

WRITE YOUR OWN SHARED LIBRARY IN PP COMPILER FOR NSBASIC:

This sample commented source enables you to write a Shared Library using [PP Compiler](#) and considering [NSBasic](#) developers as final users. To write your own library, being a NSBasic or PP Compiler developer, you have only to change **colorized** code. (This code is inspired by PP Compiler sample code written by Philippe Charrière, inspired himself, by GaussLib.pas written by Philippe Guillot, and by CodeWarrior sample code written by [Ron Glowka](#)).

Nov 22, 2003

Adrian Nicolaiev
www.niconnect.com

Here, your CreatorID

```
{ $code libr, PPNB, libr, 0 }
```

```
program PPNSBasicLib(refnum, entryP);
```

```
{ $i PalmAPI.pas }
```

Here, name of your
Shared Library

```
type
```

```
  SysLibTblEntryType = record
```

```
    dispatchTblP : pointer;
```

```
    globalsP : pointer;
```

```
    dbID : LocalID;
```

```
    CodeRsch : pointer;
```

```
  end;
```

```
  SysLibTblEntryPtr = ^SysLibTblEntryType;
```

```
  resultP = ^integer;
```

```
// program parameter declaration
```

```
// no other global must be declared
```

```
var
```

```
  refnum : UInt16;
```

```
  entryP : ^SysLibTblEntryType;
```

```
function SysLibTblEntry(refnum: UInt16): SysLibTblEntryPtr; inline($4e4f, $a0b9);
```

```
type
```

```
  This_LibGlobalst = record
```

```
    refcount: UInt16;
```

```
  end;
```

```
function This_LibOpen(refnum: UInt16): Err;
```

```
var
```

```
  entryP : SysLibTblEntryPtr;
```

```
  gl : ^This_LibGlobalst;
```

```
  e : Err;
```

```
begin
```

```
  entryP := SysLibTblEntry(refnum);
```

```
  gl := entryP^.GlobalsP;
```

```
  if gl <> nil then
```

```
  begin
```

```
    // we are already open in some other application, just increment the refcount
```

```
    gl^.refcount := gl^.refcount + 1;
```

```
    This_LibOpen := 0;
```

```
  end
```

```
  else
```

```

begin
  // need to allocate for the globals
  new(gl);
  if gl<>nil then
    begin
      entryP^.globalsP:=gl;
      e:=MemPtrSetOwner(entryP^.globalsP,0);
      if e=0 then
        begin
          gl^.refcount:=1; // initialize the globals
          This_LibOpen:=0;
        end
        else This_LibOpen:=1;
      end
      else This_LibOpen:=2; // unable to allocate
    end;

end;

function This_LibClose(refnum:UInt16;var numapps:UInt16):Err;
var
  entryP:SysLibTblEntryPtr;
  gl:^This_LibGlobalsT;
begin
  entryP:=SysLibTblEntry(refnum);
  gl:=entryP^.GlobalsP;
  if gl=nil then This_LibClose:=1 // we're not open!
  else
    begin
      gl^.refcount:=gl^.refcount-1;
      numapps:=gl^.refcount;
      if numapps=0 then
        begin
          dispose(gl);
          entryP^.GlobalsP:=nil;
        end;
      This_LibClose:=0;
    end
  end;
end;

function This_LibSleep(refnum:UInt16):Err;
begin
  This_LibSleep:=0;
end;

function This_LibWake(refnum:UInt16):Err;
begin
  This_LibWake:=0;
end;

```

```
//-----
//  specific functions
//-----
```

Here, you can write the methods
and functions of your library

```
function This_LibAdd(refnum:UInt16;a,b:integer;result:resultP):Err;
begin
  result^:=a+b;
  This_LibAdd:=0;
end;
```

```
function This_LibMul(refnum:UInt16;a,b:integer;result:resultP):Err;
begin
  result^:=a*b;
  This_LibMul:=0;
end;
```

Here, name of your
Shared Library

```
const
  LibName='PPNSBasicLib'; // library name
  LibFuncNum=6; // Number of functions defined in library
  FirstF=2*LibFuncNum+2;
  TblSize=(Length(LibName)+4+6*LibFuncNum) and $fffe;
```

VERY IMPORTANT:

number of functions

- This_LibOpen
- This_LibClose
- This_LibSleep
- This_LibWake

+ specific functions

- This_LibAdd
- This_LibMul

```
function GetTable:pointer;
inline(
  $41fa,4, // lea 4(pc),a0 ; returns table adress
  $6000+TblSize, // bra + ; skip table data (short skip<128)
  2+6*LibFuncNum, // external functions number+2
  FirstF,
  FirstF+4,
  FirstF+8,
  FirstF+12,
  FirstF+16,
  FirstF+20,
  $6000,@This_LibOpen,
  $6000,@This_LibClose,
  $6000,@This_LibSleep,
  $6000,@This_LibWake,
  $6000,@This_LibAdd,
  $6000,@This_LibMul,
  LibName
);
```

Here, you increment your functions.
The next would be:

```
FirstF+24,
$6000,@This_LibOtherFunction,
```

```
begin
  entryP^.dispatchTblP:=GetTable;
  entryP^.globalsP:=nil;
end.
```