

ADS SERVER I ALASKA Xbase++

DEO 2

FORMIRANJE BAZE PODATAKA

PRVI KORACI SA XBASE++ I ADVANTAGE DATABASE SERVEROM

ADS - ADVANTAGE DATABASE - CLIENT SERVER

ALS - ADVANTAGE LOCAL SERVER



PROJEKAT:

BAST
Business Account Software Technology
POSLOVNI I RA UNOVODSTVENI SOFTVER
ZA PREDUZETNI KE FIRME I AGENCIJE ZA
KNJIGOVODSTVO

POSTAVKA SISTEMA U RA UNARU SERVER I KLIJENT

Postavka sistema opisana je i data je u tekstu: DEO 1. ove knjige.

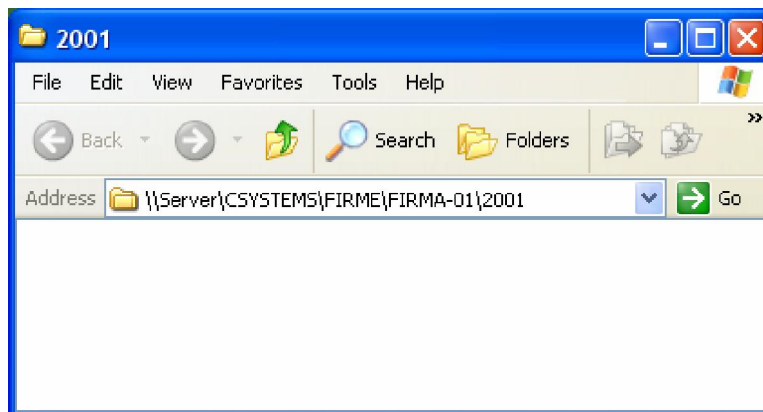
U ra unaru KLIJENT nalazi se aplikacija ADSTEST2.EXE u folderu
\\KLIJENT\CSYSTEMS\

U ra unaru klijentu fajlu ADS.INI izabran je broj 3. Rad sa ADS i ALS serverom.

U ra unaru SERVER nalazi se baza podataka u folderu
\\SERVER\CSYSTEMS\FIRME\FIRMA-01\2001

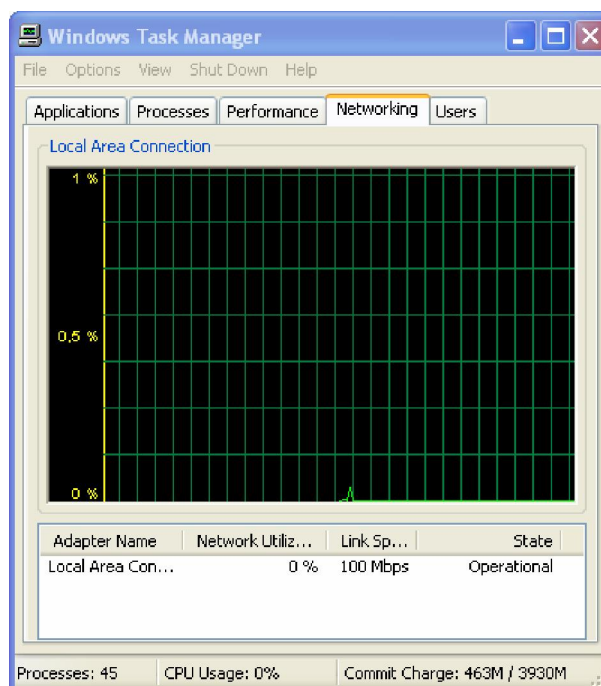
Otvorite taj folder preko shortcat-a na desktopu da bi mogli da posmatrate doganjanja u njemu i ispraznite ga. Dobićete situaciju kao na slici 1.

Slika 1.



Zatim na ra unaru serveru na desktopu otvorite Windows Task manager i aktivirajte tab Networking da bi posmatrali doganjanja na mreži

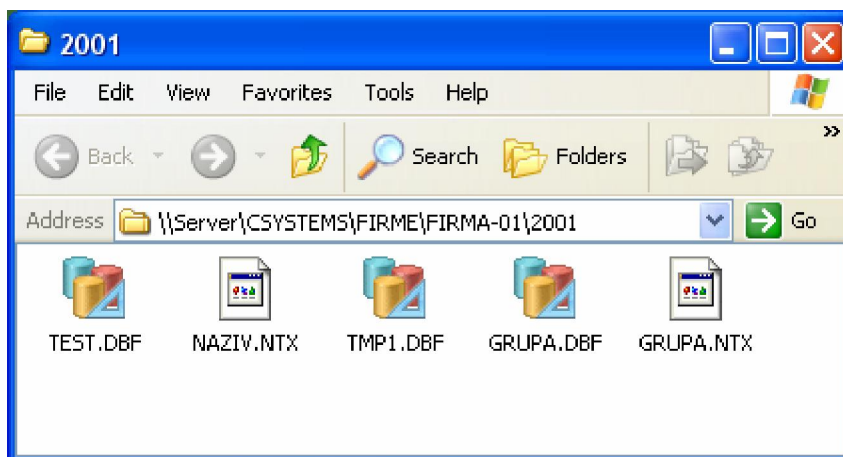
Slika 2.



Posle ovih operacija pripreme za test programa startujte na ra unaru klijentu program ADSTEST2.EXE

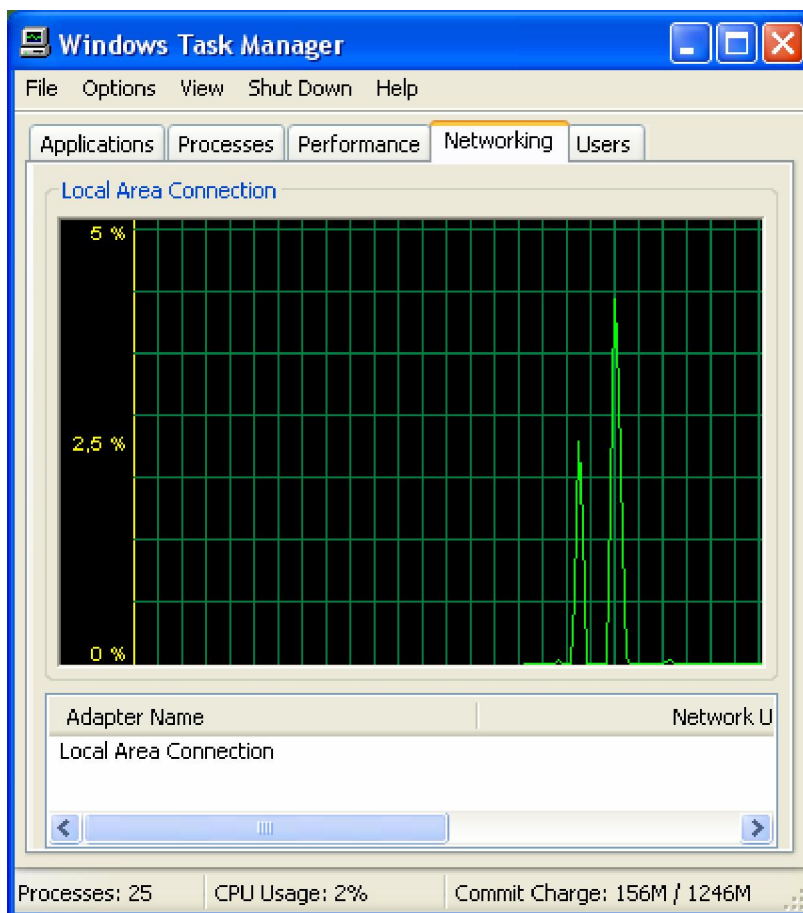
Program ADSTEST2.EXE formira e na ra unaru serveru u bazi podataka u folderu 2001 fajlove baze podataka. On e dinami ki kreirati bazu podataka na serveru. Po završetku programa i poruke o diskonektovanju sa servera dobi e se slede a slika:

Slika 3.



Istovremeno, pojavi e se prikaz mrežne aktivnosti na serveru prikazan na slici 4.

Slika 4.



Prvi pik je prikazao kreiranje i indeksiranje fajlova TEST.DBF i TMP1.DBF a drugi pik je prikazao indeksiranje i kreiranje sa TOTAL ON fajla GRUPA.DBF

PROJECT - SOURCE CODE

```
* ADSTEST2.BAT

ERASE ____OBJ\*.OBJ
PBUILD.EXE ADSTEST2.XPJ > ADSTEST2.TXT
pause

* ADSTEST2.XPJ

[PROJECT]
    COMPILE      = xpp
    COMPILE_FLAGS = /q
    DEBUG        = no
    GUI          = yes
    LINKER       = alink
    LINK_FLAGS   =
    RC_COMPILE   = arc
    RC_FLAGS     = /v
    OBJ_DIR      = ____OBJ
    ADSTEST2.XPJ

[ADSTEST2.XPJ]

    ADSTEST2.EXE

[ADSTEST2.EXE]

    ADSTEST2s.OBJ
    ADSTEST2s.PRG // start ADSDBE engine + Connection ADS+ALS

    ADAC20B.LIB // XBASE++
    ADSUTIL.LIB // ADSDBE

* ADS SERVER I ALASKA Xbase++
* © COBA Systems 07.12.2009
*
* ADSTEST2s.PRG *
*

#include "Appevent.ch"
#include "Xbp.ch"
#include "common.ch"

#include "Gra.ch"
#include "Font.ch"
#include "color.ch"

#include "xbtsys.ch"

#include "ads.ch"
#include "adsdbe.ch"

#include "Appbrow.ch"
MEMVAR appObject

PROCEDURE AppSys
// Desktop remains application window
```

```
SET CHARSET TO ANSI
RETURN

// Load ADSDBE as default database engine.
PROCEDURE DbeSys
  IF !DbeLoad( "ADSDBE" )
    ConfirmBox("U itavanje ADSDBE mašine nije uspelo", ;
              "ADSDBE STOP",XBPMB_OK,XBPMB_CRITICAL)
  ENDIF
  DbeSetDefault( "ADSDBE" )
RETURN

PROCEDURE Main

LOCAL cConnect := "DBE=ADSDBE; SERVER=\\SERVER\SYSTEMS"
LOCAL oSession := DacSession():new( cConnect )

IF !oSession.isConnected()
  ConfirmBox("Konekcija na server nije uspela", ;
            "::SERVER::STOP",XBPMB_OK,XBPMB_CRITICAL)
  QUIT
ELSE
  ConfirmBox("Konekcija na server izvršena", ;
            "::SERVER::OPEN",XBPMB_OK,XBPMB_INFORMATION)
ENDIF

SET DEFAULT TO \\SERVER\SYSTEMS\FIRME\FIRMA-01\2001

//----- CREATE, INDEX ON, COPY STRUCTURE
MsgBox("Operacija TEST.DBF", "Sledi:")
01.  build_dbf("TEST.DBF")
02.  USE TEST NEW SHARED
03.  full_dbf("TEST.DBF")
04.  brr := ALLTRIM(STR(RECCOUNT()))
05.  COPY STRUCTURE TO Tmpl.DBF
06.  INDEX ON R_NAZIV TO NAZIV
07.  GO TOP

APPBROWSE ; // Define browser
TITLE "Napravljena TEST.DBF sa brojem slogova = "+brr ;
      POSITION CENTER SIZE 40, 42 PERCENT
APPDISPLAY // Display browser modal

//----- TOTAL ON, REPLACE ALL, FLOCK(), UNLOCK
MsgBox("Operacija GRUPA.DBF", "Sledi:")
08.  INDEX ON R_GRUPA TO GRUPA
09.  TOTAL ON R_GRUPA FIELDS R_COUNT TO GRUPA
10.  USE
11.  USE GRUPA NEW SHARED
12.  IF FLOCK()
13.  REPLACE ALL R_NAZIV WITH ;
      "GRUPA ROBE "+R_GRUPA +" artikala " + ALLTRIM(STR(R_COUNT))
14.  REPLACE ALL R_SIFRA WITH SPACE(20), R_JMERE WITH SPACE(20) //...
15.  UNLOCK
16.  ENDIF
17.  GO TOP
18.  brr := ALLTRIM(STR(RECCOUNT()))

APPBROWSE ; // Define browser
TITLE "Napravljena GRUPA.DBF sa brojem slogova = "+brr ;
```

```
        POSITION CENTER SIZE 40, 42 PERCENT
APPDISPLAY      // Display browser modal
//-----

// Close databases and disconnect from server.
DbCloseAll()
oSession:disconnect()
ConfirmBox("Konekcija na server prekinuta", ;
"::SERVER::CLOSE::", XBPMB_OK, XBPMB_WARNING)

RETURN

// formiraj praznu test.dbf
FUNCTION build_dbf(DBfile)
LOCAL aStructure
radno := SELECT()
    aStructure := {;
        { "R_GRUPA",      "C",    2,    0 },,;
        { "R_SIFRA",     "C",    5,    0 },,;
        { "R_NAZIV",     "C",   25,    0 },,;
        { "R_JMERE",     "C",    3,    0 },,;
        { "R_PDV",       "N",    5,    1 },,;
        { "R_CENA",      "N",   13,    2 },,;
        { "R_ZALIHA",    "N",   15,    3 },,;
        { "R_DATUM",     "D",    8,    0 },,;
        { "R_COUNT",     "N",    7,    0 } }
    DbCreate(DBfile, aStructure)
    // DBfile -> CLOSE
SELECT(radno)
RETURN NIL

// napuni sa test podacima test dbf
FUNCTION full_dbf(DBfile)
LOCAL i, kod := 1
FOR i=1 TO 1000
    APPEND BLANK
    REPLACE R_SIFRA WITH STRZERO(i,5)
    REPLACE R_DATUM WITH DATE()+i
    REPLACE R_GRUPA WITH STRZERO(MONTH(R_DATUM),2)
    IF kod > 26
        kod := 1
    ENDIF
    REPLACE R_NAZIV WITH CHR(64+kod)+"ROBA NAZIV"
    kod := kod + 1
    REPLACE R_JMERE WITH IIF(kod>10,"KOM","Kg")
    REPLACE R_PDV WITH IIF(kod>13,18,8)
    REPLACE R_CENA WITH ROUND( (VAL(DTOS(R_DATUM))/100000)*kod ,0 )
    REPLACE R_ZALIHA WITH INT(R_CENA/10)
    REPLACE R_COUNT WITH 1
NEXT i
COMMIT
RETURN NIL
```

ANALIZA OPERACIJA

KAKO BI SVE OVO RADILO KAO KLASIKA

Ako napravite identičan program kao program ADSTEST1s.EXE ali koji ne radi sa ADSDBE engine i sa ADS serverom, ve radi sa default Alaska Xbase++ DBFNTX engineom, sve radi besprekorno. Kako to proveriti:

```
U project fajl ADSTEST2.XPJ umesto linija
  ADSTEST2s.OBJ
  ADSTEST2s.PRG // start ADSDBE engine + Connection ADS+ALS
stavite linije
  ADSTEST2x.OBJ
  ADSTEST2x.PRG // start DBFNTX engine
```

Projekat e tada aplikaciju ADSTEST2.EXE napraviti koriste i fajl ADSTEST2x.PRG odnosno napravi e standardni Alaska Xbase++ program bez servera. Kod tog programa je:

```
* ALASKA Xbase++
* © COBA Systems 07.12.2009
*
* ADSTEST2x.PRG *
*
#include "Appevent.ch"
#include "Xbp.ch"
#include "common.ch"
#include "Gra.ch"
#include "Font.ch"
#include "color.ch"
#include "xbtsys.ch"

#include "Appbrow.ch"
MEMVAR appObject

PROCEDURE AppSys
// Desktop remains application window
  SET CHARSET TO ANSI
RETURN

  PROCEDURE Main

  SET DEFAULT TO \\SERVER\SYSTEMS\FIRME\FIRMA-01\2001

//----- CREATE, INDEX ON, COPY STRUCTURE
  MsgBox("Operacija TEST.DBF", "Sledi:")
  build_dbf("TEST.DBF")
  USE TEST NEW SHARED
  full_dbf("TEST.DBF")
  brr := ALLTRIM(STR(RECCOUNT()))
  COPY STRUCTURE TO Tmp1.DBF
  INDEX ON R_NAZIV TO NAZIV
  GO TOP

  APPBROWSE ; // Define browser
  TITLE "Napravljena TEST.DBF sa brojem slogova = "+brr ;
  POSITION CENTER SIZE 40, 42 PERCENT
  APPDISPLAY // Display browser modal
```

```
//----- TOTAL ON, REPLACE ALL, FLOCK(), UNLOCK
MsgBox("Operacija GRUPA.DBF", "Sledi:")
INDEX ON R_GRUPA TO GRUPA
TOTAL ON R_GRUPA FIELDS R_COUNT TO GRUPA
USE
USE GRUPA NEW SHARED
IF FLOCK()
  REPLACE ALL R_NAZIV WITH ;
  "GRUPA ROBE "+R_GRUPA +" artikala " + ALLTRIM(STR(R_COUNT))
  REPLACE ALL R_SIFRA WITH SPACE(20), R_JMERE WITH SPACE(20) //...
  UNLOCK
ENDIF
GO TOP
brr := ALLTRIM(STR(RECCOUNT()))

RETURN
```

Funkcije:

build_dbf(DBfile) i full_dbf(DBfile) iste su kao u ADSTEST2s.PRG modulu.

KAKO SVE OVO RADI KAO KLIJENT-SERVER

Sada vršimo analizu istog koda koji se ovde koristi i za rad na serveru. Kod za dinami ko formiranje baze podataka na serveru, koji bez problema radi i bez ADS servera (bez ADSDBE engina), izgleda ovako:

Slika 5.

```
01. build_dbf("TEST.DBF")
02. USE TEST NEW SHARED
03. full_dbf("TEST.DBF")
04. brr := ALLTRIM(STR(RECCOUNT()))
05. COPY STRUCTURE TO TMP1.DBF
06. INDEX ON R_NAZIV TO NAZIV
07. GO TOP
```

U liniji 01 kreira prazan TEST.DBF fajl na serveru, u liniji 03 ga otvara za rad kao shared a u liniji 04 ga puni sa 1000 slogova. U liniji 05 kreira se temporary fajl TMP1.DBF copiranjem strukture postoje eg TEST.DBF fajla, a u liniji 06 vrši se indeksiranje TEST.DBF fajla. Sve se dešava na serveru i preko ADSDBE engina i engin-a servera.

tako e, kod za kreiranje i editovanje fajla GRUPA.DBF napisan u najkra oju mogu oju sintaksi izgleda ovako:

Slika 6.

```
08. INDEX ON R_GRUPA TO GRUPA
09. TOTAL ON R_GRUPA FIELDS R_COUNT TO GRUPA
10. USE
11. USE GRUPA NEW SHARED
12. IF FLOCK()
13. REPLACE ALL R_NAZIV WITH ;
    "GRUPA ROBE "+R_GRUPA +" artikala " + ALLTRIM(STR(R_COUNT))
14. REPLACE ALL R_SIFRA WITH SPACE(20), R_JMERE WITH SPACE(20) //...
15. UNLOCK
16. ENDIF
17. GO TOP
18. brr := ALLTRIM(STR(RECCOUNT()))
```


U liniji 09 kreira se rekapitulacija broja artikala po grupama za podatke iz tabele TEST.DBF i ista se smešta u novu tabelu GRUPA.DBF. U liniji 12 vrši se zaklju avanje tabele GRUPA.DBF da bi se u njoj izvršio upis novih podataka u liniji 13 i 14, a zatim se u liniji 15 tabela otklju ava da bi i drugi korisnici mogli da joj pristupe. Kod je isti kao da se radi bez ikakvog servera. Na ADS serveru, a tako e i na ALS serveru, sa postoje om konfiguracijom sistema, ovo bez problema funkcioniše.

Ako se ne izvrši zaklju avanje, odnosno ako se izbace linije 12, 15 i 16 do i e do prijave greške "Lock is required for this operation". Server se nije pobrinuo za zaklju avanje. O tome se, sa ovim na inom kodiranja, još uvek brine programer. O zaklju avanju koje vrši server e biti re i u delu knjige DEO 3.

Uglavnom, skoro bez ikakvih izmena, stari Clipper ili Xbase++ kod koji vrši upravljanje DBF tabelama u bazi podataka radi i sada, kada se ide preko ADS servera ili ALS servera.

Dakle, za sada ADS server radi isto ono što i "suva" Xbase++ aplikacija bez servera.

PRENOS PODATAKA SA SERVERA NA KLIJENTA

Ali šta e se desiti ako privremeni fajl GRUPA.DBF, ije kreiranje na serveru prili no traje, i ije kreiranje zauzima resurse servera, ho emo da formiramo na ra unaru klijentu i da ga tu na miru koristimo za pregled, štampu, analize i ostalo.

U tom slu aju postoje i kod sa slike 6 morao bi da se koriguje tako da klijentska aplikacija dobije pune adrese foldera i fajla GRUPA.DBF u ra unaru klijentu, a na na in kako bi se to uradilo i u standardnom kodu bez obaziranja na server, Slika 7:

Slika 7.

```
INDEX ON R_GRUPA TO GRUPA
TOTAL ON R_GRUPA ;
    FIELDS R_COUNT ;
    TO \\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA
USE
USE \\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA NEW SHARED
IF FLOCK()
    REPLACE ALL R_NAZIV WITH ;
    "GRUPA ROBE "+R_GRUPA +" artikala " + ALLTRIM(STR(R_COUNT))
    REPLACE ALL R_SIFRA WITH SPACE(20), R_JMERE WITH SPACE(20) //...
    UNLOCK
ENDIF
GO TOP
```

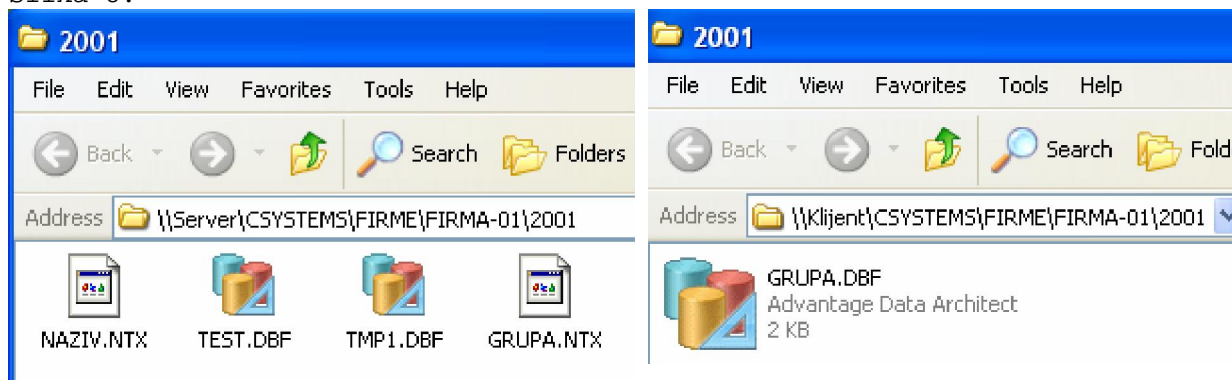
Xbase++

Ovaj kod emo prvo upisati u neserver Xbase++ aplikaciju odnosno u fajl ADSTEST2x1.PRG i u PROJECT SCRIPT FAJL ADSTEST2.XPJ upisa emo:

```
ADSTEST2x1.OBJ
ADSTEST2x1.PRG // start DBFNTX engine
```

posle starta ADSTEST2.BAT i generisanja aplikacije ADSTEST2.EXE ta aplikacija e izvršavati kod iz fajla ADSTEST2x1.PRG modula. Start aplikacije ADSREST2.EXE izvrši e sve zadate operacije besprekorno. U ra unaru serveru pojavi e se nova baza podataka a u ra unaru klijentu u folderu 2001 pojavi e se fajl privremeni fajl GRUPE.DBF i sa tog mesta bi e njegovi podaci prikazani u browser tabeli.

Slika 8.



Xbase++ ADS server

Me utim, ako isti ovaj kod sa slike 7 upišemo u kopiju modula ADSTEST2s.PRG u modul ADSTEST2s1.PRG, umesto koda na slici 6, i od toga napravimo ADSTEST2.EXE aplikaciju, pa je startujemo, ovaj kod više ne e raditi i do i e do poruke o grešci:

Slika 9.



Dakle ADS server se ovde mora druga ije programirati, jer on više ne vidi lokacije DBF fajlova koje se nalaze van Server drajva \\SERVER\SYSTEMS kao radna podru ja za rad sa bazom podataka.

Šta je ovde važno:

U Xbase++ aplikaciji, koja ne sadrži proceduru DbeSys(), što je slu aj kod programa ADSTESTx.PRG, pri startu aplikacije u itava se default Database Engine za:

DBFDDBE

NTXDBE

DBFNIX

U Xbase++ ADS server aplikaciji, koja sadrži proceduru DbeSys(), što je slu aj kod ADSTESTs.PRG, pri startu aplikacije u itava se samo DBE koji je postavljen u toj proceduri, u ovom slu aju to je:

ADSDBE

Kod rada sa ovom aplikacijom sve se izvršavati kako treba dok se nalazimo u okviru prostora ADS servera odnosno Server Drajva jer tu deluje ADSDBE engine. Van ovog prostora ne deluje ADSDBE ve DBFNTX ili DBFCDX engine, a njega nema. Ako ga nema, a treba nam, treba ga u itati. Zbog toga se program ADSTESTs1.PRG mora napisati na slede i na in:

```
* ADS SERVER I ALASKA Xbase++
* © COBA Systems 07.12.2009
*
* ADSTEST2s1.PRG *
*
#include "Appevent.ch"
#include "Xbp.ch"
#include "common.ch"
#include "Gra.ch"
#include "Font.ch"
#include "color.ch"
#include "xbtsys.ch"
#include "ads.ch"
#include "adsdbe.ch"
#include "Appbrow.ch"
MEMVAR appObject

PROCEDURE AppSys
// Desktop remains application window
SET CHARSET TO ANSI
RETURN

// Load ADSDBE as default database engine.
PROCEDURE DbeSys
  IF !DbeLoad( "ADSDBE", .F. ) // load engine for ADS/ALS
    ConfirmBox("U itavanje ADSDBE mašine nije uspelo", ;
              "ADSDBE STOP",XBPMB_OK,XBPMB_CRITICAL)
  ENDIF

  IF ! DbeLoad( "DBFDDBE", .T.) // load engine for DBF files
    msgbox( "Database Engine DBFDDBE not loaded" , "STOP" )
  ENDIF
  IF ! DbeLoad( "NTXDDBE", .T.) // load engine for NTX files
    msgbox( "Database Engine NTXDDBE not loaded" , "STOP" )
  ENDIF

  DbeBuild( "DBFNTX", "DBFDDBE", "NTXDDBE" ) // DBE=DBFNTX

  DbeSetDefault( "ADSDBE" )
RETURN

PROCEDURE Main

LOCAL cConnect := "DBE=ADSDBE; SERVER=\\SERVER\CSYSTEMS"
LOCAL oSession := DacSession():new( cConnect )

IF !oSession:isConnected()
  ConfirmBox("Konekcija na server nije uspela", ;
            "::SERVER::STOP",XBPMB_OK,XBPMB_CRITICAL)
  QUIT
ELSE
  ConfirmBox("Konekcija na server izvršena", ;
            "::SERVER::OPEN",XBPMB_OK,XBPMB_INFORMATION)
```

```
ENDIF

SET DEFAULT TO \\SERVER\CSYSTEMS\FIRME\FIRMA-01\2001

//----- CREATE, INDEX ON, COPY STRUCTURE
MsgBox("Operacija TEST.DBF", "Sledi:")
build_dbf("TEST.DBF")
USE TEST NEW SHARED
full_dbf("TEST.DBF")
brr := ALLTRIM(STR(RECCOUNT()))
COPY STRUCTURE TO TMP1.DBF
INDEX ON R_NAZIV TO NAZIV
GO TOP

APPBROWSE ; // Define browser
TITLE "Napravljena TEST.DBF sa brojem slogova = "+brr ;
      POSITION CENTER SIZE 40, 42 PERCENT
APPDISPLAY // Display browser modal

//----- TOTAL ON, REPLACE ALL, FLOCK(), UNLOCK
MsgBox("Operacija GRUPA.DBF", "Sledi:")

INDEX ON R_GRUPA TO GRUPA

// Ovaj posao se obavlja na klijentu - van servera
DbSetDefault("DBFNTX")

TOTAL ON R_GRUPA FIELDS R_COUNT ;
      TO \\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA
USE
USE \\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA NEW SHARED
IF FLOCK()
  REPLACE ALL R_NAZIV WITH ;
  "GRUPA ROBE "+R_GRUPA +" artikala " + ALLTRIM(STR(R_COUNT))
  REPLACE ALL R_SIFRA WITH SPACE(20), R_JMERE WITH SPACE(20) //...
  UNLOCK
ENDIF
GO TOP
brr := ALLTRIM(STR(RECCOUNT()))

APPBROWSE ; // Define browser
TITLE "Napravljena GRUPA.DBF sa brojem slogova = "+brr ;
      POSITION CENTER SIZE 40, 42 PERCENT
APPDISPLAY // Display browser modal

// Posao na klijentu je završen - sledi povratak na server
DbSetDefault("ADSDBE")

//-----

// Close databases and disconnect from server.
DbCloseAll()
oSession:disconnect()
ConfirmBox("Konekcija na server prekinuta", ;
  "::SERVER::CLOSE::", XBPMB_OK, XBPMB_WARNING)

RETURN
```

Ova aplikacija e na svom startu u itati Database engine za

ADSDBE
DBFDDBE
NTXDDBE
DBFNNTX

i postavi e ADSDBE kao default radni engine koji završava sav potreban posao na serveru, ali e imati mogu nost upotrebe i DBFNNTX engine-a, kada to zatreba, odnosno kada se sa DBF fajlovima radi van vidnog podru ja servera.

KOMANDE COPY TO i APPEND FROM

Samo u jednom slu aju, kada se operacije nad bazom podataka izvršavaju na samom ADS serveru, potrebno je koristiti DBFNNTX engine. To je slu aj kada se koriste komande COPY TO i APPEND FROM

Za ove dve komande Alaskin Help za ADSDBE kaže:

COPY TO dozvoljava da kopiramo tabelu na ADS serveru. Ovo e u potpunosti biti obra eno i upravljano na daljinu što e zauzvrat smanjiti optere enje mreže i bi e vremeski kra e.

Komanda COPY TO e u potpunosti biti obra ena na ADS serveru. Ovo zahteva da obe, source i target tabela, budu ADS serverove tabele (vidljive serveru i tabele sa kojima server može upravljati), a kondicija FOR mora biti takva da se može izvršiti na ADS serveru. Ako i jedan od ovih preduslova ne može biti ispunjen, onda e komanda COPY TO biti izvršena na klijentu (vidi /SOURCE/SYS/DBBULK.PRG). Isto se odnosi i na situaciju gde su u upotrebi Xbase++ podržane opcije poput REST, WHILE ili ALL. Ovo je zato što ADS server nema takvu funkcionalnost. Za više detalja u vezi COPY TO vidi Xbase++ online dokumentaciju.

Napomena: komanda COPY TO ne poštuje ni jedno SCOPE podešenje na source tabeli.

APPEND FROM Funkcija dodaje (appenduje) sadržaj odrene izvorne tabele u trenutno aktivnu target tabelu u ternutno aktivno radno polje. Uslov je da izvorna tabela mora da postoji.

Komanda APPEND FROM e biti u potpunosti obra ena na ADS serveru. Ovim postaje obavezno da obe, izvorna i odredišna (source i target) tabela, moraju biti ADS upravljane (driven) tabele, a kondicija FOR mora biti takva da se može izvršiti (executable) od strane ADS servera.

Ako i jedan od ovih preduslova ne može biti ispunjen, onda e komanda APPEND FROM biti izvršena od strane klijenta (vidi \SOURCE\SYS\DBBULK.PRG). Isto se odnosi i na situaciju gde su u upotrebi opcije podržane od strane Xbase++-a poput REST, WHILE ili ALL. Ovo se dešava zbo toga što ADS server ne omogu ava takvu funkcionalnost. Za više detalja u vezi APPEND FROM vifi Xbase++ online dokumentaciju.

Napomena: Struktura obe - source i target tabele - mora biti potpuno identična ili e u suprotnom append biti neuspešan i desi e se runtime greška.

Slika 10.

dat je primer za COPY TO:

```
// open table on ADS server
USE Test NEW SHARED
// copies source table "Test" to "TMP1"
COPY TO TMP1.DBF FOR R_GRPUPA = "08"
```

dat je primer za APPEND FROM:

```
// open table on ADS server
USE Test NEW SHARED
// appends all records matching the FOR condition to
// the current workarea.
APPEND FROM TMP1.DBF FOR R_GRPUPA = "08"
```

Prvo, ovi primeri ne rade na ADS serveru.

Drugo, iz ovog teksta ne vidi se explicitno o emu se zapravo radi.

Verovatno Alaskin pisac helpa podrazumeva da je to jasno, ili pak o ekuje pretplatu na tehni ku podršku (Subscriptions).

Objašnjenje bi trebalo da bude sledeće:

ADS server dozvoljava preko komandi COPY TO kopiranje sadržaja (slogova) DBF fajlova sa servera na klijenta (export) i obrnuto. Takođe, sa komandom APPEND FROM dozvoljava preuzimanje sadržaja DBF fajlova sa klijenta na server (import) i obrnuto, i to radi serverov engine koji kad god su ove komande u pitanju podrazumeva da je u pitanju prenos sa servera "u polje" ili "iz polja" na server, pa ak kada je u pitanju prenos između dva DBF na samom serveru. Zbog toga ADS Server zahteva preciznu sintaksu u ovim slučajevima i ona glasi:

Slika 11.

```
MsgBox("Operacija COPY TO","Sledi:")
//dat je primer za COPY TO:
// open table on ADS server
USE Test NEW SHARED
// copies source table "Test" to "TMP1"
COPY TO TMP1.DBF FOR R_GRUPA = "08" VIA "DBFNTX"
USE

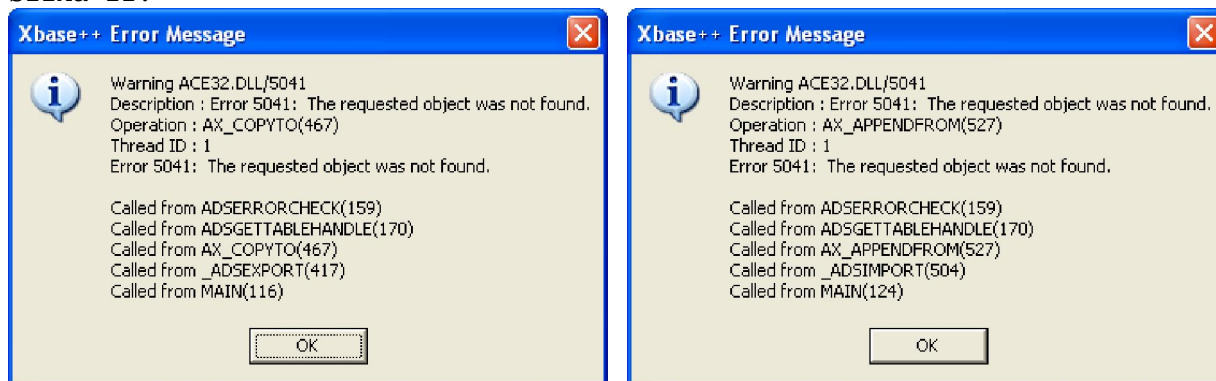
MsgBox("Operacija APPEND FROM","Sledi:")
//dat je primer za APPEND FROM:
// open table on ADS server
USE Test NEW SHARED
// appends all records matching the FOR condition to
// the current workarea.
APPEND FROM TMP1.DBF FOR R_GRUPA = "08" VIA "DBFNTX"
USE
```

Ako ovaj deo koda sa slike 11 upišete u modul ADSTEST2s1.PRG između u komande DbeSetDefault("ADSDBE") i komande DbCloseAll(), a izostavite izraz:

VIA "DBFNTX"

pa će se startom programa ADSTEST2.EXE dobiti poruka o grešci:

Slika 12.



Ako se komande zadaju sa izrazom **VIA "DBFNTX"** koji aktivira rad u području van servera, sve radi bez problema i prvo se iz fajla TEST.DBF u fajl TMP1.DBF na serveru prenese 91 slog (koliko broji grupa 08), a zatim se iz TMP1.DBF fajla u fajl TEST.DBF prenese 91 slog pa fajl TEST.DBF ima 1091 slog, što se može odmah proveriti iz Advantage Database architect-a.

Slika 13.

```
Kada se sa servera kopira na klijenta (export)
target := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF // nalazi se na serveru
COPY TO (target) VIA "DBFNTX"
```

Slika 14.

```
kada se sa klijenta kopira na server (export)
source := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE (source) NEW SHARED VIA "DBFNTX"
COPY TO TEST.DBF // nalazi se na serveru
```

Slika 15.

```
kada se sa servera preuzima iz klijenta (import)
source := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF // nalazi se na serveru
APPEND FROM (source) VIA "DBFNTX"
```

Slika 16.

```
kada se sa klijenta preuzima iz servera (import)
target := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE (target) VIA "DBFNTX"
APPEND FROM TEST.DBF // nalazi se na serveru
```

VARIJANTE I NJIHOVA UPOTREBA

_DbExport() i _DbImport()

Slika 17.

Komanda COPY TO ima varijantu u funkciji:

```
target := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF // nalazi se na serveru
_DbExport( target )
_DbExport( target,,{|| R_GRUPA=="08"} )
_DbExport( target,,{|| R_GRUPA=="08"},,,,,,"DBFNTX" )
```

Sve tri varijante ove funkcije radi e na ADS Serveru, ak i prva i druga i ako nemaju uklju ivanje na "DBFNTX" DBE engine.

Slika 18.

Komanda APPEND FROM ima varijantu u funkciji:

```
source := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF // nalazi se na serveru
_DbImport( source )
_DbImport( source,,{|| R_GRUPA=="08"} )
_DbImport( source,,{|| R_GRUPA=="08"},,,,,,"DBFNTX" )
```

Sve tri varijante ove funkcije radi e na ADS Serveru, ak i prva i druga i ako nemaju uklju ivanje na "DBFNTX" DBE engine.

Ove dve funkcije _DbExport() i _DbImport() ugra ene su u Alaska Xbase++ ali u postoje em helpu nisu dokumentovane. Date su u kodu i demonstrirane su u modulu `alaska\hppw32\source\sys\dbbulk.prg`.

Funkcije su specijalno napisane za situacije u kojima se radi sa dva razli ita DBE engina, i automatski podešavaju komunikaciju izme u dva DBE. funkcije su dosta sporije od COPY TO i APPEND FROM kada se radi sa velikim

brojem slogova. Kod prenosa malog broja slogova razlika u brzini je zanemarljiva.

ZAMENE I NJIHOVA UPOTREBA

Scatter() i Gather()

Slika 19.

Komande COPY TO i APPEND FROM mogu se zameniti i tehnikama direktnog upisa u slog podržanom kroz funkcije Scatter() i Gather()

```
USE TEST.DBF NEW SHARED ALIAS "TEST" // na serveru
USE TMP1.DBF NEW EXCLUSIVE ALIAS "TMP"
SELECT "TEST"
GO TOP
DO WHILE .NOT. EOF()

    PRIVATE aValue := {1,2,3,4,5,6,7,8,9}
    Scatter(aValue) // u itaj sadržaj sloga (svih 9 polja) iz source
SELECT "TMP"
    gather(aValue) // upiši sadržaj sloga (svih 9 polja) u target
SELECT "TEST"

SKIP
ENDDO
SELECT "TMP"
USE
SELECT "TEST"
USE
```

Ovo na slici 19 je u stvari isto kao i slede i kod na slici 20 , samo je kod sa slike 19 mnogo elegantnije i bolje napisan, što ne zna i da e uvek mo i da zameni kod sa slike 20.

Slika 20.

```
DO WHILE .NOT. EOF()

    // scatter je primenjeno umesto ovoga:
    PRIVATE xR_GRUPA, xR_SIFRA, xR_NAZIV
    xR_GRUPA := R_GRUPA
    xR_SIFRA := R_SIFRA
    xR_NAZIV := R_NAZIV
    ...
    SELECT "TMP"
    // gather je primenjeno umesto ovoga:
    APPEND BLANK
    REPLACE R_GRUPA WITH xR_GRUPA
    REPLACE R_SIFRA WITH xR_SIFRA
    REPLACE R_NAZIV WITH xR_NAZIV
    ...
    SELECT "TEST"

SKIP
ENDDO
```


Kod rada sa ADSDBE i ADS Serverom moraju se i na ovaj na in kodiranja primeniti ista pravila, pa bi u tom slu aju ako se sve dešava na serveru ostalo i važi lo kodiranje dato na slici 19 i 20.

Slika 21.

Ako se podaci prenose sa servera na klijenta

```
target := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF NEW SHARED ALIAS "TEST" // na serveru
USE (target) NEW EXCLUSIVE ALIAS "TMP" VIA "DBFNTX"
SELECT "TEST"
GO TOP
DO WHILE .NOT. EOF()
    PRIVATE aValue := {1,2,3,4,5,6,7,8,9}
    Scatter(aValue) // u itaj sadržaj sloga (svih 9 polja) iz source
    SELECT "TMP"
    gather(aValue) // upiši sadržaj sloga (svih 9 polja) u target
    SELECT "TEST"
SKIP
ENDDO
```

Slika 22.

Ako se podaci prenose sa klijenta na server

```
source := "\\KLIJENT\CSYSTEMS\FIRME\FIRMA-01\2001\GRUPA.DBF"
USE TEST.DBF NEW EXCLUSIVE ALIAS "TEST" // na serveru
USE (source) NEW SHARED ALIAS "TMP" VIA "DBFNTX"
SELECT "TMP"
GO TOP
DO WHILE .NOT. EOF()
    PRIVATE aValue := {1,2,3,4,5,6,7,8,9}
    Scatter(aValue) // u itaj sadržaj sloga (svih 9 polja) iz source
    SELECT "TEST"
    gather(aValue) // upiši sadržaj sloga (svih 9 polja) u target
    SELECT "TMP"
SKIP
ENDDO
```

Ove dve izvorne funkcije Scatter() i Gather() ugra ene su u Alaska Xbase++ ali u postoje em helpu nisu dokumentovane. Date su u kodu i demonstrirane su u modulu alaska\hppw32\source\sys\block.prg.

Funkcije su specijalno napisane za situacije u kojima se radi sa prenosom kompletnih slogova izme u dva DBF fajla, a da se za to ne bi pisao komplikovan i dug kod kada se koriste komande APPEND BLANK i REPLACE. Ove dve funkcije su elegantne i lepe, ali nisu svuda upotrebljive, a pogotovo tamo gde treba kontrolisati veli inu numerika koji se upisuje u polje sloga kako zbog velikog broja koji ne može da se smesti u polje ne bi došlo do RunTime greške.

funkcije su dosta sporije od COPY TO i APPEND FROM kada se radi sa velikim brojem slogova. Kod prenosa malog broja slogova razlika u brzini je zanemarljiva. Brzina im je ista kao i kod _DbExport() i _DbImport() funkcija.

ZAKLJU AK

Kada se radi sa ADS i ALS serverom mora se raditi sa dva ili tri DBE engine-a. Zbog toga treba dobro prou itati rad sa DBE.

Kada se aktivira ADS server treba dobro obratiti pažnju na kodiranje i na momente promene radnog DBE (po pravilu to je uvek ADSDBE) u pomo ni DBE za upotrebu DBF fajlova van servera (to je ili DBFNTX ili DBFCDX DBE).

Na ALS serveru su sve komande brze jer on pristupa bazi podataka direktno preko Windows API funkcija. Na ADS serveru su komande za masovnu obradu i prenos podataka mnogo sporije (zbog na ina rada ADS servera i svih dodatnih kontrola i zaštita koje on vrši radi sigurnosti transakcija i podataka) a sa porastom broja podataka koji se obra uju linearno se pove ava i vreme za njihovu obradu. Zbog toga treba ta no znati kada i kako treba odre enu komandu ili funkciju primeniti da se ne bi drasti no umanjile performanse i brzina rada aplikacije. Ovo se najbolje da utvrditi testiranjem.

Moji prvi testovi ADS servera 9.10 na Windows XP Professional sistemu pokazuju slede e rezultate :

1. Uvek gde se može treba koristiti COPY TO i APPEND FROM
2. Samo kada se to mora treba koristiti sporije funkcije
_DbExport i _DbImport,
kao i funkcije
Scatter i Gather,
odnosno komande
APPEND BLANK i REPLACE
3. COPY TO treba koristiti
 - kada se sa servera prenose podaci na server
 - kada se sa klijenta podaci prenose na server
4. APPEND FROM treba koristiti
 - kada se sa servera prenose podaci na klijenta

REZULTATI TESTA

1. Test sa prenosom 10280 slogova, pa zatim 20560 slogova izme u dva DBF fajla (testiran je ADS klijent-server bez ALS servera) dao je slede e rezultate u brzini prenosa tog broja slogova u sekundama:

Slika 23.

ENGINE VRSTA PRENOSA	ADSDBE COPY TO		ADSDBE APPEND FROM		DBFDDBE _DbExport() scatter/gather	DBFDDBE _DbImport() scatter/gather
	10280	20560	10280	20560	10280	10280
server-server	15	31	18	37	28	36
server-client	15	31	4	7	15	18
client-server	1	2	19	40	15	20

2. Test sa pore enjem brzine ADS i ALS servera na izvršenju istih poslova dao je slede e rezultate:

Slika 24.

SERVER	preneto slogova	COPY TO	APPEND FROM	Scatter/Gather
ADS	90	36,44 s	0,39 s	0,35 s
ALS	90	0,80 s	0,25 s	0,33 s
ADS	500	36,80 s	0,94 s	1,46 s
ALS	500	0,78 s	0,28 s	0,36 s
ADS	11970	33,11 s	17,77 s	56,88 s
ALS	11970	1,44 s	1,22 s	2,01 s
ADS	20560	30,79 s	27,60 s	55,31 s
ALS	20560	1,86 s	1,86 s	2,81 s

Iz rezultata testa se vidi da su na ADS serveru kod velikog broja podataka razlike u brzini prenosa u zavisnosti od korišćenog servera i korišćene komande višestruko veće. Na primeru zadnjeg testa slika 24. izvukao sam sledeće radne zaključke:

1. KOMANDA APPEND FROM koju izvršava ADS/ALS engine jedina je upotrebljiva za rad sa malim kao i sa velikim brojem slogova kada se radi sa ADS i ALS.
2. KOMANDA COPY TO koju izvršava ADS engine nije dobra. Ne treba je primenjivati kada se radi sa ADS, ali može se primenjivati kada se radi sa ALS.
3. KOMANDA APPEND BLANK/REPLACE odnosno Scatter()/Gather() funkcije, upotrebljivi su samo kod malog broja slogova kada se radi sa ADS, jer sa porastom broja slogova naglo i drastično ove komande usporavaju rad. Kada se radi sa ALS može se primenjivati jer ne gubi mnogo na brzini sa porastom broja slogova, ali je sporija od APPEND FROM.
4. Iz testa koji se može proveriti iz programa, proizilazi da je ALS server mnogo brži od ADS servera:

Slika 25.

1. test: ALS je brži za 1,6 puta (broj rekorda=90)
2. test: ALS je brži za 3,3 puta (broj rekorda=500)
3. test: ALS je brži za 13,7 puta (broj rekorda=11970)
4. test: ALS je brži za 15,0 puta (broj rekorda=20560)

POSEBNA NAPOMENA

Testovi se moraju prihvatiti sa rezervom, jer su izvedeni na OS Windows XP Professional sistemu, a ne na OS na kome po specifikaciji proizvoda radi ADS server a to je:

JEDAN OD WINDOWS SERVERA: NT/2000/2003 i 2008.

Nigde u specifikacijama proizvoda Sybase iAnywhere ne stoji da ADS server može da radi na bilo kom drugom Windows OS, osim na ovde pobrojanim Windows serverima.