

=====

Kako se formiraju LIB/DLL biblioteke za XBASE++

=====

Zadatak:

Od izvornog koda u tri fajla-modula napraviti LIB i DLL biblioteke
fajlovi: BIB1.PRG, BIB2.PRG i BIB3.PRG

1. Svaki PRG fajl u svoj OBJ fajl

Xbase Compiler XPP.EXE : KREIRA OBJ fajlove za DLL biblioteku
Program XPP.EXE od BIB1.PRG **fajla** (izvornog koda)
formira BIB1.OBJ fajl, a komanda i parametri za **to** glase:

XPP BIB1 /q /b /dll
XPP BIB2 /q /b /dll
XPP BIB3 /q /b /dll

Rezultat: BIB1.OBJ, BIB2.OBJ, BIB3.OBJ fajlovi

2. Svi OBJ fajlovi u jedan DEF fajl

Program XPPFILT.EXE : KREIRA DEF **fajl** (jedan fajl) za sve OBJ
fajlove a komanda i parametri za **to** glase:

XPPFILT BIB1.OBJ BIB2.OBJ BIB3.OBJ /f:BIB.DEF

rezultat: fajl BIB.DEF

3. Iz jednog DEF fajla jedan LIB fajl i jedan **EXP** fajl

Program AIMPLIB.EXE : KREIRA LIB **fajl** (jedan fajl - IMPORT LIBRARY)
i KREIRA **EXP fajl** (jedan fajl - EXPORT LIBRARY)
a komanda i parametri za **to** glase:

AIMPLIB BIB.DEF

rezultat: fajl BIB.LIB i fajl BIB.**EXP**

4. Iz jednog **EXP** fajla i svih OBJ fajlova - jedan DLL fajl

Xbase linker ALINK.EXE KREIRA jedan DLL fajl, a komande i parametri
za **to** glase:

ALINK /DLL BIB1.OBJ BIB2.OBJ BIB3.OBJ BIB.**EXP** /OUT:BIB.DLL

Rezultat: BIB.DLL fajl, koji ce se koristiti od bilo kog EXE fajla

koji je linkovan sa bibliotekom BIB.LIB

5. Iz jednog PRG fajla i jednog LIB/DLL fajla - jedan EXE fajl

Izvorni kod je GLAVNI.PRG i iz njega se kreira GLAVNI.OBJ sa:

```
XPP GLAVNI /q /b
```

a zatim se kreira GLAVNI.EXE sa:

```
ALINK GLAVNI.OBJ BIB.LIB
```

Rezultat: GLAVNI.EXE ce raditi samo ako postoji i BIB.DLL

Sve **IMPORT** (LIB) **biblioteke** (import libraries) koje sadrze reference ka odgovarajucim DLL bibliotekama moraju biti specificirane linkeru.

U gornjem primeru linker formira GLAVNI.EXE izvrsni program kao aplikaciju u **text** modu.

Ova aplikacija ne sadrzi kod koji se nalazi u BIB.DLL fajlu ali sadrzi pozive funkcijama iz tog fajla. Ove funkcije nisu startovanjem EXE fajla učitane u izvrsnu memoriju. Tek kada se iz glavnog programa pozove neka od funkcija koja se nalazi u BIB.DLL biblioteci tog momenta ce se kod te funkcije učitati u izvrsnu memoriju i izvršiti.

Dynamic linked **libraries** (DLL files) **form** the basis of the operating system and should generally be considered as part of an application during program development under Xbase++. This chapter discusses how **to** build DLL files and uses as an example the following three procedures, each of which is assumed **to** be programmed in a separate PRG **file**:

```
** File MAIN.PRG **           // This file is used to
PROCEDURE Main               // create an EXE file.
  SayHello()                 // Procedures are contained
  SayHi()                    // in a DLL file.
RETURN
```

```
** File SAYHELLO.PRG **       // These two files are used
PROCEDURE SayHello           // to create a DLL file.
  ? "Hello world"
RETURN
```

```
** File SAYHI.PRG **
PROCEDURE SayHi
  ? "Hi folks"
```

RETURN

The Xbase++ ProjectBuilder provides the easiest way **for** creation of a DLL **file** since it can perform **all** necessary steps automatically. It is only required **to create** a project **file** which contains separate sections **for** EXE and DLL files. An appropriate project template **for** the ProjectBuilder can look like this:

```
// File: PROJECT.XPJ
[PROJECT]
  ROOT

[ROOT]
  MAIN.EXE
  MYFUNCS.DLL

[MAIN.EXE]
  MAIN.PRG
  MYFUNCS.LIB

[MYFUNCS.DLL]
  SAYHELLO.PRG

  SAYHI.PRG
```

The entire project consists of one EXE and one DLL **file**. The template lists the corresponding PRG source files in two separate sections. The section **for** the EXE **file** includes the import library MYFUNCS.LIB which is necessary **for** using DLL functions. This template must be expanded by calling PBUILD **with** the /g option and a second **call** - without the /g option - finally creates both EXE and DLL **with** its import library **file**. **If** a DLL **file** is **to** be created without the ProjectBuilder, using a Make utility, **for** example, a **total** of five different steps must be performed:

1. Compile **all** PRG files required **for** the DLL **file with** the /dll compiler switch.
2. **Create** a **file with** the definitions **for** the modules in the DLL **file** (DEF **file**). This **file** contains a **list** of **all** functions or procedures which are exported **from** the DLL **file** and can be imported **to** an EXE **file**. This task is accomplished by the utility program XPPFILT.EXE.
3. **Create** an import **library** (LIB) and an export **file** (EXP) **from** the DEF **file**. This is done by the utility program AIMPLIB.EXE.
4. Link OBJ files **with** the EXP **file to** a DLL.
5. **To** be able **to use** the new DLL **file with** a newly created EXE **file**, the import **library** (LIB) must be linked **to** the EXE **file**.

The following example describes these five steps. It uses the three PRG files MAIN.PRG, SAYHELLO.PRG and SAYHI.PRG:

Step 1: Compiling

Files **for** an EXE **file** or a DLL **file** must be compiled differently:

```
XPP main    /q /b
```

```
XPP sayhello /q /b /dll
```

```
XPP sayhi   /q /b /dll
```

Three OBJ files are created. The MAIN.OBJ **file** can only be linked **to** an EXE **file** and the files SAYHELLO.OBJ and SAYHI.OBJ can only be linked **to** a DLL **file**.

Step 2: **Create** the DEF **file**

To use a DLL **file**, **all** functions or procedures which can be called **from** outside the DLL must be **known** (export definitions). Definitions **for** exported functions or procedures are listed in a DEF **file** which is created by the utility program XPPFILT.EXE. It generates a DEF **file from a list** of OBJ files which are **to** be linked **to** a DLL:

```
xppfilt sayhello.obj sayhi.obj /f:myfuncs.def
```

XPPFILT.EXE creates the **file** MYFUNCS.DEF which contains **all** information **for** creating the **file** MYFUNCS.DLL. The example DEF **file** looks as follows:

```
01: LIBRARY myfuncs INITINSTANCE TERMINSTANCE
02: DATA MULTIPLE NONSHARED READWRITE LOADONCALL
03: CODE LOADONCALL
04:
05: EXPORTS
06:
07: ;From object file: sayhello.obj
08: SAYHELLO
09:
10: ;From object file: sayhi.obj
11: SAYHI
```

In the DEF **file**, comment lines start **with** semicolons. **All** other lines contain statements **for** the linker. The statement LIBRARY declares the **file** name of the DLL **file** and indicates whether the initialization routines in the DLL **file** are executed only once during loading or each **time** a process requires the DLL **file**. **All** DLL files created **with** Xbase++ must execute their initialization routines **for** each **process** (each program) and **INITINSTANCE** (line 1) must always be specified.

When multiple Xbase++ programs access the same DLL **file**, they can only share the program code and not the variables declared in it. **All** data in a DLL **file** must be given the attributes MULTIPLE NONSHARED READWRITE. This is done using the statement **DATA** (line 2).

The option LOADONCALL specified in the statements DATA and CODE specify that the DLL **file** is loaded into **memory** only when a module contained in the DLL **file** is executed. The alternative is the option PRELOAD, which specifies that loading takes place **at** the start of the EXE **file** (lines 2 and 3).

Following the EXPORTS **statement** (line 5) **all identifiers** (names) **for** exported functions and procedures must be listed. Each identifier must appear on a line by itself. Only the functions or procedures specified following the EXPORTS statement can be called **from** an EXE **file**.

Note: If classes are declared in a DLL **file**, only the class names must be listed, not the method names declared **for** a class. The class name is also the name of the class **function** and this must be defined as exported.

Step 3: **Create** the import library

The DEF **file** is **used** by the utility program AIMPLIB.EXE **to create** an import **library** (LIB **file**) and an export **file** (**EXP file**):

```
aimplib myfuncs.def
```

As a result, the files MYFUNCS.LIB and MYFUNCS.**EXP** are created. The LIB **file** contains information about what can be imported by an EXE **file** and the **EXP file** defines what is exported **from** a DLL **file**.

Step 4: **Create** the DLL **file**

When the **EXP file** exists, the DLL **file** can be created by the linker. **All** OBJ files plus the **EXP file** must be specified:

```
ALINK /DLL sayhello.obj sayhi.obj myfuncs.exp /OUT:myfuncs.dll
```

OBJ files must be linked using the /DLL option. The name of the DLL **file** is defined using the /OUT option.

Step 5: **Create** the EXE **file**

The last step creates the executable EXE **file**. **All** import libraries containing references **to** additional DLL files must be specified **to** the linker:

```
ALINK main.obj myfuncs.lib
```

This **call to** ALINK creates the executable **file** MAIN.EXE as **text** mode application. It contains no code **from** the DLL **file**, but references the dynamic library MYFUNCS.DLL. The code **from** this **file** is loaded when a **function** contained in the DLL is called **from** MAIN.EXE.