

```
#include "Xbp.ch"
#include "Appevent.ch"
#include "dll.ch"
```

PROCEDURE X_COBALR

```
/*
*
* 25-05-2003
*
* ZASTITA PROGRAMA OD PRENOSA NA
* DRUGI NELICENCIRANI RACUNAR PREKO FABRICKOG SERIJSKOG BROJA HARD DISKA:
* Skup funkcija koje rade samo uz
* postojanje biblioteke:
*
* COBA_DS.DLL
*
* sa funkcijama:
* - Model Number - string
* - Disk serial Number - string
* - Revision Number - string
* - Buffer size - numerik
* - Heads - numerik
* - Cylinders - numerik
* - Sectors - numerik
*
* Ove tri funkcije Koriste COBA_DS.DLL biblioteku:
*
* FUNCTION DiskSB() - cita i vraca string fabricki serijski broj diska FSBHDD
* FUNCTION KompSB() - generise i vraca string - BROJ RACUNARA izveden iz FSBHDD
* FUNCTION LiceSB(KompSB()) - generise i vraca BROJ LICENCE izveden iz broja BROJ RACUNARA
*
* Ove tri funkcije pozivaju prethodne tri funkcije:
*
* FUNCTION ZASTITA1("ZASTITA.DBF") - Ako nema licencnog fajla ZASTITA.DBF, tada
* Generise licencni fajl ZASTITA.DBF upisuje u njega DATUM i
* upisuje u njega BROJ RACUNARA. Ne upisuje BROJ LICENCE.
* Ako ima licencnog fajla ZASTITA.DBF cita iz njega BROJ LICENCE
* generise broj licence za racunar (za FSBHDD) i uporedjuje ih
* ako su brojevi isti vraca .T. (racunar je licenciran)
* u protivnom vraca .F. (racunar nije licenciran)
*
* FUNCTION ZASTITA2("ZASTITA.DBF") - Otvara Licencni fajl ZASTITA.DBF i vrsi upis BROJA LICENCE u
* taj fajl, zatim generise licencu za racunar, poredi je sa upisanom
* i izdaje poruku da je racunar licenciran (.T.) ili da nije (.F.)
* Vraca .T. ili .F. (licenciran, nije licenciran)
*
* FUNCTION ZASTITA3() - Iz rucno upisanog broja racunara BROJ RACUNARA generise broj
* BROJ LICENCE koji snima u fajl: LICENCR.TXT i taj fajl se
* E-mail-om salje korisniku programa da BROJ LICENCE upise u
* licencni fajl ZASTITA.DBF
*
* FUNCTION COBA111() - Upotreba COBA_DS.DLL za citanje parametara HDD diska.
```

Nije primenjena ni u jednoj od ostalih funkcija i procedura ovog modula, služi kao potpuno samostalna procedura

Primer poziva DLL fajla, koji nije napravljen iz Xbase++ već iz nekog drugog programskog alata, i poziva funkcija iz tog DLL fajla. Ovde se koristi fajl COBA_DS.DLL koji je identican sa fajlom DiskSerial.DLL iz foldera DiskSerial, samo što je ovaj FREE i nijedna funkcija u njemu nema parametar "LpRegNumber" koji služi za prihvatanje i proveru registarskog broja (crack) za rad sa DLL fajlom da ne bi izdavao ometajući ekran.

```

*
* PRIMENA U APLIKACIJI:
*
*****
* AKO SE IMPLEMENTIRA ZASTITA,
* POZIV NA POCETKU MAIN PROCEDURE:
*****
*
* PROCEDURE ZASTITAz()..... Citanje i Kontrola LICENCE RACUNARA iz licencnog fajla: COBA_DS.DLL
*                               Generise PUBLIC LicencaRacunara, koristi procedure iz X_COBALR.PRG
*                               iz koga poziva FUNKCTON ZASTITA1() i generise fajl ZASTITA.DBF na
*                               podrucju lokacije EXE programa - ST_adresa.
*                               LicencaRacunara=.F. znaci da nije izdata licenca
*
*****
* AKO SE IMPLEMENTIRA ZASTITA,
* POZIV IZ MENIJA RADI UPISA:
*****
* PROCEDURE ZASTITAy()..... Citanje i Kontrola LICENCE RACUNARA iz licencnog fajla: COBA_DS.DLL
*                               Generise PUBLIC LicencaRacunara, koristi procedure iz X_COBALR.PRG
*                               iz koga, samo ako je LicencaRacunara=.F., poziva FUNKCTON ZASTITA2()
*                               i dozvoljava upis LICENCE pa zatim poziv ZASTITA1() i generise fajl
*                               ZASTITA.DBF na podrucju lokacije EXE programa - ST_adresa.
*
*****
* AKO SE IMPLEMENTIRA ZASTITA,
* POZIV PRE SVAKE PROCEDURE:
*****
* FUNCTION smetaj(LicencaRacunara) ako je LicencaRacunara=.F. izdaje se poruka:
*                               MsgBox("RACUNAR NEMA LICENCU ZA RAD","STOP")
*
*/
RETURN

```

FUNCTION DiskSB()

PRIVATE nDrive,LpBuffer,LpRegNumber,xSerial_N

xSerial_N := "b"

nDrive := 0

LpBuffer := **space**(32)

// ucitaj DLL

```

nDll := DllLoad( "COBA_DS.DLL" )
IF nDll <> 0
  q = DllCall( nDll,DLL_STDCALL, "GetSerialNumber", nDrive, @LpBuffer )
  xSerial_N := ALLTRIM(@LpBuffer)
ENDIF
// Release DLL
DllUnload( nDll)
RETURN(xSerial_N)

```

```

*****

```

```

FUNCTION KompSB()

```

```

*****

```

```

// Broj racunara izveden iz serijskog broja HDD diska:

```

```

PRIVATE xBroj, sBroj_1,sBroj_2,nBroj_2,sBroj_3,xBroj

```

```

PRIVATE yBroj,n,s,i

```

```

yBroj := "" // BROJ RACUNARA

```

```

xBroj := ""

```

```

sBroj_1 := DiskSB() // SERIJSKI BROJ HDD

```

```

IF sBroj_1 = "p"

```

```

  yBroj = sBroj_1

```

```

  RETURN(yBroj)

```

```

ENDIF

```

```

// string SERIJSKI BROJ DISKA se završava null karakterom

```

```

n := len(sBroj_1) // i ima duzinu = broj znakova + null

```

```

sBroj_1 := SUBSTR(sBroj_1,1,n-1) // uklanjanje null znaka

```

```

n := len(sBroj_1)

```

```

FOR i = 1 TO n

```

```

  sBroj_2 := SUBSTR(sBroj_1,i,1) // iz stringa (brojevi) uzmi znak

```

```

  nBroj_2 := 122 - ASC(sBroj_2) // nadji mu ASC kod i oduzmi ga od 122

```

```

  sBroj_3 := CHR(nBroj_2) // rezultat prevedi u CHR kod novi znak

```

```

  xBroj := xBroj+sBroj_3 // formiraj novi string (slova i ostalo)

```

```

NEXT i

```

```

yBroj := ""

```

```

n := len(xBroj)

```

```

FOR i =1 TO n

```

```

  yBroj := yBroj+SUBSTR(xBroj,i,1)

```

```

IF i = 4 .or. i = 8 .or. i = 12 .or. i = 16

```

```

  IF i < n

```

```

    s := str(n-i,1)

```

```

    yBroj := yBroj+s+ "-"

```

```

  ENDIF

```

```

IF i = n

```

```

  s := str(n-i,1)

```

```
    yBroj := yBroj+s
ENDIF

ENDIF

NEXT i
yBroj=ALLTRIM(yBroj)
RETURN(yBroj)

*****
FUNCTION LicSB()
*****
// Licenca racunara izvedena iz broja racunara:
PARAMETERS b_rac

PRIVATE yLic, sLic_1
yLic=""

sLic_1 := b_rac // BROJ RACUNARA

IF sLic_1 = "p"
    yLic = sLic_1
    RETURN(yLic)
ENDIF

n = len(sLic_1) // duzina
DECLARE p1[n] // formiraj niz
AFILL(p1, " ")
FOR i=1 to n
    p1[i] := SUBSTR(sLic_1,n+1-i,1) // u ogledalu
NEXT i

ssLic=""
FOR i=1 to n

    // povecaj ASC kod svakog velikog slova za 3
    IF ASC( p1[i] ) >= 65 .AND. ASC( p1[i] ) <= 122

        broj = ASC(p1[i])+3
        IF broj > 122
            broj := 122
        ENDIF

        p1[i] := CHR( broj )

ENDIF

// sve sto nije broj ili slovo je "0"
IF ASC( p1[i] ) < 48 .OR. ASC( p1[i] ) > 122
    p1[i] := "3"
ENDIF
```

```
ssLic = ssLic + p1[i]
```

```
NEXT i
```

```
yLic := ALLTRIM(SUBSTR(ssLic,1,12))
```

```
IF EMPTY(yLic)=.T.
```

```
    yLic="XXXX-XXXX-XXXX"
```

```
ENDIF
```

```
RETURN(yLic)
```

```
*****
```

```
FUNCTION ZASTITA1
```

```
*****
```

```
*****
```

```
* FORMIRAJ LICENCNI FAJL I UPISI U NJEGA DATUM I BROJ RACUNARA
```

```
* NE UPISUJ BROJ LICENCE !!!
```

```
*****
```

```
PARAMETERS _fajl_
```

```
PRIVATE rac_broj, lic_broj, dat_broj, xlic_broj, vrati
```

```
SET DATE GERMAN
```

```
SET CENTURY ON
```

```
vrati := .F.
```

```
*****
```

```
IF FILE(_fajl_) = .F.
```

```
PRIVATE RADNO
```

```
RADNO=SELECT()
```

```
SELECT 111
```

```
CREATE pomocna.dbf
```

```
USE pomocna.dbf EXCLUSIVE
```

```
append blank
```

```
REPLACE field_name WITH 'VEZA1_',field_type WITH 'C',field_len WITH 50
```

```
append blank
```

```
REPLACE field_name WITH 'VEZA2_',field_type WITH 'C',field_len WITH 50
```

```
append blank
```

```
REPLACE field_name WITH 'VEZA3_',field_type WITH 'C',field_len WITH 50
```

```
USE
```

```
CREATE &_fajl_ FROM pomocna.dbf
```

```
DELETE FILE pomocna.dbf
```

```
COMMIT
```

```
* Formiraj broj racunara:
```

```
rac_broj = KompSB()
```

```
* Formiraj datum licence:
```

```
dat_broj = DTOC(DATE())
```

```
USE &_fajl_ EXCLUSIVE
```

APPEND BLANK

```
REPLACE VEZA3_ WITH rac_broj
REPLACE VEZA2_ WITH dat_broj
* REPLACE VEZA1_ WITH lic_broj: BROJ LICENCE SE OSTAVLJA PRAZAN, kako bi ovakav
* fajl korisnik programa mogao da posalje autoru na davanje licence, odnosno
* generisanje i upis broja licence u licencni fajl i povratak fajla korisniku.
```

```
USE
COMMIT
SELECT(RADNO)
```

```
ENDIF // FILE(_fajl_) = .F.
```

```
*****
```

```
*****
```

```
IF FILE(_fajl_) = .T.
```

```
*****
```

```
lic_broj := LiceSB(KompSB())
```

```
*****
```

```
radno=SELECT()
```

```
SELECT 111
```

```
USE &_fajl_
```

```
    xlic_broj = VEZA1_
```

```
USE
SELECT(radno)
```

```
IF ALLTRIM(xlic_broj) == ALLTRIM(lic_broj)
```

```
    vrati := .T.
```

```
ELSE
```

```
    vrati := .F.
```

```
ENDIF
```

```
ENDIF // FILE(_fajl_) = .T.
```

```
*****
```

```
RETURN(vrati)
```

```
*****
```

```
FUNCTION ZASTITA2
```

```
*****
```

```
*****
```

```
* PROGRAM ZA UPIS LICENCE U LICENCNI FAJL - ALAT KORISNIKA
```

```
*****
```

PARAMETERS _fajl_**PRIVATE** XX3,radno,vrati,prozorl**PRIVATE** rac_broj, lic_broj, xrac_broj, xlic_broj, xdat_broj**PRIVATE** sDa,sNe,sBr,sLr**SET DATE** GERMAN**SET CENTURY** ON

vrati := .F.

LicencaRacunara = .F.

IF FILE(_fajl_)=.F.

* nema licencnog fajla

RETURN(vrati)**ENDIF**

rac_broj := **KompSB**()lic_broj := **LiceSB**(**KompSB**())

radno=**SELECT**()**SELECT** 111**USE** &_fajl_ **EXCLUSIVE****IF EMPTY**(VEZA3_)=.T.**REPLACE** VEZA3_ **WITH** rac_broj**ENDIF**

xrac_broj = VEZA3_

xlic_broj = VEZA1_

xdat_broj = VEZA2_

USE**SELECT**(radno)**IF ALLTRIM**(xlic_broj) == **ALLTRIM**(lic_broj)

vrati := .T.

ELSE

vrati := .F.

ENDIF

save **screen to** prozorl

sDa := "RACUNAR JE LICENCIRAN ZA RAD"

sNe := "RACUNAR NIJE LICENCIRAN ZA RAD"

sBr := "SR.BROJ RACUNARA: "

sLr := "LICENCA RACUNARA: "

```
@ 09,14 CLEAR TO 14,66
@ 09,14 TO 14,66
```

```
IF vrati = .T.
```

```
@ 10,15 say sDa
@ 12,15 say sBr+ALLTRIM(xrac_broj)
@ 13,15 say sLr+ALLTRIM(xlic_broj)
INKEY(0)
```

```
ENDIF
```

```
IF vrati = .F.
```

```
SET CURSOR ON
@ 10,15 say sNe
@ 12,15 say sBr+ALLTRIM(xrac_broj)
@ 13,15 say sLr GET xlic_broj PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
READ
CLEAR GETS
```

```
DO WHILE .T.
  INKEY(0)
  * // sa Alt+Y automatski se upisuje broj licence
  IF LASTKEY()=277 && // alt+Y
    xlic_broj := lic_broj
  ENDIF
  EXIT
ENDDO
```

```
radno=SELECT()
SELECT 111
USE &_fajl_ EXCLUSIVE
REPLACE VEZA1_ WITH ALLTRIM(xlic_broj), VEZA2_ WITH DTOC(())
xlic_broj := VEZA1_
USE
COMMIT
SELECT(radno)
```

```
IF ALLTRIM(xlic_broj) == ALLTRIM(lic_broj)
  vrati := .T.
ELSE
  vrati := .F.
ENDIF
```

```
ENDIF
```

```
restore screen from prozorl
release prozorl
```

```
IF vrati = .T.
  alert(sDa)
ELSE
  alert(sNe)
ENDIF
```


RETURN(vrati)

FUNCTION ZASTITA3

* PROGRAM ZA GENERISANJE LICENCE IZ BROJA RACUNARA

PRIVATE boja,prozorl,xrac_broj,xlic_broj,sBr,sLr

boja=**SETCOLOR**()

save **screen to** prozorl

xrac_broj = **space**(32)

xlic_broj = **space**(32)

sBr := "SR.BROJ RACUNARA: "

sLr := "LICENCA RACUNARA: "

@ 09,14 **CLEAR TO** 14,66

@ 09,14 **TO** 14,66

@ 10,15 **say** "Upis BROJA RACUNARA i generisanje LICENCE RACUNARA:"

@ 12,15 **say** sBr **GET** xrac_broj **PICTURE** "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

@ 13,15 **say** sLr **GET** xlic_broj **PICTURE** "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

CLEAR GETS

@ 12,15 **say** sBr **GET** xrac_broj **PICTURE** "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

READ

CLEAR GETS

xlic_broj := **LiceSB**(**ALLTRIM**(xrac_broj))

@ 13,15 **say** sLr **GET** xlic_broj **PICTURE** "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

CLEAR GETS

INKEY(0)

SET PRINTER TO LICENCR.TXT

SET PRINT ON

SET CONSOLE OFF

? ''

? 'COBA Systems '

? **DTOC**(**DATE**())

? **TIME**()

? ''

? 'Licenca racunara za rad programa COBA2000 je:'

? sBr+xrac_broj

? sLr+xlic_broj

? ''

? 'Pozdrav i uspesan rad'

? ''

SET PRINTER TO

```
SET PRINT OFF
SET CONSOLE ON
```

```
*****
```

```
restore screen from prozorl
release prozorl
SETCOLOR(boja)
```

```
RETURN(0)
```

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

```
*****
```

```
FUNCTION COBA111
```

```
*****
```

```
/*
```

Primer poziva DLL fajla, koji nije napravljen iz Xbase++ vec iz nekog drugog programskog alata, i poziva funkcija iz tog DLL fajla. Ovde se koristi fajl COBA_DS.DLL koji je identican sa fajlom DiskSerial.DLL iz foldera DiskSerial, samo sto je ovaj FREE i nijedna funkcija u njemu nema parametar "LpRegNumber" koji služi za prihvatanje i proveru registarskog broja (crack) za rad sa DLL fajlom da ne bi izdavao ometajuci ekran.

```
*/
```

```
PRIVATE nDrive, LpBuffer, LpRegNumber, nDll
PRIVATE xModel_N,xSerial_N,xRevision_N
PRIVATE xHeads_S,xBuffer_S,xCylinders_S,xSectors_S
```

```
//alert("Upotreba COBA_DS.DLL za citanje parametara HDD diska")
```

```
nDrive    := 0
LpBuffer   := space(32)
```

```
nDll := DllLoad( "COBA_DS.DLL")
```

```
IF nDll <> 0
```

```
// Call via macro-operator (fastest call)
// ? &("MyFunction")( p1, @p2 )
```

```
// Call via DllCall()
nDrive    := 0
LpBuffer   := space(32)
q = DllCall( nDll, DLL_STDCALL, "GetModelNumber", nDrive, @LpBuffer )
//alert("HDD Model Number: "+ALLTRIM(@LpBuffer))
xModel_N := ALLTRIM(@LpBuffer)
```

```

nDrive    := 0
LpBuffer   := space(32)
q = DllCall( nDll, DLL_STDCALL, "GetSerialNumber", nDrive, @LpBuffer )
//alert("HDD Disk Serial: "+ALLTRIM(@LpBuffer))
xSerial_N := ALLTRIM(@LpBuffer)

nDrive    := 0
LpBuffer   := space(32)
q = DllCall( nDll, DLL_STDCALL, "GetRevisionNumber", nDrive, @LpBuffer )
//alert("HDD Revision Number: "+ALLTRIM(@LpBuffer))
xRevision_N := ALLTRIM(@LpBuffer)

```

*----- numerici

```

nDrive    := 0
q = DllCall( nDll, DLL_STDCALL, "GetBufferSize", nDrive )
//alert("HDD Buffer Size: "+ALLTRIM(str(q)))
xBuffer_S := ALLTRIM(str(q))

nDrive    := 0
q = DllCall( nDll, DLL_STDCALL, "GetDiskCylinders", nDrive )
//alert("HDD Disk Cylinders: "+ALLTRIM(str(q)))
xCylinders_S := ALLTRIM(str(q))

nDrive    := 0
q = DllCall( nDll, DLL_STDCALL, "GetDiskHeads", nDrive )
//alert("HDD Disk Heads: "+ALLTRIM(str(q)))
xHeads_S := ALLTRIM(str(q))

nDrive    := 0
q = DllCall( nDll, DLL_STDCALL, "GetSectorsOfTrack", nDrive )
//alert("HDD Sectors: "+ALLTRIM(str(q)))
xSectors_S := ALLTRIM(str(q))

```

```

// stringovi koji su dobijeni iz DLL funkcije:
// Stringovi su terminirani sa nul stringom i on se mora ukloniti,
// uklanja se zadnji znak u stringu - nul string,
// inace se u GUI modu ne mogu nastavljati jedan na drugi sa chr(13)

```

```

xModel_N   = substr(xModel_N,1,len(xModel_N)-1)
xSerial_N  = substr(xSerial_N,1,len(xSerial_N)-1)
xRevision_N = substr(xRevision_N,1,len(xRevision_N)-1)

// ... ovo ne vazi za stringove koji su dobijeni iz numerika
// xHeads_S   = substr(xHeads_S,1,len(xHeads_S)-1)
// xBuffer_S  = substr(xBuffer_S,1,len(xBuffer_S)-1)
// xCylinders_S= substr(xCylinders_S,1,len(xCylinders_S)-1)
// xSectors_S = substr(xSectors_S,1,len(xSectors_S)-1)

```

```

/*
    cr=", "
    alert("HDD Model Number: "+xModel_N+cr+;

```

```

"HDD Serial Number: "+xSerial_N+cr+;
"HDD Revision Number: "+xRevision_N+cr+;
"HDD Heads: "+xHeads_S+cr+;
"HDD Buffer Size: "+xBuffer_S+cr+;
"HDD Cylinders: "+xCylinders_S+cr+;
"HDD Sectors: "+xSectors_S)

```

```

*/

```

```

cr=chr(13)+chr(10)
MsgBox("HDD Model Number   :'+xModel_N+cr+;
      "HDD Serial Number   :'+xSerial_N+cr+;
      "HDD Revision Number :'+xRevision_N+cr+;
      "HDD Heads           :'+xHeads_S+cr+;
      "HDD Buffer Size     :'+xBuffer_S+cr+;
      "HDD Cylinders       :'+xCylinders_S+cr+;
      "HDD Sectors        :'+xSectors_S;
      "HDD parametri:")

```

```

// Release DLL

```

```

DllUnload( nDll)

```

```

ENDIF // nDll <> 0

```

```

RETURN(0)

```

```

*****
*****

```

```

*****

```

```

PROCEDURE ZASTITAZ

```

```

PUBLIC LicencaRacunara

```

```

*

```

```

* Citanje i Kontrola LICENCE RACUNARA iz licencnog fajla: COBA_DS.DLL

```

```

*

```

```

IF FILE("COBA_DS.DLL")=.T.

```

```

  * DllLoad("COBA_DS.DLL") // u programu X_COBALR.PRG

```

```

ELSE

```

```

  MsgBox("COBA_DS.DLL", "NEMA FAJLA")

```

```

ENDIF

```

```

ZASTITA_DBF = ST_adresa+ "ZASTITA.DBF"

```

```

LicencaRacunara = ZASTITA1(ZASTITA_DBF) // --> .T. ili .F. X_COBALR.PRG

```

```

* IF LicencaRacunara = .F.

```

```

  * preduzmi sledece mere:

```

```

  * okoncaj program ili

```

```

  * zavedi restrikcije

```

```

  * smetaj(LicencaRacunara) // ovde-smetaj kod poziva svake operacije...

```

```

  * // prikazuj poruku: MsgBox("RACUNAR NEMA LICENCU ZA RAD", "STOP")

```

```

* ELSE // LicencaRacunara = .T.
* smetaj(LicencaRacunara) // ne prikazuj nikakvu poruku...
* MsgBox("RACUNAR IMA LICENCU ZA RAD","NASTAVI")
* ENDIF

RETURN

*****
FUNCTION smetaj(LicencaRacunara)
*****
IF LicencaRacunara=.F.
    MsgBox("RACUNAR NEMA LICENCU ZA RAD","STOP")
ENDIF
RETURN(0)

*****
FUNCTION ZASTITAY
*****

// ako racunar nema licencu prikazi broj racunara i mesto za upis
// licence kako bi korisnik mogao da je upise, a ako je upisana
// licenca neispravna prikazi poruku o tome i uputstvo da korisnik
// treba da posalje serijski broj racunara autoru kako bi dobio
// licencu.
ZASTITA_DBF = ST_adresa+ "ZASTITA.DBF"
LicencaRacunara = ZASTITA1(ZASTITA_DBF) // --> .T. ili .F.

IF LicencaRacunara = .F. // racunar nema licencu

    LicencaRacunara = ZASTITA2(ZASTITA_DBF) // vidi broj racunara
    // i upisi licencu
    IF LicencaRacunara = .F.
        MsgBox("Serijski broj racunara treba poslati"+chr(13)+chr(10)+;
            "autoru ili prodavcu programa da bi"+chr(13)+chr(10)+;
            "na osnovu njega dobili broj licence"+chr(13)+chr(10)+;
            "koji treba upisati ovde.", "LICENCIRANJE")
        ENDIF
    ELSE

        SET CENTURY ON
        SELECT 123
        IF NET_USE(ZASTITA_DBF,.F.,5)=.T.
            XVEZA1=ALLTRIM(VEZA1_) // broj licence - STRING
            XVEZA2=ALLTRIM(VEZA2_) // datum upisa licence - STRING
            XVEZA3=ALLTRIM(VEZA3_) // serijski broj racunara - STRING
            USE

            MsgBox( XVEZA2+CHR(13)+CHR(10)+;
                XVEZA3+CHR(13)+CHR(10)+;
                XVEZA1,"LICENCA JE OK")
            ELSE

                MsgBox("LICENCA JE OK","LICENCA")

```

ENDIF
ENDIF

RETURN(0)