PostgreSQL database of every enterprise EXE application. It serves for communication with the database (with any PostgreSQL database).

U delu 5 ove knjige opisana je i dokumentovana poslovna aplikacija koja je sa Alaska Xbase++ DBF ISAM baze podataka preneta na Alaska Xbase++ PostgreSQL upsize bazu podataka.
In part 5 of this book, a business application that was transferred from the Alaska Xbase++ DBF ISAM database to the Alaska Xbase++ PostgreSQL upsize database will be described and documented.

KASA-STOLOVI.EXE (Fiscal register cash register for a restaurant)

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 od aplikacije KASA-STOLOVI.EXE. A na kraju će biti prikazana i aplikacija KASA-STOLOVI.EXE.

Zbog toga je za ovaj prethodni 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 than the KASA-STOLOVI.EXE (CASH-TABLES.EXE) application. And at the end, the KASA-STOLOVI.EXE application will be displayed.

Therefore, for this previous view, the application ARTICLES-CODEBOOK.EXE was chosen, 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 i ISAM komandi i funkcija, već samo uz korišćenje PGDBE SQL komandi i funkcija.

Part 5b

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

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)

APLIKACIJA ARTICLES-CODEBOOK.EXE

Funkcija:

Registar artikala ili Šifarnik artikala mora da omogući korisniku sledeće funkcije:

1. Pregled i štampa spiska artikala i rukovanje i upravljanje spiskom artikala
2. Dodavanje novog artikla na spisak artikala
3. Brisanje artikla iz spiska artikala
4. Formiranje registar kartice svakog artikla sa svim podacima o artiklu
5. Pregled, štampa i editovanje registar kartice artikla
6. Formiranje i upotreba spiska artikala sa svim artiklima u spisku
7. Formiranje i upotreba spiska artikala samo sa zadatom grupom artikala
8. Pretragu spiska artikala po nazivu artikla, po šifri artikla, po barkodu artikla, po kataloškom broju artikla, po šifri dobavljača, po šifri veze, po reči koja se nalazi bilo gde u nazivu artikla bez obzira na velika i mala slova
9. Slanje šifre i naziva i ostalih podataka izabranog artikla iz spiska artikala u eksterni ASCII txt fajl iz koga te podatke može preuzeti bilo koja poslovna aplikacija

Function:

The register of articles or the Codebook of articles must enable the user the following functions:

1. Viewing and printing the list of articles and handling and managing the list of articles
2. Adding a new aricel to the list of articels
3. Deleting an article from the list of articles
4. Forming a register card of each article with all information about the article
5. Viewing, printing and editing of the article register card
6. Formation and use of the list of articles with all articles in the list
7. Formation and use of a list of articles only with a given group of articles
8. Searching the list of articles by the name of the article, by the code of the article, by the barcode of the article, by the catalog number of the article, by the code of the supplier, by the code of the link, by the word that is found anywhere in the name of the article, regardless of upper and lower case letters
9. Sending the code and name and other data of the selected article from the list of articles to an external ASCII txt file from which any business application can download this data

Ova aplikacija urađena je u tri varijante:
This application is made in three variants:

1. **ARTICLE-CODEBOOK1.EXE** (AlaskaXbase++/eXpress++/ISAM DBFNTX engine)

2. **ARTICLE-CODEBOOK2.EXE** (AlaskaXbase++/eXpress++/SQL PGDBE + ISAM UPSIZE engine)

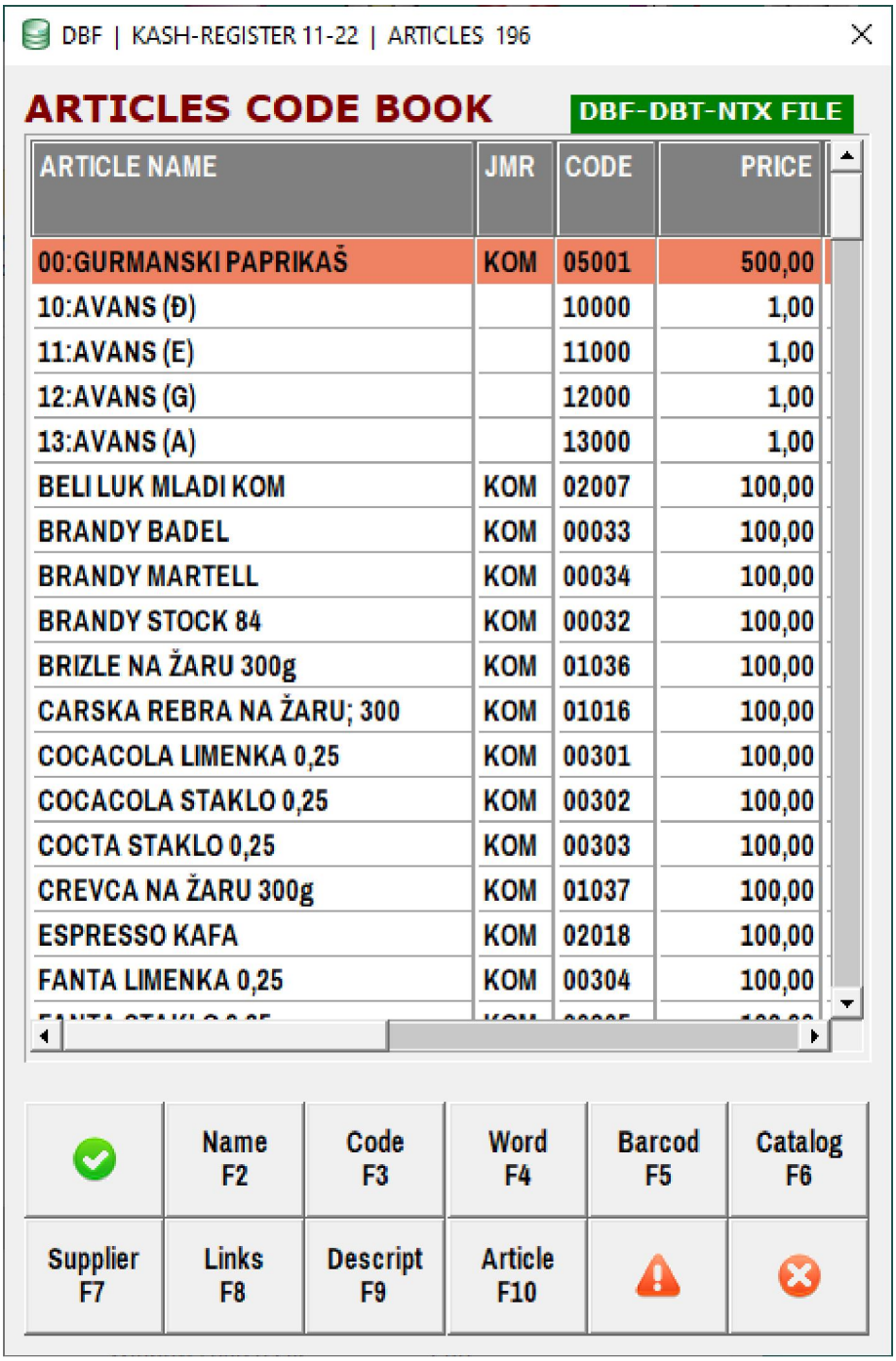
3. **ARTICLE-CODEBOOK3.EXE** (AlaskaXbase++/eXpress++/SQL PGDBE engine)

Ovde je dat poseban kod za sve tri aplikacije tako da se može vršiti poređenje i sagledavanje izmena u tom kodu kod svake od ovih varijanti. Na osnovu toga programer

može doneti svoje zaključke i svoju odluku o tome kako će svoje DBF aplikacije prevoditi na postgreSQL aplikacije.

A special code for all three applications is given here so that you can compare and see the changes in that code for each of these variants. Based on this, the developer can make his own conclusions and his decision on how to translate his DBF applications to postgreSQL applications.

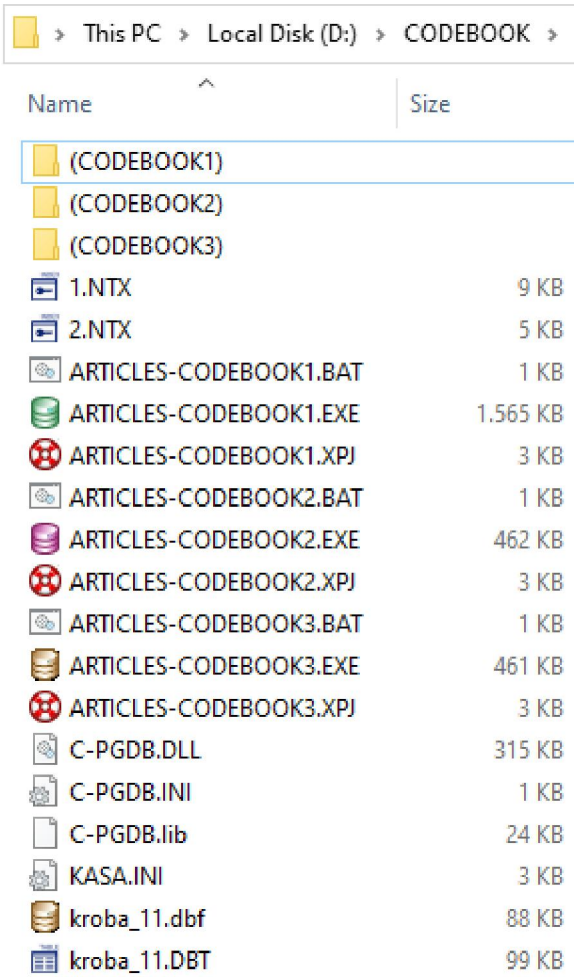
Picture 1. ARTICLE-CODEBOOK1.EXE



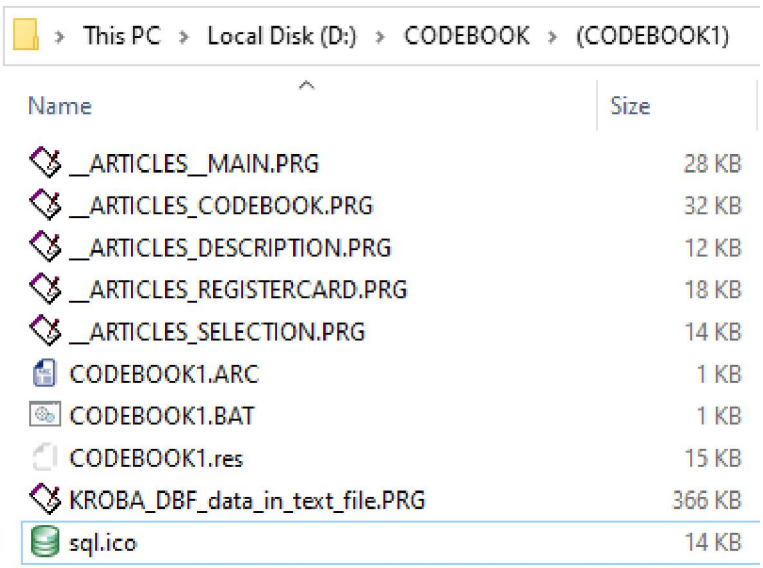
Picture 2. ARTICLE-CODEBOOK2.EXE

Picture 3. ARTICLE-CODEBOOK3.EXE

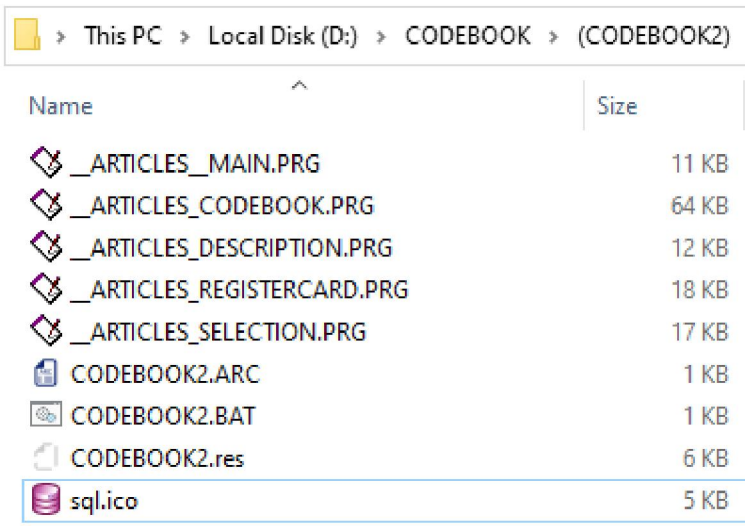
Picture 4. PROJECT ARTICLES-CODEBOOK.EXE



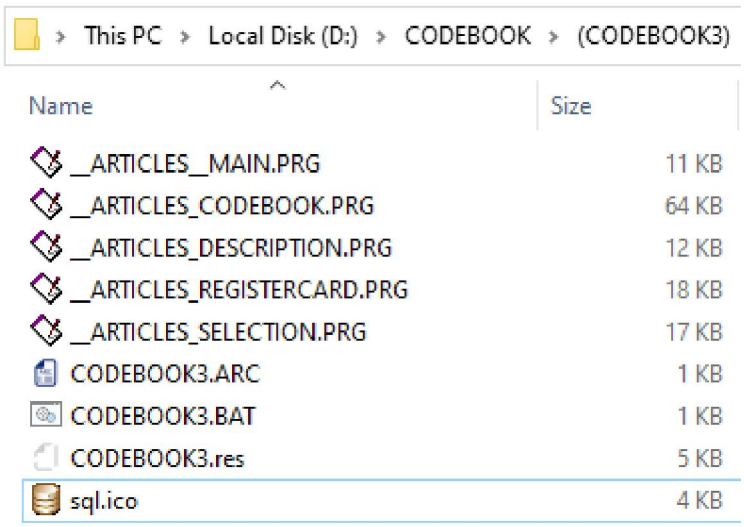
Picture 5. PROJECT ARTICLES-CODEBOOK1.EXE



Picture 6. PROJECT ARTICLES-CODEBOOK2.EXE



Picture 7. PROJECT ARTICLES-CODEBOOK3.EXE



```
// ARTICLES-CODEBOOK1.XPJ --- START

[PROJECT]
    COMPILE       = xpp
    COMPILE_FLAGS = /q
    DEBUG         = no
    GUI           = yes
    LINKER        = alink
    LINK_FLAGS    =
    OBJ_DIR       = __________ARTICLES-CODEBOOK1.EXE
    RC_COMPILE    = arc
    RC_FLAGS      = /v
    ARTICLES-CODEBOOK1.XPJ

[ARTICLES-CODEBOOK1.XPJ]
 ARTICLES-CODEBOOK1.EXE
[ARTICLES-CODEBOOK1.EXE]
// $START-AUTODEPEND
//---------------------------------
//  (CODEBOOK1)
//---------------------------------
  __ARTICLES__MAIN.OBJ  // MAIN()
  __ARTICLES_CODEBOOK.OBJ
  __ARTICLES_SELECTION.OBJ
  __ARTICLES_DESCRIPTION.OBJ
  __ARTICLES_REGISTERCARD.OBJ
  KROBA_DBF_data_in_text_file.OBJ

// $STOP-AUTODEPEND
  (CODEBOOK1)\CODEBOOK1.RES
  XBTBASE1.LIB              // XbToolsIII
  XBTBASE2.LIB              // XbToolsIII
  dclipx.lib                // eXpress++

  BAZNE.LIB
  BAZNEx.LIB
// KASA.INI
//---------------------------------
//  (CODEBOOK1)
//---------------------------------
  (CODEBOOK1)\__ARTICLES__MAIN.PRG
  (CODEBOOK1)\__ARTICLES_CODEBOOK.PRG
  (CODEBOOK1)\__ARTICLES_SELECTION.PRG
  (CODEBOOK1)\__ARTICLES_DESCRIPTION.PRG
  (CODEBOOK1)\__ARTICLES_REGISTERCARD.PRG
  (CODEBOOK1)\KROBA_DBF_data_in_text_file.PRG

// ARTICLES-CODEBOOK1.XPJ --- END

// ARTICLES-CODEBOOK1.BAT --- START
@ECHO OFF
PBUILD.EXE ARTICLES-CODEBOOK1.XPJ > ARTICLES-CODEBOOK1________.TXT
NOTEPAD ARTICLES-CODEBOOK1________.TXT
// ARTICLES-CODEBOOK1.BAT --- END
```

```
// ARTICLES-CODEBOOK2.XPJ --- START

[PROJECT]
    COMPILE       = xpp
    COMPILE_FLAGS = /q /dUSE_POSTGRES
    DEBUG         = no
    GUI           = yes
    LINKER        = alink
    LINK_FLAGS    =
    OBJ_DIR       = __________ARTICLES-CODEBOOK2.EXE
    RC_COMPILE    = arc
    RC_FLAGS      = /v
    ARTICLES-CODEBOOK2.XPJ

[ARTICLES-CODEBOOK2.XPJ]
ARTICLES-CODEBOOK2.EXE
[ARTICLES-CODEBOOK2.EXE]
// $START-AUTODEPEND
//---------------------------------
//  (CODEBOOK2)
//---------------------------------
  __ARTICLES__MAIN.OBJ  // MAIN()
  __ARTICLES_CODEBOOK.OBJ
  __ARTICLES_SELECTION.OBJ
  __ARTICLES_DESCRIPTION.OBJ
  __ARTICLES_REGISTERCARD.OBJ

// $STOP-AUTODEPEND
  (CODEBOOK2)\CODEBOOK2.RES
  XBTBASE1.LIB              // XbToolsIII
  XBTBASE2.LIB              // XbToolsIII
  dclipx.lib                // eXpress++

  BAZNE.LIB
  BAZNEx.LIB
  C-PGDB.LIB

// C-PGDB.INI
// KASA.INI
//---------------------------------------------------------
//  (CODEBOOK2)
//---------------------------------------------------------
  (CODEBOOK2)\__ARTICLES__MAIN.PRG
  (CODEBOOK2)\__ARTICLES_CODEBOOK.PRG
  (CODEBOOK2)\__ARTICLES_SELECTION.PRG
  (CODEBOOK2)\__ARTICLES_DESCRIPTION.PRG
  (CODEBOOK2)\__ARTICLES_REGISTERCARD.PRG

// ARTICLES-CODEBOOK2.XPJ --- END


// ARTICLES-CODEBOOK2.BAT --- START
@ECHO OFF
PBUILD.EXE ARTICLES-CODEBOOK2.XPJ > ARTICLES-CODEBOOK2________.TXT
NOTEPAD ARTICLES-CODEBOOK2________.TXT
// ARTICLES-CODEBOOK2.BAT --- END
```

PRG FOR ARTICLES-CODEBOOK1.EXE

__ARTICLES__MAIN.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES__MAIN.PRG                              DBF-DBT-NTX    //
//                                                                    //
//   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++  D B F N T X  DATABASE
 ŠIFARNIK SVIH ARTIKALA PRODAJNOG OBJEKTA ILI RESTORANA
 SA FILTEROM PO GRUPI ARTIKALA

 APPLICATION (main program)
 Alaska Xbase++ and eXpress++  D B F N T X  DATABASE
 ARTICLES OF SALES OBJECT OR RESTAURANT
 WITH FILTER BY GROUP OF ARTICLES

-------------------------------
 PROCEDURE AppSys()
 PROCEDURE dbesys()
 PROCEDURE main()
   FUNCTION DBF_CONTROL()
   FUNCTION KROBA_create(SPISAK_ROBE)
   FUNCTION KROBA_DBF_data_from_text_file(SPISAK_ROBE)

   FUNCTION KROBA_DBF_data_in_text_file(SPISAK_ROBE)
   FUNCTION KROBA_variables(nIzbor)
   FUNCTION KROBA_variables_from_fields()
```

```
   FUNCTION KROBA_variables_to_fields()
   FUNCTION KROBA_index(SPISAK_ROBE)


*/

*-----------------------------------
* Xbase++
#include "Xbp.ch"
#include "AppEvent.ch"
#include "Common.ch"
*-----------------------------------
* XbToolIII++
#include "xbtsys.ch"
*-----------------------------------
* eXpress++
#include "dcdialog.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 := "DBF"

* --- 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 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
  *======================================================================
 // --- sales object code and point of sale code --- start

   PUBLIC _MAG_ := "11", _KAS_ := "22"
   PUBLIC cObjekat_Kase, cBroj_Kase

    _MAG_ := INI_READ("C","KASA_CFG","cObjekat_Kase",gde_exe()+"\KASA.INI")
    _KAS_ := INI_READ("C","KASA_CFG","cBroj_Kase"   ,gde_exe()+"\KASA.INI")
   IF EMPTY(_MAG_)
      INI_WRITE("C","KASA_CFG","cObjekat_Kase","11",gde_exe()+"\KASA.INI")
      _MAG_ := INI_READ("C","KASA_CFG","cObjekat_Kase",gde_exe()+"\KASA.INI")
   ENDIF
   IF EMPTY(_KAS_)
      INI_WRITE("C","KASA_CFG","cBroj_Kase","22",gde_exe()+"\KASA.INI")
      _KAS_ := INI_READ("C","KASA_CFG","cBroj_Kase",gde_exe()+"\KASA.INI")
   ENDIF

   _MAG_ := STRZERO( VAL(_MAG_), 2 )   // od '00' do '99'
   _KAS_ := STRZERO( VAL(_KAS_), 2 )   // od '00' do '99'

   cObjekat_Kase := _MAG_    // šifra profajnog objekta  - sales object code
   cBroj_Kase    := _KAS_    // šifra prodajnog mesta    - point of sale code

   * INI_READ(), INI_WRITE() // BAZNE.DLL

  // --- sales object code and point of sale code --- end

  // D A T A B A S E

   SPISAK_ROBE  := "kroba_"+_MAG_+".dbf"  // kroba_11.dbf - kroba_11.dbt
   SPISAK_ROBE1 := "1.ntx"                // INDEX ON ROBN_ TO 1.ntx
   SPISAK_ROBE2 := "2.ntx"                // INDEX ON ROBS_ TO 2.ntx


    _lDBcontrol_ := .T.
    IF _lDBcontrol_=.T.
      // control DBF-DBT-NTX database
      // kontrola database or create database if not exists
        ********************
        qq := DBF_CONTROL() // here
        ********************
        * qq=.T. or qq=.F.
    ELSE
      // no control DBF-DBT-NTX 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.
       ***************
       quit_quit() // bazne.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()
*" establish connection on input to main()

* 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()
*" disconnect on exit from main()


****************
 quit_quit() // disconnect all sessions bazne.dll
****************
RETURN









* FUNCTION DBF_CONTROL()
* FUNCTION KROBA_create(SPISAK_ROBE)
* FUNCTION KROBA_index(SPISAK_ROBE)
* FUNCTION KROBA_DBF_data_from_text_file(SPISAK_ROBE)

* FUNCTION KROBA_DBF_data_in_text_file(SPISAK_ROBE)
* FUNCTION KROBA_variables(nIzbor)
* FUNCTION KROBA_variables_from_fields()
* FUNCTION KROBA_variables_to_fields()



*************************************************************************
FUNCTION DBF_CONTROL()
*************************************************************************
LOCAL radno:=SELECT()

 *  KROBA_11.DBF - LISTA POLJA
  *
  *     1 GRUP_      C    2 0 ____________________________
  *     2 ROBS_      C    5 0 ____________________________
  *     3 ROBN_      C   25 0 ____________________________
  *     4 ROBK_      C   25 0 ____________________________
  *     5 MERA_      C    3 0 ____________________________
  *     6 TARI_      C    1 0 ____________________________
  *     7 PPUP_      N    5 1 ____________________________
  *     8 PPAP_      N   10 2 ____________________________
  *     9 CENA_FAK_  N   13 2 ____________________________
```

```
  *    10 CENA_NAB_  N   13 2 _____________________________
  *    11 CENA_VPR_  N   13 2 _____________________________
  *    12 CENA_MPR_  N   13 2 _____________________________
  *    13 CENA_DEV_  N   13 2 _____________________________
  *    14 ZALI_      N   15 3 _____________________________
  *    15 DAT0_      D    8 0 _____________________________
  *    16 MAGS_      C    2 0 _____________________________
  *    17 ZNAK_      C    1 0 _____________________________
  *    18 FLAG_      C    1 0 _____________________________
  *    19 STAT_      C    1 0 _____________________________
  *    20 CENA_PRO_  N   13 3 _____________________________
  *    21 OSNO_AKC_  N   13 3 _____________________________
  *    22 ROK_       N    3 0 _____________________________
  *    23 DOBS_      C    4 0 _____________________________
  *    24 DOBK_      C   25 0 _____________________________
  *    25 VEZA_      C   15 0 _____________________________
  *    26 NAME_      C  200 0 _____________________________
  *    27 OPIS_      M   10 0 _____________________________
  *                       452

lRet:=.T.

  IF FILE(SPISAK_ROBE)=.F.
     KROBA_create(SPISAK_ROBE)
  ENDIF
  IF FILE("1.NTX")=.F. .OR. FILE("2.NTX")=.F.
     KROBA_index(SPISAK_ROBE)
  ENDIF

  IF FILE(SPISAK_ROBE)=.F.
     RETURN .F.
  ENDIF
  IF FILE("1.NTX")=.F. .OR. FILE("2.NTX")=.F.
     RETURN .F.
  ENDIF


  USE (SPISAK_ROBE) NEW SHARED ALIAS "KR"
  COUNTER:=reccount()
  USE
  IF COUNTER=0
     IF C__NEDA("DATABAZA JE PRAZNA - NAPUNITE JE DEMO PODACIMA" +chr(59)+;
                "THE DATABASE IS EMPTY - FILL IT WITH DEMO DATA" +chr(59)+;
                "","DATABASE IS EMPTY",,"G3")==.F.
        RETURN .T.
     ENDIF
     ****************************************
     KROBA_DBF_data_from_text_file(SPISAK_ROBE)
     ****************************************
     C__PORUKA("DATABAZA SADRŽI DEMO PODATKE" +chr(59)+;
               "DATABASE CONTAINS DEMO DATA"  +chr(59)+;
               "","DATABASE IS FULL",,"G3")
  ENDIF

RETURN .T.

  ***************************************************************************
  FUNCTION KROBA_create(SPISAK_ROBE)
  ***************************************************************************
```

```
      LOCAL radno := SELECT(), astructure := {}
      SELECT 0
      aStructure := { ;
        { "GRUP_"       ,"C",   2, 0 } ,;
        { "ROBS_"       ,"C",   5, 0 } ,;
        { "ROBN_"       ,"C",  25, 0 } ,;
        { "ROBK_"       ,"C",  25, 0 } ,;
        { "MERA_"       ,"C",   3, 0 } ,;
        { "TARI_"       ,"C",   1, 0 } ,;
        { "PPUP_"       ,"N",   5, 1 } ,;
        { "PPAP_"       ,"N",  10, 2 } ,;
        { "CENA_FAK_"   ,"N",  13, 2 } ,;
        { "CENA_NAB_"   ,"N",  13, 2 } ,;
        { "CENA_VPR_"   ,"N",  13, 2 } ,;
        { "CENA_MPR_"   ,"N",  13, 2 } ,;
        { "CENA_DEV_"   ,"N",  13, 2 } ,;
        { "ZALI_"       ,"N",  15, 3 } ,;
        { "DAT0_"       ,"D",   8, 0 } ,;
        { "MAGS_"       ,"C",   2, 0 } ,;
        { "ZNAK_"       ,"C",   1, 0 } ,;
        { "FLAG_"       ,"C",   1, 0 } ,;
        { "STAT_"       ,"C",   1, 0 } ,;
        { "CENA_PRO_"   ,"N",  13, 3 } ,;
        { "OSNO_AKC_"   ,"N",  13, 3 } ,;
        { "ROK_"        ,"N",   3, 0 } ,;
        { "DOBS_"       ,"C",   4, 0 } ,;
        { "DOBK_"       ,"C",  25, 0 } ,;
        { "VEZA_"       ,"C",  15, 0 } ,;
        { "NAME_"       ,"C", 200, 0 } ,;
        { "OPIS_"       ,"M",  10, 0 }  ;
      }

      DbCreate( SPISAK_ROBE, aStructure, "DBFNTX" )

      USE
      SELECT(radno)

  RETURN NIL

  * End FUNCTION KROBA_create()



SELECT(radno)
RETURN NIL

  ***************************************************************************
  FUNCTION KROBA_index(SPISAK_ROBE)
  ***************************************************************************
  LOCAL radno:=SELECT()
  USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "KR"
  IF NetErr()
     RETURN NIL
  ENDIF
  INDEX ON ROBN_ TO 1.NTX
  INDEX ON ROBS_ TO 2.NTX
  USE
  SELECT(radno)
  RETURN NIL
```

```
  ***************************************************************************
  FUNCTION KROBA_DBF_data_from_text_file(SPISAK_ROBE)
  ***************************************************************************
  LOCAL radno := SELECT()
  IF C__NEDA("RECEIVING DATA FROM THE APPLICATION IN "+SPISAK_ROBE,"RECEIVING DATA",,"G3")=.F.
     RETURN NIL
  ENDIF

  * (SPISAK_ROBE) = KROBA_11.DBF/KROBA_11.DBT
  USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "KR"
  IF NetErr()
     s_net(SPISAK_ROBE)
     RETURN NIL
  ENDIF
  br:=reccount()
  //stop("1. DBF ROWS = "+var2char(br),1)
  ZAP  // prazni se DBF/DBT  |  empty DBF/DBT
  COMMIT
  br:=reccount()
  //stop("2. ZAP DBF ROWS = "+var2char(br),1)
  USE

  USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "KR"
  IF NetErr()
     s_net(SPISAK_ROBE)
     RETURN NIL
  ENDIF
  **************  //---------------------------------
  KROBA___fill()  // KROBA_DBF_data_in_text_file.PRG
  **************  //---------------------------------
  br:=reccount()
  // stop("3. FILL DBF ROWS = "+var2char(br),1)
  INDEX ON ROBN_ TO 1.NTX
  INDEX ON ROBS_ TO 2.NTX
  USE

  SELECT(radno)
  C__PORUKA("DATA HAS BEEN TAKEN","OK",,"G3")
  RETURN NIL
```

```
*############################ SERVICE FUNCTIONS ############################  START
/*
Funkcije:
 FUNCTION KROBA_DBF_data_in_text_file(SPISAK_ROBE)
 FUNCTION KROBA_variables(nIzbor)
 FUNCTION KROBA_variables_from_fields()
 FUNCTION KROBA_variables_to_fields()
ne koriste se u ovoj aplikaciji.

Ove funkcije služile su da se iz dbf/dbt fajlova kroba_11.dbf i kroba_11.dbt
napravi izvorni kod u fajlu: KROBA_DBF_data_in_text_file.PRG
Ovaj izvorni kod sadrži 196 artikala - rekorda - sa podacima o artiklima.
Fajl: KROBA_DBF_data_in_text_file.PRG linkuje se u ovu aplikaciju i u njemu
je funkcija KROBA___fill() koja puni KROBA_11.DBF/DBT sa demo podacima
```

```
---

Functions:
  FUNCTION KROBA_DBF_data_in_text_file(LIST_GOODS)
  FUNCTION KROBA_variables(nChoice)
  FUNCTION KROBA_variables_from_fields()
  FUNCTION KROBA_variables_to_fields()
are not used in this application.

These functions served to extract from the dbf/dbt files kroba_11.dbf and
kroba_11.dbt make the source code in the file: KROBA_DBF_data_in_text_file.PRG
This source code contains 196 articles - records - with article data.
File: KROBA_DBF_data_in_text_file.PRG links to and in this application
is the function KROBA___fill() which fills KROBA_11.DBF/DBT with demo data
*/

  ***************************************************************************
  FUNCTION KROBA_DBF_data_in_text_file(SPISAK_ROBE)
  ***************************************************************************

  LOCAL radno := SELECT()
  LOCAL x, OOPIS_, cr, lq, red, XDAT0_

  SET DATE GERMAN
  SET CENTURY ON
  SET CHARSET TO ANSI

  SET CONSOLE OFF
  SET PRINT ON
  SET PRINTER TO (gde_exe()+"\(CODEBOOK1)\KROBA_DBF_data_in_text_file.PRG")

  ?  " * "+DTOC(DATE())+" - "+TIME()
  ?  " "
  ?  " * --- KROBA_DBF_data_in_text_file.PRG                              "
  ?  " * ---                                                              "
  ?  " * Iz fajla KROBA_11.DBF - KROBA_11.DBT šalju se podaci        "
  ?  " * u tekst fajl KROBA_DBF_data_in_text_file.PRG                     "
  ?  " * Fajl KROBA_DBF_data_in_text_file.PRG koristi se iz ove aplikacije"
  ?  " * za punjenje podacima dbf fajla KROBA_11.DBF/KROBA_11.DBT         "
  ?  " * ---                                                              "
  ?  " * Data is being sent from the file KROBA_11.DBF - KROBA_11.DBT     "
  ?  " * in the text file KROBA_DBF_data_in_text_file.PRG                 "
  ?  " * File KROBA_DBF_data_in_text_file.PRG is used from this application"
  ?  " * for data filling of the dbf file KROBA_11.DBF/KROBA_11.DBT        "


  ? "*******************************************************"
  ? "FUNCTION KROBA___fill()"
  ? "*******************************************************"
  ?  "// --- START ---"
  USE (SPISAK_ROBE) NEW SHARED ALIAS "KR"
  GO TOP
  DO WHILE .NOT. EOF()
  KROBA_variables(1)
  KROBA_variables_from_fields()


    OOPIS_:=""
    cr:=chr(13)+chr(10)
```

```
    tokeninit(@mOPIS_,cr)  // XbToolsIII++
    x:=0
    DO WHILE .T.       // list all rows
      lq := tokenend() // XbToolsIII++
      IF lq==.T.
         EXIT
      ENDIF
      x := x+1
      red := tokennext(@mOPIS_) // XbToolsIII++
      OOPIS_ := OOPIS_ + '"' + alltrim(red) + '"' + " + chr(13)+chr(10) +;" + cr
    ENDDO

  OOPIS_ := OOPIS_ + '" *** "'
  XDAT0_ := 'ctod("' + dtoc(dDAT0_) + '")'

  ?  "// ---"
  ?  "APPEND BLANK"
  ?  "REPLACE   OPIS_        WITH  ;"
  ?  OOPIS_
  ?  "REPLACE   DAT0_        WITH  " + XDAT0_

  ?  "REPLACE   GRUP_        WITH  " + '"' + cGRUP_        + '"'
  ?  "REPLACE   ROBS_        WITH  " + '"' + cROBS_        + '"'
  ?  "REPLACE   ROBN_        WITH  " + '"' + cROBN_        + '"'
  ?  "REPLACE   ROBK_        WITH  " + '"' + cROBK_        + '"'
  ?  "REPLACE   MERA_        WITH  " + '"' + cMERA_        + '"'
  ?  "REPLACE   TARI_        WITH  " + '"' + cTARI_        + '"'
  ?  "REPLACE   MAGS_        WITH  " + '"' + cMAGS_        + '"'
  ?  "REPLACE   ZNAK_        WITH  " + '"' + cZNAK_        + '"'
  ?  "REPLACE   FLAG_        WITH  " + '"' + cFLAG_        + '"'
  ?  "REPLACE   STAT_        WITH  " + '"' + cSTAT_        + '"'
  ?  "REPLACE   DOBS_        WITH  " + '"' + cDOBS_        + '"'
  ?  "REPLACE   DOBK_        WITH  " + '"' + cDOBK_        + '"'
  ?  "REPLACE   VEZA_        WITH  " + '"' + cVEZA_        + '"'
  ?  "REPLACE   NAME_        WITH  " + '"' + cNAME_        + '"'
  ?  "REPLACE   PPUP_        WITH  " + var2char(nPPUP_)
  ?  "REPLACE   PPAP_        WITH  " + var2char(nPPAP_)
  ?  "REPLACE   CENA_FAK_    WITH  " + var2char(nCENA_FAK_)
  ?  "REPLACE   CENA_NAB_    WITH  " + var2char(nCENA_NAB_)
  ?  "REPLACE   CENA_VPR_    WITH  " + var2char(nCENA_VPR_)
  ?  "REPLACE   CENA_MPR_    WITH  " + var2char(nCENA_MPR_)
  ?  "REPLACE   CENA_DEV_    WITH  " + var2char(nCENA_DEV_)
  ?  "REPLACE   ZALI_        WITH  " + var2char(nZALI_)
  ?  "REPLACE   CENA_PRO_    WITH  " + var2char(nCENA_PRO_)
  ?  "REPLACE   OSNO_AKC_    WITH  " + var2char(nOSNO_AKC_)
  ?  "REPLACE   ROK_         WITH  " + var2char(nROK_)
  ?  "// ---"
  KROBA_variables(0)
  SKIP
  ENDDO
  COMMIT
  USE

  ?  "// --- END ---"
  ?  "RETURN NIL"
  SET CONSOLE ON
  SET PRINT OFF
  SET PRINTER TO

 SELECT(radno)
```

```
RETURN NIL



*********************************************************************
FUNCTION KROBA_variables(nIzbor)
*********************************************************************

DEFAULT nIzbor TO 1

    IF nIzbor = 1
    // --- Definicija i inicijalizacija Varijabli DBF polja
    // --- Definition and initialization of DBF field variables
            PUBLIC ;
                    mOPIS_,;
                    dDAT0_,;
                    cGRUP_,;
                    cROBS_,;
                    cROBN_,;
                    cROBK_,;
                    cMERA_,;
                    cTARI_,;
                    cMAGS_,;
                    cZNAK_,;
                    cFLAG_,;
                    cSTAT_,;
                    cDOBS_,;
                    cDOBK_,;
                    cVEZA_,;
                    cNAME_,;
                    nPPUP_,;
                    nPPAP_,;
                    nCENA_FAK_,;
                    nCENA_NAB_,;
                    nCENA_VPR_,;
                    nCENA_MPR_,;
                    nCENA_DEV_,;
                    nZALI_    ,;
                    nCENA_PRO_,;
                    nOSNO_AKC_,;
                    nROK_
    ELSE // nIzbor = 0
    // --- BRISANJE - RELEASE Varijabli DBF polja
    // --- DELETE - RELEASE DBF field variables
            RELEASE ;
                    mOPIS_,;
                    dDAT0_,;
                    cGRUP_,;
                    cROBS_,;
                    cROBN_,;
                    cROBK_,;
                    cMERA_,;
                    cTARI_,;
                    cMAGS_,;
                    cZNAK_,;
                    cFLAG_,;
                    cSTAT_,;
                    cDOBS_,;
                    cDOBK_,;
```

```
                    cVEZA_,;
                    cNAME_,;
                    nPPUP_,;
                    nPPAP_,;
                    nCENA_FAK_,;
                    nCENA_NAB_,;
                    nCENA_VPR_,;
                    nCENA_MPR_,;
                    nCENA_DEV_,;
                    nZALI_    ,;
                    nCENA_PRO_,;
                    nOSNO_AKC_,;
                    nROK_
      ENDIF // nIzbor = 0
RETURN NIL

*********************************************************************
FUNCTION KROBA_variables_from_fields()
*********************************************************************
    //--- MEMO
         mOPIS_          := OPIS_
    //--- DATE
         dDAT0_          := DAT0_
    //--- CARACTER
         cGRUP_          := GRUP_
         cROBS_          := ROBS_
         cROBN_          := ROBN_
         cROBK_          := ROBK_
         cMERA_          := MERA_
         cTARI_          := TARI_
         cMAGS_          := MAGS_
         cZNAK_          := ZNAK_
         cFLAG_          := FLAG_
         cSTAT_          := STAT_
         cDOBS_          := DOBS_
         cDOBK_          := DOBK_
         cVEZA_          := VEZA_
         cNAME_          := NAME_
    //--- NUMERIC
         nPPUP_          := PPUP_
         nPPAP_          := PPAP_
         nCENA_FAK_      := CENA_FAK_
         nCENA_NAB_      := CENA_NAB_
         nCENA_VPR_      := CENA_VPR_
         nCENA_MPR_      := CENA_MPR_
         nCENA_DEV_      := CENA_DEV_
         nZALI_          := ZALI_
         nCENA_PRO_      := CENA_PRO_
         nOSNO_AKC_      := OSNO_AKC_
         nROK_           := ROK_
RETURN NIL
* End FUNCTION

*********************************************************************
FUNCTION KROBA_variables_to_fields()
*********************************************************************
    //---MEMO-punjenje iz VARIJABLI:
         REPLACE OPIS_           WITH mOPIS_
    //---DATE-punjenje iz VARIJABLI:
         REPLACE DAT0_           WITH dDAT0_
```

```
      //---CARACTER-punjenje iz VARIJABLI:
           REPLACE GRUP_            WITH cGRUP_
           REPLACE ROBS_            WITH cROBS_
           REPLACE ROBN_            WITH cROBN_
           REPLACE ROBK_            WITH cROBK_
           REPLACE MERA_            WITH cMERA_
           REPLACE TARI_            WITH cTARI_
           REPLACE MAGS_            WITH cMAGS_
           REPLACE ZNAK_            WITH cZNAK_
           REPLACE FLAG_            WITH cFLAG_
           REPLACE STAT_            WITH cSTAT_
           REPLACE DOBS_            WITH cDOBS_
           REPLACE DOBK_            WITH cDOBK_
           REPLACE VEZA_            WITH cVEZA_
           REPLACE NAME_            WITH cNAME_
      //---NUMERIC-punjenje iz  VARIJABLI:
           REPLACE PPUP_            WITH nPPUP_
           REPLACE PPAP_            WITH nPPAP_
           REPLACE CENA_FAK_        WITH nCENA_FAK_
           REPLACE CENA_NAB_        WITH nCENA_NAB_
           REPLACE CENA_VPR_        WITH nCENA_VPR_
           REPLACE CENA_MPR_        WITH nCENA_MPR_
           REPLACE CENA_DEV_        WITH nCENA_DEV_
           REPLACE ZALI_            WITH nZALI_
           REPLACE CENA_PRO_        WITH nCENA_PRO_
           REPLACE OSNO_AKC_        WITH nOSNO_AKC_
           REPLACE ROK_             WITH nROK_
RETURN NIL
  * End FUNCTION

*############################ SERVICE FUNCTIONS ##############################  END
```

__ARTICLES_CODEBOOK.PRG

```
//////////////////////////////////////////////////////////////////////
//                                                                  //
//                                                                  //
//   __ARTICLES_CODEBOOK.PRG                          DBF-DBT-NTX   //
//                                                                  //
//   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
-------------------------------

 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)

   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:=" DBF-DBT-NTX FILE "

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  A N D  I N D E X  N A M E S
//=========================================================================
  * TABLE NAMES AND INDEX NAMES
  *
  * * DBFNTX aplication tables and indexes:

     PRIVATE SPISAK_ROBE, SPISAK_ROBE1, SPISAK_ROBE2
     SPISAK_ROBE  :="kroba_11.DBF"    // there is also a file kroba_11.DBT
     SPISAK_ROBE1 :="1.NTX"           // INDEX ON ROBN_ TO 1.NTX
     SPISAK_ROBE2 :="2.NTX"           // INDEX ON ROBS_ TO 2.NTX




    PRIVATE MAG := _MAG_
    PRIVATE KAS := _KAS_
```

```
  // za program: __ARTIKAL_selekcija_spisak.prg
  // proveriti da li ovo treba !!!
    *-----------------------------------------------------------------
    PRIVATE S_DBF := SPISAK_ROBE
    PRIVATE ;
    NAZIV1_ := "" ,;
    MAG := right(strtran(upper(SPISAK_ROBE),".DBF"),2),;  // kroba_11.dbf -> "11"
    xMAG := MAG,;                  // ŠIFRA EXTERNOG MAGACINA
    naziv_objekta := xMAG+" "+NAZIV1_,; // NAZIV OBJEKTA
    OTIP := "R0",;                 // TIP OBJEKTA - MALOPRODAJA
    ZNAK := "4"                    // TIP OBJEKTA MALOPRODAJA __ARTIKAL_selekcija_spisak.prg

    *** stop(KAS,MAG,OTIP,ZNAK,naziv_objekta,1)
    *-----------------------------------------------------------------
  // za program: __ARTIKAL_selekcija_spisak.prg
  // proveriti da li ovo treba !!!



//=========================================================================
// T A B L E  N A M E S  A N D  I N D E X  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  D B F  F I L E S  (T A B L E)
//=========================================================================

* --- STEP 1 ---

  ************************************************************
    USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "ART"
    IF NetErr()
       s_net()
       RETURN NIL
    ENDIF
  * or
  * DBUSEAREA(.T.,_oSession_,(SPISAK_ROBE),"ART",.F.,.F.)
  * USE (cTablename) NEW EXCLUSIVE ALIAS "ART" VIA (oSession)
  ************************************************************

  SELECT "ART" // OBAVEZNO | REQUIRED

* --- STEP 2 --- FILTER PO GRUPI  --- ISAM
*               FILTERS BY GROUP --- ISAM
  IF EMPTY(GRUPA)
     *" NO FILTER
     svega   := reccount()
  ELSE
     *" YES FILTER
      SET FILTER TO ALLTRIM(DOBS_)=ALLTRIM(GRUPA)
      GO TOP
      count all to svega
      GO TOP
  ENDIF // IF EMPTY(GRUPA)

* --- STEP 3 --- CREATE INDEXES --- ISAM
```

```
 INDEX ON ROBN_ TO 1
 INDEX ON ROBS_ TO 2
 SET INDEX TO 1,2

* --- STEP 4 ---

 SELECT "ART"
 GO TOP


 * Note: Ovo ispravno radi sa DBF-DBT-NTX fajlovima
 * i sa ISAM indeksima i sa ISAM komandama i funkcijama.
 * ---
 * Note: This works correctly with a DBF-DBT-NTX files
 * both with ISAM indexes and with ISAM commands and functions.
 * ---
 * Videti funkcije | See functions:
 * find_name(oBrowse)
 * find_code()
 * find_word(prozor)
 * find_barcod()
 * find_cataloque()
 * find_supplier(prozor)
 * find_link(prozor)


//=========================================================================
// D C B R O W S E  C O N T E N T  F R O M  D B F  F I L E S  (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
@  0,42 DCSAY cTit SAYFONT "10.Verdana Bold"  SAYCOLOR GRA_CLR_WHITE,GRA_CLR_DARKGREEN 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 katalоškom 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(oBrowse,1),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

  IF EMPTY(GRUPA)
    svega:=reccount()
  ELSE
    rek:=recno()
    COUNT ALL TO svega  // reccount() not works
    goto rek
  ENDIF

  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 katalоškom 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+;
"Artikal se na Listu dodaje sa [INSERT]"     +cr+;
"Artikal se sa Liste obriše sa [DELETE]"     +cr+;
```

```
"Rekonstrukcija Liste Artikala [ALT]+[DEL]"    +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+;
"Article is added to the List with [INSERT]"   +cr+;
"Article is deleted from List with [DELETE]"   +cr+;
"Reconstruction List of Articles [ALT]+[DEL]"  +cr+;
" "                                            +cr+;
""
* alertbox(,ctxt,{"    OK    "},,cTitle,"G1") // bazne.dll
c__procitaj(ctxt,"ARTICLES CODE BOOK",,"G1")  // 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")


   SET ORDER TO 1
   SET SOFTSEEK ON
   GO TOP
   SEEK cUpis

/*
* OPTION:
   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)


   SET ORDER TO 2
   SET SOFTSEEK ON
   GO TOP
   SEEK cUpis
```

```
/*
* OPTION:
   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)

   SET ORDER TO 1
   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)

   SET ORDER TO 1
   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,SPISAK_ROBE,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,SPISAK_ROBE,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,SPISAK_ROBE,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)

*********************************************************************
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
PACK

   * --- 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)


   SET ORDER TO 2
   * SET SOFTSEEK ON
   * GO TOP
   SEEK cUpis
   IF FOUND()
      c__poruka("ARTICLE EXISTS","OK",,"G3")
      CLEAR TYPEAHEAD
      RETURN NIL
   ENDIF

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

   *"APPEND BLANK possible with USE SHARED
   *"APPEND BLANK possible with USE EXCLUSIVE
   APPEND BLANK
   REPLACE ROBS_ WITH cUpis
   REPLACE ROBN_ WITH " *** NEW ARTICLE ***"
   REPLACE DOBS_ WITH xGRUPA
   COMMIT
   DC_GetRefresh(GetList)
   oBrowse:RefreshAll()
   setappfocus(oBrowse)

SELECT(radno)
RETURN NIL


*************************************************************************
STATIC FUNCTION RECONSTRUCTION(oBrowse)
*************************************************************************
LOCAL radno:=SELECT()

IF C__NEDA("REKONSTRUKCIJA SPISKA ARTIKALA"  +chr(59)+;
           "RECONSTRUCTION LIST OF ARTICLES" +chr(59)+;
           "","YES-NO",,"G3")=.F.
```

```
    RETURN NIL
ENDIF

* ---
  USE
  ************************
  KROBA_index(SPISAK_ROBE)   // __ARTICLES_MAIN.PRG
  ************************
  USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "ART"
  IF NetErr()
     s_net()
     QUIT_QUIT()
  ENDIF
  SELECT "ART" // OBAVEZNO | REQUIRED
  svega := reccount()
  GO TOP
  IF EMPTY(GRUPA)
     *" NO FILTER
  ELSE
     *" YES FILTER
      SET FILTER TO ALLTRIM(DOBS_)=ALLTRIM(GRUPA)
      GO TOP
  ENDIF // IF EMPTY(GRUPA)
  SET INDEX TO 1,2
  oBrowse:Refreshall()
* ---

C__PORUKA("SPISAK ARTIKALA REKONSTRUISAN"  +chr(59)+;
          "ARTICLES LIST RECONSTRUCTED  "  +chr(59)+;
           "","OK",,"G3")

SELECT(radno)
RETURN NIL
```

__ARTICLES_DESCRIPTION.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES_DESCRIPTION.PRG                          DBF-DBT-NTX  //
//                                                                    //
//   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"

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) )


    ************************************************************
    SELECT(radno)

    * IF DBRLOCK() // TABLE IS USE EXCLUSIVE

       REPLACE OPIS_ WITH cTxt
       // for table is use shared
       // error: * file is opened in read-only mode
    * DBRUNLOCK()
    * ENDIF

    DBREFRESH()
    oBrowse:refreshall() // ova komanda osvežava browser sa novim podatkom
                         // this command refreshes the browser with new data
    ************************************************************

 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                        DBF-DBT-NTX   //
//                                                                    //
//   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()

 OPCIJA:="DBF"
 * OPCIJA:="SQL"


 Artikli_Maticni_Podaci_Public()
 Artikli_Maticni_Podaci_Citaj()

 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:=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)
         dc_getrefresh(GetList)

         Artikli_Maticni_Podaci_Pisi(oBrowse)    // WRITE DATA

         Artikli_Maticni_Podaci_Release()

         * SELECT "ART"
         SELECT(radno)
         COMMIT
         dc_getrefresh(GetList)
         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(oBrowse)
************************************************************
LOCAL radno := SELECT()

IF OPCIJA="DBF"

 REPLACE ROBS_ WITH ROBS

 REPLACE DAT0_ WITH DAT0
 REPLACE GRUP_ WITH GRUP
 REPLACE ROBN_ WITH ROBN
 REPLACE ROBK_ WITH ROBK
 REPLACE DOBK_ WITH DOBK
 REPLACE VEZA_ WITH VEZA
 REPLACE DOBS_ WITH DOBS
 REPLACE MERA_ WITH MERA
 REPLACE TARI_ WITH TARI
 REPLACE PPUP_ WITH PPUP
 REPLACE PPAP_ WITH PPAP
 REPLACE OSNO_AKC_ WITH OSNO_AKC
 REPLACE ROK_  WITH ROK

 REPLACE CENA_FAK_ WITH CENA_FAK
 REPLACE CENA_NAB_ WITH CENA_NAB
 REPLACE CENA_VPR_ WITH CENA_VPR
 REPLACE CENA_MPR_ WITH CENA_MPR

 REPLACE CENA_PRO_ WITH CENA_PRO
 REPLACE CENA_DEV_ WITH CENA_DEV
 REPLACE ZALI_ WITH ZALI

 REPLACE STAT_ WITH STATx
 REPLACE FLAG_ WITH FLAG
 REPLACE ZNAK_ WITH ZNAK
 REPLACE NAME_ WITH NAME
 REPLACE OPIS_ WITH OPIS

ENDIF



IF OPCIJA="SQL"

***
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
cDAT0 := DTOC(DAT0)

TEXT INTO cSQL WRAP
```

```
UPDATE &ztable
SET
DAT0_ = '&cDAT0',
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_ = '&ROBS'
ENDTEXT
* dc_memoedit(cSQL,1) // test

     oStmt := DacSqlStatement():fromChar(cSQL)
     oStmt:build():execute()
     ********************
     * SELECT "ART"
     SELECT(radno)
     ********************
     SetAppFocus(oBrowse)
     DBREFRESH()
     oBrowse:refreshall()
     dc_getrefresh(GetList)
***
ENDIF
RETURN(0)



************************************************************
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(0)
*************************************************************
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(0)
*************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Citaj()
*************************************************************
IF OPCIJA="DBF"

 ROBS := ROBS_
 DAT0 := DAT0_
 GRUP := GRUP_
 ROBN := ROBN_
```

```
 ROBK := ROBK_
 DOBK := DOBK_
 VEZA := VEZA_
 DOBS := DOBS_
 MERA := MERA_
 TARI := TARI_
 PPUP := PPUP_
 PPAP := PPAP_
 OSNO_AKC := OSNO_AKC_
 CENA_PRO := CENA_PRO_
 CENA_DEV := CENA_DEV_
 ZALI := ZALI_
 ROK  := ROK_
 CENA_FAK := CENA_FAK_
 CENA_NAB := CENA_NAB_
 CENA_VPR := CENA_VPR_
 CENA_MPR := CENA_MPR_
 STATx    := STAT_
 FLAG     := FLAG_
 ZNAK     := ZNAK_
 NAME     := NAME_
 OPIS     := OPIS_

ENDIF

IF OPCIJA="SQL"

 ROBS := ROBS_
 DAT0 := DTOC(DAT0_)
 GRUP := GRUP_
 ROBN := ROBN_
 ROBK := ROBK_
 DOBK := DOBK_
 VEZA := VEZA_
 DOBS := DOBS_
 MERA := MERA_
 TARI := TARI_
 PPUP := var2char(PPUP_)
 PPAP := var2char(PPAP_)
 OSNO_AKC := var2char(OSNO_AKC_)
 CENA_PRO := var2char(CENA_PRO_)
 CENA_DEV := var2char(CENA_DEV_)
 ZALI := var2char(ZALI_)
 ROK  := var2char(ROK_)
 CENA_FAK := var2char(CENA_FAK_)
 CENA_NAB := var2char(CENA_NAB_)
 CENA_VPR := var2char(CENA_VPR_)
 CENA_MPR := var2char(CENA_MPR_)
 STATx    := STAT_
 FLAG     := FLAG_
 ZNAK     := ZNAK_
 NAME     := NAME_
 OPIS     := OPIS_

ENDIF

RETURN(0)


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

```
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                          DBF-DBT-NTX  //
//                                                                  //
//   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: "ARTICLES.DBF" filename - if it exists that table is used
* 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


VARIJANTA:="A"

********************************************************************* START
IF VARIJANTA="A"

// VARIJANTA A:
// AKO JE | IF USE
//   USE (SPISAK_ROBE) NEW SHARED
//   USE (SPISAK_ROBE) NEW EXCLUSIVE
// Formiraj temporary fajl kao LISTU selekcije:
// Create a temporary file as a selection LIST:

   ec_p("Selekcija...")
   fajl_selekcija     := Gde_Tmp("1")  // TMP FILE // bazne.dll
   fajl_selekcija_ntx := Gde_Tmp("2")  // TMP FILE // bazne.dll

   IF vrsta = "ROBN_"
      COPY ALL TO (fajl_selekcija) FOR AT(UPPER(txt),UPPER(ROBN_))<>0
   ENDIF
   IF vrsta = "DOBS_"
      COPY ALL TO (fajl_selekcija) FOR AT(UPPER(txt),UPPER(DOBS_))<>0
   ENDIF
   IF vrsta = "VEZA_"
      COPY ALL TO (fajl_selekcija) FOR AT(UPPER(txt),UPPER(VEZA_))<>0
   ENDIF

   USE (fajl_selekcija) NEW EXCLUSIVE ALIAS LISTA    // open LIST
   INDEX ON ROBN_ TO (fajl_selekcija_ntx)
   ec_k("Selekcija...")

ENDIF // IF VARIJANTA="A"
********************************************************************* END


********************************************************************* START
IF VARIJANTA="B"
// VARIJANTA B:
// SAMO AKO JE: USE (SPISAK_ROBE) NEW SHARED
//---------------------------------------
// Formiraj drugu instancu spiska artikala SPISAK_ROBE
// ili prvu instancu neke druge DBF
// i filtriraj po zadatoj selekciji da dobijes LISTU
// koja sadrži samo artikle iz selekcije

// ONLY IF: USE (SPISAK_ROBE) NEW SHARED
//---------------------------------------
// Create a second instance of the articles list SPISAK_ROBE
// or the first instance of some other DBF
// and filter by the given selection to get a LIST
// which contains only items from the selection

IF FILE(xxsdbf) = .F.
   C__GRESKA("FILE NOT FOUND "+xxsdbf,"STOP",,"G3")
   RETURN NIL
ENDIF
```

```
* ---
// NAPRAVI NOVU INSTANCU fajla: SPISAK_ROBE
// CREATE A NEW INSTANCE of the file: SPISAK_ROBE
   USE (xxsdbf) NEW SHARED ALIAS LISTA
   IF NetErr()
      SELECT(radno)
      s_net("FILTER SELECTIONS")
      RETURN(nil)
   ENDIF
   GO TOP
* ---

// POSTAVI FILTER  -- SET FILTER
ec_p("Selekcija...")
         IF vrsta = "ROBN_"
            SET FILTER TO IIF(AT(selekcija,ROBN_) = 0,.F.,.T.)
         ENDIF
         IF vrsta = "VEZA_"
            SET FILTER TO IIF(AT(selekcija,VEZA_) = 0,.F.,.T.)
         ENDIF
         IF vrsta = "DOBS_"
            SET FILTER TO IIF(AT(selekcija,DOBS_) = 0,.F.,.T.)
         ENDIF
GO TOP  // aktiviraj filter
ec_k("Selekcija...")

ENDIF // IF VARIJANTA="B"
********************************************************************** END


    ******************************
     PRIVATE sifra_artikla := ""
    // ovde se smesta sifra izabranog artikla iz fajl_selekcija
    // here is the code of the selected article from file_selection
    ******************************


@  0,0 DCSTATIC TYPE XBPSTATIC_TYPE_BITMAP PIXEL SIZE 22,22 CAPTION BMP_DOC PARENT BrowBox //
BAZNE.DLL
@  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 ; // PRESENTATION DC_BrowPres() ;  NOHSCROLL ;
                   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                              // zatvori listu

// Varijanta A:
   DELETE FILE (fajl_selekcija)
   DELETE FILE (fajl_selekcija_ntx)

// varijanta A i B
SELECT(radno) // selektuj na spisak artikala


IF EMPTY(sifra_artikla)=.F.  // samo ako je sifra artikla preuzeta iz liste
   SET ORDER TO 2            // indeks na sifru
   SEEK sifra_artikla        // lociraj artikal na spisku artikala
   SET ORDER TO 1            // indeks na naziv
ELSE
   GOTO rek      // vrati se na aktuelni slog  spiska artikala
ENDIF

/*
* OPTIONS:
IF EMPTY(sifra_artikla)=.F.  // samo ako je sifra artikla preuzeta iz liste
   GO TOP
   LOCATE FOR alltrim(ROBS_) == alltrim(sifra_artikla)
ELSE
   GOTO rek      // vrati se na aktuelni slog  spiska artikala
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)
```

PRG FOR ARTICLES-CODEBOOK2.EXE

Ovde su se javili sledeći problemi
The following problems occurred here

1. Problem sa UPSIZE INDEKSIMA
2. Problem sa UPSIZE FILTEROM
3. Problem sa UPSIZE komandama DELETE, PACK, APPEND BLANK
1. UPSIZE INDEXES are a problem
2. UPSIZE FILTER is a problem
3. The problem is the UPSIZE command, DELETE, PATCH, APPEND BLANK

1. Problem sa UPSIZE INDEKSIMA
1. UPSIZE INDEXES are a problem

CREATE INDEX FOR PGDBE UPSIZE TABLE
Kada indeksi za upsize tablu kroba_11 ne postoje već se potrebni indeksi kreiraju iz ove aplikacije PGDBE to radi na sledeći način sa ISAM komandama:

When indexes for the upsize table kroba_11 do not exist the necessary indexes are already created from this application PGDBE does it as follows with ISAM commands:

```
USE kroba_11 NEW EXCLUSIVE ALIAS "ART"
INDEX ON ROBN_ TO 1
INDEX ON ROBS_ TO 2
```

Posle ovih komandi:
after these commands:

U tablu "kroba_11" dodate su 2 nove kolone :
2 new columns were added to the "kroba_11" table:

```
__order_1_1  sadži|contains: ROBN_@__record
__order_2_2  sadži|contains: ROBS_@__record
```

U tablu "kroba_11" u sekciju: Indexes dodata su 4 nova indeksa:
In the table "kroba_11" in the section: Indexes, 4 new indexes were added:

```
kroba_11__order_1_1_4like
kroba_11__order_1_1_4seek
kroba_11__order_2_2_4like
kroba_11__order_2_2_4seek
```

U tablu "alaska-software.isam.orders" dodata su 2 reda:
2 rows were added to the "alaska-software.isam.orders" table:

```
row1 sadži|contains: __order_1_1,  ROBN_, kroba_11...
row2 sadži|contains: __order_2_2,  ROBS_, kroba_11...
```

Note:
Pogledajte kako su ovi problemi rešeni u : **__ARTICLES_CODEBOOK.PRG -> __ARTICLES_CODEBOOK()**
See how these problems are solved in : __ARTICLES_CODEBOOK.PRG -> __ARTICLES_CODEBOOK()

2. Problem sa UPSIZE FILTEROM
2. UPSIZE FILTER is a problem

PGDBE ENGINE NE PODRŽAVA ISAM KOMANDU **SET FILTER TO** PA SE TA KOMANDA SIMULIRA
PGDBE ENGINE DOES NOT SUPPORT THE **SET FILTER TO** COMMAND ITSELF, SO THAT COMMAND IS SIMULATED
Kako će se SET FILTER TO simulirati putem drugih ISAM komandi zavisi od invencije programera. Ja ovde prikazujem dva pomoćna načina za dobijanje filtera po grupi artikala samo upotrebom ISAM komandi. Oznaka za grupu artikla upisana je u kolonu dobs_, a filter se u ovom primeru traži za grupu artikala dobs_="5"

PGDBE ENGINE DOES NOT SUPPORT THE SET FILTER TO COMMAND ITSELF, SO THAT COMMAND IS SIMULATED
PGDBE ENGINE DOES NOT SUPPORT THE SET FILTER TO COMMAND ITSELF, SO THAT COMMAND IS SIMULATED
How SET FILTER TO will be simulated by other ISAM commands is up to the programmer's imagination. I show here two helpful ways to get filter by group of items just using ISAM command. The tag for the article group is entered in the dobs_ column, and the filter in this example is searched for the article group dobs_="5"

a) Prvi način je da se ISAM komande za dobijanje filtera za frupu "5":
a) The first way is to use the ISAM command to get the filter for "5":

```
USE kroba_11 NEW EXCLUSIVE ALIAS "ART"
SET FILTER TO alltrim(dobs_)=="5"
GO TOP
```

zamene sa ISAM pravljenjem temporary table koja će sadržati samo artikle iz grupe "5"
replacement with ISAM by creating a temporary table that will contain only items from group "5"

```
	USE kroba_11 NEW EXCLUSIVE ALIAS "ART"
	COPY ALL TO tmptable FOR alltrim(dobs_)=="5"
	USE
	USE tmptable NEW EXCLUSIVE ALIAS "ART"
	INDEX ON robn_ TO 1 // index by article name
	INDEX ON robs_ TO 2 // index by article code
	SET INDEX TO 1,2
	WORKING...
```

b) Drugi način je da se ISAM komande za dobijanje filtera za frupu "5":
b) Another way is to use the ISAM command to get the filter for "5":

```
USE kroba_11 NEW EXCLUSIVE ALIAS "ART"
SET FILTER TO alltrim(dobs_)=="5"
GO TOP
```

zamene sa ISAM pravljenjem nezgrapne simulacije filtera za grupu "5"
replace with ISAM making a clunky filter simulation for group "5"

```
	SET DELETED OFF
	DELETE ALL
	RECALL ALL FOR ALLTRIM(dobs_)=ALLTRIM(GRUPA)
	SET DELETED ON
	GO TOP
	INDEX ON robn_ TO 1 // index by article name
	INDEX ON robs_ TO 2 // index by article code
	SET INDEX TO 1,2
	WORKING...
```

NOTE: RENUMERACIJA kolone __record u svim redovima posle DELETE_ARTICLE() NEĆE DA RADI SA OVIM FILTEROM VEĆ SAMO SA CELOM TABLOM.
Pravilo: brisanje reda iz tabele treba raditi samo na tabli koja sadrži sve redove i koja nije filtrirana ni na koji način kako bi se posle brisanja uspešno izvršila RENUMERACIJA kolone __record u svim redovima table (videti funkciju DELETE_ARTICLE())

NOTE: RENUMBERATION column __record in all rows after DELETE_ARTICLE() IT WILL NOT WORK WITH THIS FILTER BUT ONLY WITH THE WHOLE TABLE.

Rule: deleting a row from a table should only be done on the table which contains all rows and is not filtered in any way in order to successfully perform a RENUMBERATION of the __record column after deletion in all rows of the table (see function DELETE_ARTICLE())

Kao što se vidi, zamena ISAM komanda SET FILTER TO se mora izvršiti sa više ili manje uspešnog koda, pa je u tom slučaju najbolje koristiti PGDBE SQL komande koje su mnogo brže i sigurnije a imaju i veće mogućnosti, na primer:
As you can see, the replacement of the ISAM command SET FILTER TO must be performed with more or less successful code, so in that case it is best to use the PGDBE SQL commands, which are much faster and safer and have greater capabilities, for example:

c) Treći način:
c) The third way

```
cSQL := "SELECT * FROM kroba_11 WHERE dobs_ = '5' ORDER BY robn_ ;"
oStmt := DacSqlStatement():fromChar(cSQL)
xAlias  := oStmt:build():query(,"ART")
ART:=xAlias
SELECT "ART"
WORKING WITH ISAM COMMANDS...
```

Međutim, sada se pojavljuje još ozbiljniji problem: kada se tabla kroba_11 koristi kao QUERY odnosno kao alias := ART := QUERY tada se **NE MOŽE** izvršiti ISAM indeksiranje iz aplikacije putem komandi:
However, now an even more serious problem appears: when the kroba_11 board is used as QUERY, i.e. as an alias := ART := QUERY, then ISAM indexing from the application cannot be performed using commands:

```
INDEX ON robn_ TO 1 // index by article name
INDEX ON robs_ TO 2 // index by article code
SET INDEX TO 1,2
```

Ostaje samo opcija da se u potpunoti odustane od upotrebe simulacije ISAM indeksa u Alaska Xbase++ PGDBE postgreSQL aplikacijama. Ali u ovoj aplikaciji to nećemo uraditi pa zadržavamo opciju b).
The only option left is to completely abandon the use of ISAM index simulation in Alaska Xbase++ PGDBE postgreSQL applications. But in this application we will not do that, so we keep option b).

Note:
Pogledajte kako su ovi problemi rešeni u : **__ARTICLES_CODEBOOK.PRG -> __ARTICLES_CODEBOOK()**
See how these problems are solved in : __ARTICLES_CODEBOOK.PRG -> __ARTICLES_CODEBOOK()

3. Problem sa UPSIZE komandama DELETE, PACK, APPEND BLANK
3. The problem is the UPSIZE command, DELETE, PATCH, APPEND BLANK

PGDBE engine ne podržava ISAM komandu PACK. Tamo gde je potrebno iz SQL table odmah ukloniti obrisani red to je moguće učiniti samo SQL komandam:
The PGDBE engine does not support the ISAM PACK command. Where it is necessary to immediately remove the deleted row from the SQL table, it can be done only with the SQL command:

```
cSQL:="DELETE FROM kroba_11 WHERE __deleted = TRUE;"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()

cSQL := "VACUUM FULL kroba_11;"
oStmt := DacSqlStatement():fromChar(cSQL)
oStmt:build():execute()
```

Međutim, ovo brisanje reda iz table sada prouzrokuje drugi problem:
However, this deletion of the row from the board now causes another problem:

SCENARIO KOD IZVRŠENJA DELETE-PACK-APPEND U UPSIZE TABLI

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

Kod upotrebe upsize tabli putem 'ručno' napravljenog upsize mehanizma koji je prikazan i koji se koristi ovde, a ne putem upsize mehanizma koji je napravljen putem izvršenja DBFUPSIZE.EXE programa, javlja se sledeći problem:

- naredba ZAP radi (briše sve redove iz table)
- naredba DELETE radi (označava kolonu __deleted = TRUE)
- naredba PACK ne radi

Dakle, kod brisanja redova iz table treba primenjivati taktiku:

```
USE kroba_11 NEW ; SET DELETED ON ; DELETE ; USE
```

a posle nekog vremena treba očistiti tablu kroba_11 od svih redova koji imaju __deleted=TRUE što je moguće samo SQL komandom:

```
cSQL:="DELETE FROM ONLY kroba_11 WHERE __deleted=TRUE;"
cSQL:="VACUUM FULL kroba_11;"
```

Ove dve SQL komande zamenjuju ISAM komandu PACK

Međutim, ja u svojim DBF aplikacijama imam slučaj gde odmah posle DELETE jedne stavke, moram da izvršim i PACK i da uklonim tu stavku, koja ima __deleted = TRUE, iz databaze. Iz tog razloga odmah koristim SQL:

```
cSQL:="DELETE FROM ONLY kroba_11 WHERE robs_='12345';"
cSQL:="VACUUM FULL kroba_11;"
```

I tu sada nastaje problem sa kolonom __record koja je Public Key.

Problem je sledeći:

1 Ako u tabli postoji 6 redova oni moraju imati u koloni __record sukcesivne brojeve: __record = 1,2,3,4,5,6

2 Posle toga, Komanda APPEND BLANK dodaje novi red čiji __record = 7

3 Ako se u toj tabli obriše red 4, ostane 5 redova, koji više nemaju sukcesivne brojeve: __record = 1,2,3,5,6

4 Posle toga, Komanda APPEND BLANK dodaje novi red čiji __record = 6 ali __record = 6 već postoji u tabeli i pošto je __record kreiran kao Public Key ne može se duplirati, pa postgreSQL prijavi grešku

Moje rešenje:

5 Rešenje je da se, posle brisanja reda, u tabli izvrši RENUMERACIJA __record brojeva tako da budu sukcesivni: __record = 1,2,3,4,5, pa će komanda APPEND BLANK dodati novi red pod brojem __record = 6

6 RENUMERACIJA je problem, jer je __record istovremeno i Public Key pa ga nije moguće duplirati. Zbog toga sam morao da pronađem metod za RENUMERACIJU svih redova u tabli pri čemu se nikada neće vršiti dupliranje postojećih __record brojeva:

```
 __record  1,2,3,5,6       stanje posle DELETE __record = 4
 __record  -1,-2,-3,-5,-6  stanje posle renumeracije - prvi prolaz
 __record  1,2,3,4,5       stanje posle renumeracije - drugi prolaz
```

RENUMERACIJA je ovde realizovana funkcijom: RENUMBERATION(table)

```
7  Međutim, može se desiti nasilni prekid programa posle izvršenja
   renumeracije - prvi prolaz i tada će u tabli ostati stanje:
    __record  -1,-2,-3,-5,-6  stanje posle renumeracije - prvi prolaz
   U tom slučaju se ponovna renumeracija ne može izvesti, pa treba
   izvršiti REKONSTRUKCIJU na sledeći način:
    __record  -1,-2,-3,-5,-6  stanje posle havarije
    __record  -100001,-100002,-100003,-100005,-100006 rekonstrukcija
    __record  1,2,3,4,5       stanje posle rekonstrukcije - drugi prolaz
    REKONSTRUKCIJA je ovde realizovana funkcijom RECONSTRUCTION(table)

  Zaključak:

  8  U mojim Xbase++ postgreSQL UPSIZE aplikacijama koje koriste ISAM
     komande i funkcije, posle brisanja reda iz table mora se odmah
     izvršiti funkcija RENUMBERATION(table). Ako ta funkcija ne uspe
     aplikacija mora da automatski izvrši funkciju RECONSTRUCTION(table).
     I korisniku se mora omogućiti da može na svoj zahtev da izvrši
     funkciju RECONSTRUCTION(table).
     Takođe, ako komanda APPEND BLANK ne uspe, sama aplikacija mora da
     automatski izvrši funkciju RECONSTRUCTION(table).

  9  Mehanizam za RENUMERACIJU i REKONSTRUKCIJU demonstriran je ovde u
     aplikaciji ARTICLES-CODEBOOK.EXE u programu __ARTICLED_CODEBOOK.PRG
     u funkcijama:
                  DELETE_ARTICLE()
                  INSERT_ARTICLE()
```

DELETE-PACK-APPEND EXECUTION CODE SCENARIO IN UPSIZE TABLE

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

When using an upsize table through a 'handmade' upsize mechanism which is shown and used here,
not through the upsize mechanism which is created by executing the DBFUPSIZE.EXE program,
occurs following problem:
       - ZAP command works (deletes all lines from the board)
       - DELETE command works (marks column __deleted = TRUE)
       - the PACK command does not work
Therefore, when deleting rows from the board, the following tactics should be applied:
        USE kroba_11 NEW ; SET DELETED ON ; DELETE ; USE
and after some time you need to clear the krob_11 table of all rows which have __deleted=TRUE
which is only possible with SQL command:
      cSQL:="DELETE FROM ONLY kroba_11 WHERE __deleted=TRUE;"
      cSQL:="VACUUM FULL kroba_11;"
These two SQL commands replace the ISAM PACK command

However, I have a case in my DBF applications where right after DELETE of one item, I need to
do a PACK as well and remove that item, which has __deleted = TRUE, from the database. For this
reason I immediately use SQL:
      cSQL:="DELETE FROM ONLY kroba_11 WHERE robs_='12345';"
      cSQL:="VACUUM FULL kroba_11;"
And now there is a problem with the __record column, which is the Public Key.

 The problem is this:

  1 If there are 6 rows in the table, they must be in the __record column
    successive numbers: __record = 1,2,3,4,5,6
```

```
 2 After that, the command APPEND BLANK adds a new line whose __record = 7

 3 If row 4 is deleted in that table, 5 rows remain, which no longer have any
    successive numbers: __record = 1,2,3,5,6

 4 After that, the command APPEND BLANK adds a new line whose __record = 6
    but __record = 6 already exists in the table and since __record is created
    as a Public Key cannot be duplicated, so postgreSQL reports an error

My solution:

5  The solution is to RENUMBER after deleting the row in the table
   __record of numbers so that they are successive: __record = 1,2,3,4,5,
   so the command APPEND BLANK will add a new row numbered __record = 6

6  RENUMERATION is a problem, because __record is also a Public Key
   so it cannot be duplicated. That's why I had to find a method
   to RENUMBER all the rows in the table where it will never be done
   duplicating existing __record numbers:
    __record 1,2,3,5,6      state after DELETE __record = 4
    __record -1,-2,-3,-5,-6 state after renumeration - first pass
    __record 1,2,3,4,5      state after renumeration - second pass
   RENUMBERATION is implemented here with function RENUMBERATION(table)

7  However, a program can be violently terminated after execution
   renumeration - the first pass and then in the board:
    __record -1,-2,-3,-5,-6 state after accident
   In that case, renumbering cannot be performed, so it should
   perform the RECONSTRUCTION as follows:
    __record -1,-2,-3,-5,-6 state after accident
    __record -100001,-100002,-100003,-100005,-100006 reconstruction
    __record 1,2,3,4,5 state after reconstruction - second pass
   RECONSTRUCTION is implemented here with function RECONSTRUCTION(table)

 Conclusion:

 8  In my Xbase++ postgreSQL UPSIZE applications using ISAM
    commands and functions, after deleting a line from the table,
    it must be done immediately execute function RENUMBERATION(table).
    If that function fails the application must automatically execute
    RECONSTRUCTION(table) function. And the user must be enabled to
    execute his request function RECONSTRUCTION(table).
    Also, if the APPEND BLANK command fails, the application itself must
    automatically execute the RECONSTRUCTION(table) function.

 9  The mechanism for RENUMERATION and RECONSTRUCTION is demonstrated
    here in to:
    ARTICLES-CODEBOOK.EXE application in __ARTICLED_CODEBOOK.PRG program
    in functions:
                 DELETE_ARTICLE()
                 INSERT_ARTICLE()
```

Note:
Pogledajte kako su ovi problemi rešeni u : __ARTICLES_CODEBOOK.PRG -> DELETE_ARTICLE()
Pogledajte kako su ovi problemi rešeni u : __ARTICLES_CODEBOOK.PRG -> INSERT_ARTICLE()
See how these problems are solved in : __ARTICLES_CODEBOOK.PRG -> RENUMBERATION()
See how these problems are solved in : __ARTICLES_CODEBOOK.PRG -> RECONSTRUCTION()

__ARTICLES_MAIN.PRG

```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//                                                                    //
//   __ARTICLES__MAIN.PRG                              UPSIZE 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()


*/

*-----------------------------------
* 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                          UPSIZE 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
-------------------------------

 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)

   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:=" UPSIZE 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  A N D  I N D E X  N A M E S
//========================================================================
  * TABLE NAMES AND INDEX NAMES
  * PGDBE upsize aplication tables and indexes:

     PRIVATE SPISAK_ROBE, SPISAK_ROBE1, SPISAK_ROBE2
     SPISAK_ROBE  :="kroba_11"
     SPISAK_ROBE1 :="1"  // INDEX ON ROBN_ TO 1
     SPISAK_ROBE2 :="2"  // INDEX ON ROBN_ TO 2


   *------------------------------------------------------------- START
   * DOCUMENTATION :: CREATE INDEX FOR PGDBE UPSIZE TABLE
   *-------------------------------------------------------------
   *
   * Kada indeksi za upsize tablu kroba_11 ne postoje
   * već se potrebni indeksi kreiraju iz ove aplikacije
   * PGDBE to radi na sledeći način sa ISAM komandama:
   * ---
   * When indexes for the upsize table kroba_11 do not exist
   * the necessary indexes are already created from this application
   * PGDBE does it as follows with ISAM commands:
```

```
   *
   * USE kroba_11 NEW EXCLUSIVE ALIAS "ART"
   * INDEX ON ROBN_ TO 1
   * INDEX ON ROBS_ TO 2
   * Posle ovih komandi:
   * after these commands:
   *
   * U tablu "kroba_11" dodate su 2 nove kolone :
   * 2 new columns were added to the "kroba_11" table:
   *
   *         __order_1_1  sadži|contains: ROBN_@__record
   *         __order_2_2  sadži|contains: ROBS_@__record
   *
   * U tablu "kroba_11" u sekciju: Indexes dodata su 4 nova indeksa:
   * In the table "kroba_11" in the section: Indexes,
   * 4 new indexes were added:
   *
   *          kroba_11__order_1_1_4like
   *          kroba_11__order_1_1_4seek
   *          kroba_11__order_2_2_4like
   *          kroba_11__order_2_2_4seek
   *
   * U tablu "alaska-software.isam.orders" dodata su 2 reda:
   * 2 rows were added to the "alaska-software.isam.orders" table:
   *
   *          row1 sadži|contains: __order_1_1,  ROBN_, kroba_11...
   *          row2 sadži|contains: __order_2_2,  ROBS_, kroba_11...
   *
   *-----------------------------------------------------------------
   * DOCUMENTATION :: CREATE INDEX FOR PGDBE UPSIZE TABLE
   *----------------------------------------------------------------- END

    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()

    PRIVATE MAG := _MAG_  // __pg__ini() C-PGDB.DLL
    PRIVATE KAS := _KAS_  // __pg__ini() C-PGDB.DLL


//=========================================================================
// T A B L E  N A M E S  A N D  I N D E X  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
//=========================================================================


* --- STEP 1 ---

* Spakuj nazive indeksa u varijable za formiranje SQL stringa
* Pack index names into variables to form SQL strings
```

```
  i14like := cTablename+"__order_1_1_4like" // kroba_11__order_1_1_4like
  i14seek := cTablename+"__order_1_1_4seek" // kroba_11__order_1_1_4seek
  i24like := cTablename+"__order_2_2_4like" // kroba_11__order_2_2_4like
  i24seek := cTablename+"__order_2_2_4seek" // kroba_11__order_2_2_4seek

* --- STEP 2 --- DELETE INDEXES --- PGDBE SQL

* Isprazni službenu Alaska Xbase++ tablu "alaska-software.isam.orders"
* Empty the official Alaska Xbase++ table "alaska-software.isam.orders"

 TEXT INTO cSQL WRAP
  DELETE FROM ONLY "alaska-software.isam.orders";
 ENDTEXT
 oStmt := DacSqlStatement():fromChar(cSQL)
 oStmt:build():execute()

* ako postoje - obriši indekse iz table
* if they exist - delete indexes from table

 TEXT INTO cSQL WRAP
  DROP INDEX IF EXISTS &i14like CASCADE;
  DROP INDEX IF EXISTS &i14seek CASCADE;
  DROP INDEX IF EXISTS &i24like CASCADE;
  DROP INDEX IF EXISTS &i24seek CASCADE;
 ENDTEXT
 oStmt := DacSqlStatement():fromChar(cSQL)
 oStmt:build():execute()


* --- STEP 3 --- 1 OPEN TABLE --- PGDBE ISAM

  * Otvaranje table i formiranje dva indeksa po
  * nazivu artikla i po šifri artikla upotrebom
  * ISAM komandi - identično kao u DBF-NTX aplikaciji
  * ---
  * Opening the table and forming two indexes per
  * by name of article and by code of article by use
  * ISAM command - identical to the DBF-NTX application
  * ---
  ************************************************************
    USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "ART"
    IF NetErr()
       S_NET("TABLE: "+SPISAK_ROBE)
       RETURN NIL
    ENDIF
  * or
  * DBUSEAREA(.T.,_oSession_,(SPISAK_ROBE),"ART",.F.,.F.)
  * USE (cTablename) NEW EXCLUSIVE ALIAS "ART" VIA (oSession)
  ************************************************************
  * Note:
  * If: USE (SPISAK_ROBE) NEW SHARED ALIAS "ARTIKLI"
  *     APPEND BLANK Error -> unnamed prepared statement does not exist
  * If: USE (SPISAK_ROBE) NEW EXCLUSIVE ALIAS "ARTIKLI"
  *     APPEND BLANK -> OK


  SELECT "ART" // OBAVEZNO | REQUIRED

* --- STEP 4 --- SIMULACIJA FILTERA PO GRUPI --- PGDBE ISAM
*               SIMULATION OF FILTERS BY GROUP
```

```
* verzija A
*------------
*  COPY ALL TO tmptable FOR ALLTRIM(DOBS_)=ALLTRIM(GRUPA)
*  USE
*  USE tmptable NEW EXCLUSIVE ALIAS "ART"


* verzija B
*------------
IF EMPTY(GRUPA)
   *" NO FILTER
ELSE
   *" YES FILTER
 IF FLOCK()
    SET DELETED OFF
    DELETE ALL
    RECALL ALL FOR ALLTRIM(DOBS_)=ALLTRIM(GRUPA)
    SET DELETED ON
    UNLOCK
 ENDIF
 GO TOP
ENDIF

 *-------------------------------------------------------------------------
 * --- AFTER DELETING ROW FOLLOWS RESET COLUMN __record ------------ START
 *-------------------------------------------------------------------------
 * NOTE 01: RENUMERACIJA kolone __record u svim redovima posle DELETE_ARTICLE()
 * NEĆE DA RADI SA OVIM FILTEROM VEĆ SAMO SA CELOM TABLOM
 * Pravilo: brisanje reda iz tabele treba raditi samo na tabeli
 * koja sadrži sve redove i koja nije filtrirana ni na koji način
 * kako bi se posle brisanja uspešno izvršio RENUMERACIJU kolone __record
 * u svim redovima tabele (vidi ovde funkciju DELETE_ARTICLE())

 * NOTE 01: RENUMBERATION column __record in all rows after DELETE_ARTICLE()
 * IT WILL NOT WORK WITH THIS FILTER BUT ONLY WITH THE WHOLE TABLE
 * Rule: deleting a row from a table should only be done on the table
 * which contains all rows and is not filtered in any way
 * in order to successfully perform a RENUMBERATION of the __record column
 * after deletion in all rows of the table (see function DELETE_ARTICLE() here)
 *-------------------------------------------------------------------------
 * --- AFTER DELETING ROW FOLLOWS RESET COLUMN __record -------------- END
 *-------------------------------------------------------------------------

* --- STEP 5 --- CREATE INDEXES --- PGDBE ISAM

                       // Formira se u postgreSQL tabli:
                       // It is created in postgreSQL table:

 INDEX ON ROBN_ TO 1   // kroba_11__order_1_1_4like
                       // kroba_11__order_1_1_4seek

 INDEX ON ROBS_ TO 2   // kroba_11__order_2_2_4like
                       // kroba_11__order_2_2_4seek
 SET INDEX TO 1,2

* --- STEP 6 ---
 SELECT "ART"
```

```
 * Posebno važno:
 * Kada se u aplikaciji mešaju PGDBE SQL komande i ISAM komande
 * obavezno se mora zadati: SELECT "ART" pre izvršenja bilo
 * koje ISAM komande
 * ---
 * Especially important:
 * When PGDBE SQL commands and ISAM commands are mixed in the application
 * must be specified: SELECT "ART" before executing either
 * which ISAM commands

   DbRefresh()
 * svega   := reccount()
   count all to svega
   GO TOP


 * Note: Ovo ispravno radi sa postgreSQL tablom
 * i sa ISAM indeksima i sa ISAM komandama i funkcijama.
 * ---
 * Note: This works correctly with a postgreSQL table
 * both with ISAM indexes and with ISAM commands and functions.
 * ---
 * Videti funkcije | See functions:
 * find_name(oBrowse)
 * find_code()
 * find_word(prozor)
 * find_barcod()
 * find_cataloque()
 * find_supplier(prozor)
 * find_link(prozor)

//======================================================================
// 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
@  0,42 DCSAY cTit SAYFONT "10.Verdana Bold"  SAYCOLOR GRA_CLR_WHITE,GRA_CLR_DARKRED 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 katalоškom 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

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+;
"Artikal se na Listu dodaje sa [INSERT]"       +cr+;
"Artikal se sa Liste obriše sa [DELETE]"       +cr+;
"Rekonstrukcija Liste Artikala [ALT]+[DEL]"    +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+;
"Article is added to the List with [INSERT]"   +cr+;
"Article is deleted from List with [DELETE]"   +cr+;
"Reconstruction List of Articles [ALT]+[DEL]"  +cr+;
" "                                            +cr+;
""
* alertbox(,ctxt,{"    OK    "},,cTitle,"G1") // bazne.dll
c__procitaj(ctxt,"ARTICLES CODE BOOK",,"G1")  // 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")


   SET ORDER TO 1
   SET SOFTSEEK ON
   GO TOP
   SEEK cUpis

/*

   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)

   SET ORDER TO 2
   SET SOFTSEEK ON
   GO TOP
   SEEK cUpis

/*
   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)

   SET ORDER TO 1
   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)

   SET ORDER TO 1
   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


*-----------------------------------------
bErrorHandler :=ErrorBlock( {|e| Break(e)} )
BEGIN SEQUENCE
*-----------------------------------------

   SELECT(radno)
   DELETE  // now the __deleted column is set to TRUE
```

```
   COMMIT


   // 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"

    *********************************
    RENUMBERATION(cTablename,oBrowse,0)  // here function and documentation
    *********************************

   * --- 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")

*-----------------------------------------
         // Break() radi sa RECOVER
RECOVER  // izvrši kod dođe do greške
         // vezano za definisani Break(e)
*-----------------------------------------


   ch:=chr(59)
   c__greska(;
   "Došlo je do greške. Rekonstruiši Listu"    +ch+;
   "An error occurred. Reconstruct the List"   +ch+;
   "                                   "       +ch+;
   "           tasters [ALT]+[DEL]"            +ch+;
   "","STOP DELETE",,"G3")


   *********************************
   RECONSTRUCTION(cTablename,oBrowse,1) // here function and documentation
   *********************************
```

```
   // stop(1,1)
   SELECT(radno) // SELECT "ART"
   DbRefresh()
   oBrowse:refreshall()
   GO TOP
   c__poruka("DELETED AND RECONSTRUCTION","OK",,"G3")

*-----------------------------------------
END SEQUENCE
ErrorBlock( bErrorHandler ) // reset old codeblock
*-----------------------------------------

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)


   SET ORDER TO 2
   * SET SOFTSEEK ON
   * GO TOP
   SEEK cUpis
   IF FOUND()
      c__poruka("ARTICLE EXISTS","OK",,"G3")
      CLEAR TYPEAHEAD
      RETURN NIL
   ENDIF

/*
   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()
* PRAVILO-RULE
* BEGIN SEQUENCE
*   no return - must not have a RETURN command
* RECOWER
*   yes return - may have a RETURN command
* END SEQUENCE
* RETURN
*-----------------------------------------
bErrorHandler :=ErrorBlock( {|e| Break(e)} )
BEGIN SEQUENCE
*-----------------------------------------
   * za testiranje greške kod APPEND BLANK komentariši
   *    RENUMBERATION(cTablename,oBrowse,0) I
   *    RECONSTRUCTION(cTablename,oBrowse,1)
   * u funkciji DELETE_ARTICLE()
   * APPEND BLANK only possible with USE EXCLUSIVE !
   APPEND BLANK
   REPLACE ROBS_ WITH cUpis
   REPLACE ROBN_ WITH " *** NEW ARTICLE ***"
   REPLACE DOBS_ WITH xGRUPA
   COMMIT
   DC_GetRefresh(GetList)
   oBrowse:RefreshAll()
*-----------------------------------------
         // Break() radi sa RECOVER
RECOVER  // izvrši kad dođe do greške
         // vezano za definisani Break(e)
*-----------------------------------------
   ch:=chr(59)
   c__greska(;
   "Došlo je do greške. Rekonstruiši Listu"    +ch+;
   "An error occurred. Reconstruct the List"   +ch+;
   "                                   "       +ch+;
   "           tasters [ALT]+[DEL]"            +ch+;
   "","STOP APPEND BLANK",,"G3")
   **********************************   // STOP
   * RECONSTRUCTION(cTablename,oBrowse,1) // here // here function and documentation
   **********************************   // PUKNE ???
   SELECT(radno) // SELECT "ART"
   DbRefresh()
   oBrowse:refreshall()
   setappfocus(oBrowse)
   GO TOP
   c__poruka("NO ARTICLE ADDED - RECONSTRUCTION","OK",,"G3")
*-----------------------------------------
END SEQUENCE
ErrorBlock( bErrorHandler ) // reset old codeblock
*-----------------------------------------
SELECT(radno)
RETURN NIL
```

```
------------------------------------------------------------------------------ START
 SCENARIO KOD IZVRŠENJA DELETE-PACK-APPEND U UPSIZE TABLI
  FUNCTION RENUMBERATION()
  FUNCTION RECONSTRUCTION()
  FUNCTION RENUMBER__record()
------------------------------------------------------------------------------
 Kod upotrebe upsize tabli putem 'ručno' napravljenog upsize mehanizma
 koji je prikazan i koji se koristi ovde, a ne putem upsize mehanizma
 koji je napravljen putem izvršenja DBFUPSIZE.EXE programa, javlja se
 sledeći problem:
      - naredba ZAP radi (briše sve redove iz table)
      - naredba DELETE radi (označava kolonu __deleted = TRUE)
      - naredba PACK ne radi
 Dakle, kod brisanja redova iz table treba primenjivati taktiku:
 USE kroba_11 NEW ; SET DELETED ON ; DELETE ; USE
 a posle nekog vremena treba očistiti tablu kroba_11 od svih redova
 koji imaju __deleted=TRUE što je moguće samo SQL komandom:
  cSQL:="DELETE FROM ONLY kroba_11 WHERE __deleted=TRUE;"
  cSQL:="VACUUM FULL kroba_11;"
 Ove dve SQL komande zamenjuju ISAM komandu PACK

 Međutim, ja u svojim DBF aplikacijama imam slučaj gde odmah posle DELETE
 jedne stavke, moram da izvršim i PACK i da uklonim tu stavku, koja ima
 __deleted = TRUE, iz databaze. Iz tog razloga odmah koristim SQL:
  cSQL:="DELETE FROM ONLY kroba_11 WHERE robs_='12345';"
  cSQL:="VACUUM FULL kroba_11;"
 I tu sada nastaje problem sa kolonom __record koja je Public Key.

 Problem je sledeći:

 1  Ako u tabli postoji 6 redova oni moraju imati u koloni __record
    sukcesivne brojeve: __record = 1,2,3,4,5,6

 2  Posle toga, Komanda APPEND BLANK dodaje novi red čiji __record = 7

 3  Ako se u toj tabli obriše red 4, ostane 5 redova, koji više nemaju
    sukcesivne brojeve: __record = 1,2,3,5,6

 4  Posle toga, Komanda APPEND BLANK dodaje novi red čiji __record = 6
    ali __record = 6 već postoji u tabeli i pošto je __record kreiran
    kao Public Key ne može se duplirati, pa postgreSQL prijavi grešku

 Moje rešenje:

 5  Rešenje je da se, posle brisanja reda, u tabli izvrši RENUMERACIJA
    __record brojeva tako da budu sukcesivni: __record = 1,2,3,4,5,
    pa će komanda APPEND BLANK dodati novi red pod brojem __record = 6

 6  RENUMERACIJA je problem, jer je __record istovremeno i Public Key
    pa ga nije moguće duplirati. Zbog toga sam morao da pronađem metod
    za RENUMERACIJU svih redova u tabli pri čemu se nikada neće vršiti
    dupliranje postojećih __record brojeva:
     __record  1,2,3,5,6       stanje posle DELETE __record = 4
     __record  -1,-2,-3,-5,-6  stanje posle renumeracije - prvi prolaz
     __record  1,2,3,4,5       stanje posle renumeracije - drugi prolaz
    RENUMERACIJA je ovde realizovana funkcijom: RENUMBERATION(table)

 7  Međutim, može se desiti nasilni prekid programa posle izvršenja
    renumeracije - prvi prolaz i tada će u tabli ostati stanje:
```

```
     __record  -1,-2,-3,-5,-6  stanje posle renumeracije - prvi prolaz
    U tom slučaju se ponovna renumeracija ne može izvesti, pa treba
    izvršiti REKONSTRUKCIJU na sledeći način:
     __record  -1,-2,-3,-5,-6  stanje posle havarije
     __record  -100001,-100002,-100003,-100005,-100006 rekonstrukcija
     __record  1,2,3,4,5       stanje posle rekonstrukcije - drugi prolaz
     REKONSTRUKCIJA je ovde realizovana funkcijom RECONSTRUCTION(table)

  Zaključak:

  8  U mojim Xbase++ postgreSQL UPSIZE aplikacijama koje koriste ISAM
     komande i funkcije, posle brisanja reda iz table mora se odmah
     izvršiti funkcija RENUMBERATION(table). Ako ta funkcija ne uspe
     aplikacija mora da automatski izvrši funkciju RECONSTRUCTION(table).
     I korisniku se mora omogućiti da može na svoj zahtev da izvrši
     funkciju RECONSTRUCTION(table).
     Takođe, ako komanda APPEND BLANK ne uspe, sama aplikacija mora da
     automatski izvrši funkciju RECONSTRUCTION(table).

  9  Mehanizam za RENUMERACIJU i REKONSTRUKCIJU demonstriran je ovde u
     aplikaciji ARTICLES-CODEBOOK.EXE u programu __ARTICLED_CODEBOOK.PRG
     u funkcijama:
                   DELETE_ARTICLE()
                   INSERT_ARTICLE()


-------------------------------------------------------------------------
 SCENARIO KOD IZVRŠENJA DELETE-PACK-APPEND U UPSIZE TABLI
------------------------------------------------------------------------- END

------------------------------------------------------------------------- START
 DELETE-PACK-APPEND EXECUTION CODE SCENARIO IN UPSIZE TABLE
  FUNCTION RENUMBERATION()
  FUNCTION RECONSTRUCTION()
  FUNCTION RENUMBER__record()
-------------------------------------------------------------------------

  When using an upsize table through a 'handmade' upsize mechanism
  which is shown and used here, not through the upsize mechanism
  which is created by executing the DBFUPSIZE.EXE program, occurs
  following problem:
       - ZAP command works (deletes all lines from the board)
       - DELETE command works (marks column __deleted = TRUE)
       - the PACK command does not work
  Therefore, when deleting rows from the board, the following tactics
  should be applied:
  USE kroba_11 NEW ; SET DELETED ON ; DELETE ; USE
  and after some time you need to clear the krob_11 table of all rows
  which have __deleted=TRUE which is only possible with SQL command:
   cSQL:="DELETE FROM ONLY kroba_11 WHERE __deleted=TRUE;"
   cSQL:="VACUUM FULL kroba_11;"
  These two SQL commands replace the ISAM PACK command

  However, I have a case in my DBF applications where right after DELETE
  of one item, I need to do a PACK as well and remove that item, which has
  __deleted = TRUE, from the database. For this reason I immediately use SQL:
   cSQL:="DELETE FROM ONLY kroba_11 WHERE robs_='12345';"
   cSQL:="VACUUM FULL kroba_11;"
  And now there is a problem with the __record column, which is the Public Key.
```

The problem is this:

1 If there are 6 rows in the table, they must be in the __record column
successive numbers: __record = 1,2,3,4,5,6

2 After that, the command APPEND BLANK adds a new line whose __record = 7

3 If row 4 is deleted in that table, 5 rows remain, which no longer have any
successive numbers: __record = 1,2,3,5,6

4 After that, the command APPEND BLANK adds a new line whose __record = 6
but __record = 6 already exists in the table and since __record is created
as a Public Key cannot be duplicated, so postgreSQL reports an error

My solution:

5 The solution is to RENUMBER after deleting the row in the table
__record of numbers so that they are successive: __record = 1,2,3,4,5,
so the command APPEND BLANK will add a new row numbered __record = 6

6 RENUMERATION is a problem, because __record is also a Public Key
so it cannot be duplicated. That's why I had to find a method
to RENUMBER all the rows in the table where it will never be done
duplicating existing __record numbers:
__record 1,2,3,5,6 state after DELETE __record = 4
__record -1,-2,-3,-5,-6 state after renumeration - first pass
__record 1,2,3,4,5 state after renumeration - second pass
RENUMBERATION is implemented here with function RENUMBERATION(table)

7 However, a program can be violently terminated after execution
renumeration - the first pass and then in the board:
__record -1,-2,-3,-5,-6 state after accident
In that case, renumbering cannot be performed, so it should
perform the RECONSTRUCTION as follows:
__record -1,-2,-3,-5,-6 state after accident
__record -100001,-100002,-100003,-100005,-100006 reconstruction
__record 1,2,3,4,5 state after reconstruction - second pass
RECONSTRUCTION is implemented here with function RECONSTRUCTION(table)

Conclusion:

8 In my Xbase++ postgreSQL UPSIZE applications using ISAM
commands and functions, after deleting a line from the table,
it must be done immediately execute function RENUMBERATION(table).
If that function fails the application must automatically execute
RECONSTRUCTION(table) function. And the user must be enabled to
execute his request function RECONSTRUCTION(table).
Also, if the APPEND BLANK command fails, the application itself must
automatically execute the RECONSTRUCTION(table) function.

9 The mechanism for RENUMERATION and RECONSTRUCTION is demonstrated
here in to:
ARTICLES-CODEBOOK.EXE application in __ARTICLED_CODEBOOK.PRG program
in functions:
DELETE_ARTICLE()
INSERT_ARTICLE()

```
------------------------------------------------------------------------
 DELETE-PACK-APPEND EXECUTION CODE SCENARIO IN UPSIZE TABLE
------------------------------------------------------------------------ END
*/


* 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                       UPSIZE 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"

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) )

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

```
    SELECT(radno)

    * IF DBRLOCK() // TABLE IS USE EXCLUSIVE

       REPLACE OPIS_ WITH cTxt
       // for table is use shared
       // error: * file is opened in read-only mode
    * DBRUNLOCK()
    * ENDIF

    DBREFRESH()
    oBrowse:refreshall() // ova komanda osvežava browser sa novim podatkom
                         // this command refreshes the browser with new data
    ********************************************************

 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                      UPSIZE 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()

Artikli_Maticni_Podaci_Public()
 Artikli_Maticni_Podaci_Citaj()

 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:=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)
         dc_getrefresh(GetList)

         Artikli_Maticni_Podaci_Pisi(oBrowse)    // WRITE DATA

         Artikli_Maticni_Podaci_Release()

         * SELECT "ART"
         SELECT(radno)
         COMMIT
         dc_getrefresh(GetList)
         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(oBrowse)
*************************************************************
LOCAL radno := SELECT()

 REPLACE ROBS_ WITH ROBS
 REPLACE DAT0_ WITH DAT0
 REPLACE GRUP_ WITH GRUP
 REPLACE ROBN_ WITH ROBN
 REPLACE ROBK_ WITH ROBK
 REPLACE DOBK_ WITH DOBK
```

```
 REPLACE VEZA_ WITH VEZA
 REPLACE DOBS_ WITH DOBS
 REPLACE MERA_ WITH MERA
 REPLACE TARI_ WITH TARI
 REPLACE PPUP_ WITH PPUP
 REPLACE PPAP_ WITH PPAP
 REPLACE OSNO_AKC_ WITH OSNO_AKC
 REPLACE ROK_  WITH ROK

 REPLACE CENA_FAK_ WITH CENA_FAK
 REPLACE CENA_NAB_ WITH CENA_NAB
 REPLACE CENA_VPR_ WITH CENA_VPR
 REPLACE CENA_MPR_ WITH CENA_MPR

 REPLACE CENA_PRO_ WITH CENA_PRO
 REPLACE CENA_DEV_ WITH CENA_DEV
 REPLACE ZALI_ WITH ZALI

 REPLACE STAT_ WITH STATx
 REPLACE FLAG_ WITH FLAG
 REPLACE ZNAK_ WITH ZNAK
 REPLACE NAME_ WITH NAME
 REPLACE OPIS_ WITH OPIS

RETURN(0)

*************************************************************
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(0)

*************************************************************
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(0)

************************************************************
STATIC FUNCTION Artikli_Maticni_Podaci_Citaj()
************************************************************
 ROBS := ROBS_
 DAT0 := DAT0_
 GRUP := GRUP_
 ROBN := ROBN_
 ROBK := ROBK_
 DOBK := DOBK_
 VEZA := VEZA_
 DOBS := DOBS_
 MERA := MERA_
 TARI := TARI_
 PPUP := PPUP_
 PPAP := PPAP_
 OSNO_AKC := OSNO_AKC_
 CENA_PRO := CENA_PRO_
 CENA_DEV := CENA_DEV_
 ZALI := ZALI_
 ROK  := ROK_
 CENA_FAK := CENA_FAK_
 CENA_NAB := CENA_NAB_
 CENA_VPR := CENA_VPR_
 CENA_MPR := CENA_MPR_
 STATx    := STAT_
 FLAG     := FLAG_
 ZNAK     := ZNAK_
 NAME     := NAME_
 OPIS     := OPIS_
```

```
RETURN(0)


*********************************************************************************
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                        UPSIZE 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
   SET ORDER TO 2            // indeks na sifru
   SEEK sifra_artikla        // lociraj artikal na spisku artikala
   SET ORDER TO 1            // indeks na naziv
ELSE
   // GO TOP --> NO
   // vrati se na aktuelni record spiska artikala
   // return to the current item list record
ENDIF

/*
* OPCIJA:
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)
```

NOTE

Sledeći Deo 5b ove knjige
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 i ISAM komandi i funkcija, već samo uz korišćenje PGDBE SQL komandi i funkcija.

Next Part 5b of this book
C) It shows the same application ARTICLES-CODEBOOK.EXE called **ARTICLES-CODEBOOK3.EXE**, which is made in Alaska Xbase++ PostgreSQL database technology, without using UPSIZE MACHINE and ISAM commands and functions, but only using PGDBE SQL commands and functions.

Zadnji Deo 5c ove knjige
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)

Last Part 5c of this book
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)