

# Экранирование кодовых файлов первичного каталога: кросс-разработка на Блэкбокс для Windows.

Темиргалеев Е. Э.  
(30.10.2012 20:26:09 +0400)

Основная идея заключается в разработке компонентов, ориентированных на определённую реализацию каркаса Блэкбокс, используя в качестве инструмента Блэкбокс, работающий поверх другой реализации каркаса.

Пример — "консольный" каркас (в качестве реализации графических элементов каркаса используются компоненты-пустышки) как целевой и "обычный" (с полноценным графическим интерфейсом) как инструментальный.

Более конкретный пример — разработка Linux-приложений при помощи Блэкбокс 1.6-rc6 для Windows, работающего непосредственно в Linux-ОС посредством "платформы" [wine](#) →  
←. Последняя обеспечивает выполнение Windows-приложений на POSIX-совместимых ОС, что позволяет говорить о возможности распространения данного опыта за пределы ОС семейства Linux.

Для избежания пересечения одноимённых модулей из разных реализаций каркаса требуется использование "серверной" конфигурации:

- в первичной папке размещен инструментальный Блэкбокс и общие платформенно-независимые компоненты;
- во вторичной — реализация целевого каркаса и рабочий проект.

Остаётся проблема перекрытия кодовых файлов платформеннозависимых модулей из первичной папки одноимёнными модулями во вторичной, приводящее к неработоспособности инструментального Блэкбокс. Решается использованием специально собранного выполняемого файла среды. Стандартный файл дополняется модулем *PrivShield* — "экраном" кодовых файлов заданных подсистем, для которых кодовые файлы во время динамической загрузки модулем *StdLoader* читаются только из первичной папки.

Ниже приведены модуль *PrivShield* с инструкцией по сборке специального вып. файла для Windows реализации Блэкбокс версии 1.6-rc6 и [пример](#) организации рабочего места в Linux-ОС (Ubuntu).

Схема может использоваться как с другими версиями Блэкбокс для Windows при условии проверки и уточнения по необходимости реализации *PrivShield*, **завязанной на особенности *StdLoader***, так и, по аналогии, с Блэкбокс для других ОС при наличии аналогичного "серверной" конфигурации режима работы.

## Сборка CrossBB.exe

(выполняется при запуске из первичного каталога)

- **донастройка** в исходнике *PrivShield*, если требуется добавить нестандартные экранируемые подсистемы;

- компиляция *PrivShield*;

-  DevLinker.Link CrossBB.exe := Kernel\$+ Files HostFiles PrivShield StdLoader

1 Applogo.ico 2 Doclogo.ico 3 SFLogo.ico 4 CFLogo.ico 5 DtyLogo.ico 6 folderimg.ico 7 openimg.ico

8 leafimg.ico 1 Move.cur 2 Copy.cur 3 Link.cur 4 Pick.cur 5 Stop.cur 6 Hand.cur 7 Table.cur ▾

 "ert0devCommanders.SelectAndDo('DevCompiler.CompileSelection')"

MODULE PrivShield;

```
IMPORT Files, WinApi, Kernel, SYSTEM;
```

```
TYPE
```

```
Directory = POINTER TO RECORD (Files.Directory)
```

```
inner: Files.Directory;
```

```
subCodeLocs, startupSubCodeLocs: ARRAY 4+1 OF Files.Locator (* максимум папок *)
```

```
END;
```

```
VAR stdLoaderBase, stdLoaderTop: INTEGER;
```

```
(* Настройка *)
```

```
PROCEDURE AddSub (IN sub: ARRAY OF CHAR; d: Directory; base: Files.Locator);  
  CONST  codeLoc = "Code";  
  VAR    i: INTEGER; loc1, loc2: Files.Locator;  
BEGIN  
  i := 0; WHILE (i < LEN(d.subCodeLocs) - 1) & ~(d.subCodeLocs[i] = NIL) DO INC(i) END;  
  loc1 := Files.dir.This(sub).This(codeLoc); ASSERT(loc1 # NIL, 100);  
  ASSERT(loc1.res = 0, 101);  (* на всякий случай *)  
  loc2 := base.This(sub); ASSERT(loc2 # NIL, 102);  
  ASSERT(loc2.res = 0, 103);  
  loc2 := loc2.This(codeLoc); ASSERT(loc2 # NIL, 104);  
  ASSERT(loc2.res = 0, 105);  
  d.subCodeLocs[i] := loc1; d.startupSubCodeLocs[i] := loc2  
END AddSub;
```

```
PROCEDURE Configure (d: Directory; base: Files.Locator);  
BEGIN  
  AddSub("", d, base); AddSub("System", d, base);  
  AddSub("Host", d, base);  
  AddSub("Std", d, base)  
END Configure;
```

```
(* Directory *)
```

```
PROCEDURE (d: Directory) Delete (loc: Files.Locator; name: Files.Name);  
BEGIN  
  d.inner.Delete(loc, name)  
END Delete;
```

```
PROCEDURE (d: Directory) FileList (loc: Files.Locator): Files.FileInfo;  
BEGIN  
  RETURN d.inner.FileList(loc)  
END FileList;
```

```
PROCEDURE (d: Directory) GetFileName (name: Files.Name; type: Files.Type; OUT filename: Files.Name);  
BEGIN  
  d.inner.GetFileName(name, type, filename)  
END GetFileName;
```

```
PROCEDURE (d: Directory) LocList (loc: Files.Locator): Files.LocInfo;  
BEGIN  
  RETURN d.inner.LocList(loc)  
END LocList;
```

```
PROCEDURE (d: Directory) New (loc: Files.Locator; ask: BOOLEAN): Files.File;  
BEGIN  
  RETURN d.inner.New(loc, ask)  
END New;
```

```
PROCEDURE (d: Directory) Old (loc: Files.Locator; name: Files.Name; shared: BOOLEAN): Files.File;  
  CONST  sameName = "x";  
  VAR    i: INTEGER; loc1: Files.Locator; x: INTEGER;  
BEGIN  
  SYSTEM.GETREG(5, x);  (* frame pointer *)  
  SYSTEM.GET(x + 4, x); (* ret addr *)
```

```

(* Files.dir.Old вызывает только StdLoader.ThisObjFile *)
IF (stdLoaderBase <= x) & (x < stdLoaderTop) THEN
  i := 0; loc1 := d.subCodeLocs[i];
  WHILE (loc1 # NIL) & ~d.inner.SameFile(loc1, sameName, loc, sameName) DO
    INC(i); loc1 := d.subCodeLocs[i]
  END;
  IF loc1 # NIL THEN loc := d.startupSubCodeLocs[i] END
END;
RETURN d.inner.Old(loc, name, shared)
END Old;

PROCEDURE (d: Directory) Rename (loc: Files.Locator; old, new: Files.Name; ask: BOOLEAN);
BEGIN
  d.inner.Rename(loc, old, new, ask)
END Rename;

PROCEDURE (d: Directory) SameFile (loc0: Files.Locator; name0: Files.Name; loc1: Files.Locator; name1:
Files.Name): BOOLEAN;
BEGIN
  RETURN d.inner.SameFile(loc0, name0, loc1, name1)
END SameFile;

PROCEDURE (d: Directory) Temp (): Files.File;
BEGIN
  RETURN d.inner.Temp()
END Temp;

PROCEDURE (d: Directory) This (IN path: ARRAY OF CHAR): Files.Locator;
BEGIN
  RETURN d.inner.This(path)
END This;

(* Инициализация *)

PROCEDURE GetStartupDir (OUT path: ARRAY OF CHAR);
(* непортируемая — см. (Win)HostFiles.Init и (Win)HostFiles.GetPath *)
  VAR i: INTEGER;
BEGIN
  i := WinApi.GetModuleFileNameW(0, path, LEN(path));
  ASSERT((0 < i) & (i < LEN(path)), 100);
  WHILE (i > 0) & (path[i] # "\") & (path[i] # "/") & (path[i-1] # ":") DO DEC(i) END;
  path[i] := 0X
END GetStartupDir;

PROCEDURE GetStdLoaderAddress;
  VAR mod: Kernel.Module;
BEGIN
  mod := Kernel.modList;
  WHILE (mod # NIL) & ~(mod.name = "StdLoader") DO
    mod := mod.next
  END;
  ASSERT(mod # NIL, 100); (* ожидается, что StdLoader слинкован *)
  stdLoaderBase := mod.code; stdLoaderTop := mod.code + mod.csize
END GetStdLoaderAddress;

PROCEDURE Init;
  VAR startupDir: ARRAY 1024 OF CHAR; d: Directory; loc: Files.Locator;
BEGIN
  GetStdLoaderAddress;

```

```
ASSERT(Files.dir # NIL, 100); NEW(d); d.inner := Files.dir;
GetStartupDir(startupDir); loc := Files.dir.This(startupDir); ASSERT(loc.res = 0, 101);
Configure(d, loc);
Files.SetDir(d)
END Init;
```

```
BEGIN
  Init
END PrivShield.▲
```

## Пример организации рабочего места в Linux-ОС

1) Инструментальный Блэкбокс: некоторая сборка [BlackBox Component Builder 1.6 rc6](#) для Windows, расположенная в папке ~/blbox (первичная). Содержит стандартный выполняемый файл среды (BlackBox.exe) и специальный (CrossBB.exe).

2) Скрипты bb и cbb в домашней папке для старта среды из текущей папки как вторичной (команды ~/bb и ~/cbb):

Сохранить bb.sh:

```
! "ert0devCmds.RewriteThis('', 'bb.sh', 'utf8')"#!/bin/bash
wine ~/blbox/BlackBox.exe -use "" winepath -w "$PWD"" $* &
```

▲

Сохранить cbb.sh:

```
! "ert0devCmds.RewriteThis('', 'cbb.sh', 'utf8')"#!/bin/bash
wine ~/blbox/CrossBB.exe -use "" winepath -w "$PWD"" $* &
```

▲

Переместить:

```
! "omcCmdline.Exec('lin')"
mv -v ./bb.sh ~/bb ; chmod -v a+x ~/bb ;
mv -v ./cbb.sh ~/cbb ; chmod -v a+x ~/cbb ;
echo готово ; read ▲
```

3) Рабочая папка ~/wrk/wkspc-core с папками проектов. Каждая папка проекта (вторичная) содержит разрабатываемые компоненты (обычно в виде ссылок на папки подсистем, непосредственно размещённые отдельно в какой-то системе управления версиями), файл рабочего стола (см. [i21sysDesktop](#)) и побочные файлы (результаты компиляции, тестовые и т. п.).

Для удобства в рабочей папке лежит (файл .mc.menu) локальное меню Midnight Commander-a, изображённое на рисунке.

Сохранить .mc.menu:

```
! "ert0devCmds.RewriteThis('', '.mc.menu', 'utf8')"
```

```
b start bb
  cd %f
  ~/bb
```

```
l start cbb
  cd %f
  ~/cbb
```

▲

Время правки	Имя	Размер	Время правки
Окт 22 16:06	/..	-ВВЕРХ-	Окт 19 08:00
Окт 21 18:14	/.svn	4096	Окт 30 08:50
Окт 6 17:09	~0-bb-lin-~v-simple	46	Окт 28 2011
Окт 29 20:54	/l-linconsole	4096	Окт 30 08:50
Окт 17 16:21	/ADru	4096	Апр 4 2011
Окт 8 09:42	/Abf	4096	Май 25 17:49
Окт 8 2010	/Dia	4096	Окт 19 20:00
		4096	Ноя 21 2011
		4096	Ноя 14 2011
		4096	Май 18 09:39
		4096	Окт 19 20:00
		4096	Июн 1 18:00
		4096	Фев 14 2012
	/Mysql-emb	4096	Май 27 2011
	/Neospace	4096	Апр 4 2011
	/Odf	4096	Май 10 2011
	/OglLessons	4096	Апр 4 2011
	/Rocot	4096	Авг 5 2011
	/l-linconsole		

  

Меню пользователя	
b	start bb
l	start cbb

Рабочая папка синхронизируется между машинами. (Здесь — за счёт хранения всей структуры кроме побочных файлов в Subversion-хранилище.)