

# Black Box Environment

## Project Info

Over 323.839 lines or source code  
Over 1.000 unit files  
Debug lifecycle since 1995

[bbeproject1995@libero.it](mailto:bbeproject1995@libero.it)

Fax ref:(Italy) 02 700432967

## BlackBox Environment

**Language/Compiler:** Project is developed under Borland Delphi 6.0 Professional (object Pascal)

[www.borland.com](http://www.borland.com)

**Database server:** Interbase 6.01/ FireBird 1.5 (Except Pulsar.exe application)

[www.borland.com](http://www.borland.com)  
[www.ibphoenix.com](http://www.ibphoenix.com)

**Thirdy components:**

- ✓ TeeChart Professional
- ✓ TMS Comm32  
Async32 by Turbo Power
- ✓ Report Builder Enterprise

<http://www.teemach.com/>

<http://www.digital-metaphors.com/>

First application of BBE was developed in 1995 using Borland Delphi 1.0.

### Stats

(This results is provided by “Analisi” an application included in BlackBox)


Applications (exe files):	24 applications
Device Drivers (exe files):	6 driver (using DCOM technology)
Tools (exe files):	7
Source units Files « .pas »:	<b>Over 1.000 files</b>
Forms Files « .dfm »:	<b>Over 700 files</b>
Project Files “.dpr”:	Over 100
Total amount of lines source (files “.pas”):	<b>323.839 lines of source code</b>
DLL:	3
Types library:	21

### Database elements:








Number of tables in database	Over 170 tables
Number of stored procedure	4
Number of View	7
Number of generators	27
Oldest running installation	Since 1995 with nine years of data on line.

## MAIN APPLICATIONS




### IPC Server

Icon	Name	Description
	IPC Server (ipcmmain2.exe)	The corner stone of BBE. Filesize: 3.916.800 Sourced compiled: <b>68.825</b> lines. Classes : 139 Interfaces:58 Born in 1997 – Delphi 3.0










### Client applications of IPC Server

Client	Icon	Description
IPC Supervisor (ipcsuperv.exe)		Client of IPC Server Sourced compiled: <b>34.098</b> lines. Connect to data by: DCOM technology Born in 1997 – Delphi 3.0
Videoimpianto (videoimpianto.exe)		Client of IPC Server Sourced compiled: <b>23.035</b> lines. Connect to data by: database client server/multi tier, DCOM technology Born in 2003 – Delphi 6.0
IPC Programmazione (ipcproclient.exe)		Client of IPC Server Sourced compiled: <b>51.981</b> lines. Born in 1997 – Delphi 3.0 Connect to data by: DCOM technology, database client server/multi tier
IPC Talk (iptalk.exe)		Client of IPC Server Born in 2000 – Delphi 5.0 Connect to data by: DCOM technology
IPC SMS (ipcsms.exe)		Client of IPC Server Born in 2004 – Delphi 6.0 Connect to data by: DCOM technology, database
IPC Loader (ipcloader.exe)		IPC loader is a gateway from IPC Server and other applications Sourced compiled:6.234 lines. Connect to data by: DCOM technology, database client server/multi tier Born in 1999
IPC View (ipcview.exe)		Client of IPC Server (Socket Connection) Born in 2000
EnterPoint (enterpoint.exe)		Client of IPC Server (DCOM technology) Sourced compiled:12.482 lines. Born in 2001

## Main applications

Icon	Name	Description
	Planview (planview.exe)	PlanView is a server application. It is also a client application of IPC Server. Sourced compiled: <b>77.477</b> lines. Connect to data by: <ul style="list-style-type: none"> <li>✓ Database client/ server</li> <li>✓ DCOM by Dual mode with IPC Server</li> <li>✓ Socket Connection</li> </ul> Born in 1999
	Analisi (analisi2.exe)	Sourced compiled: <b>44.178</b> lines. Connect to data by: Database client/ server Born in 1995 (Delphi 1.0). Complete rebuild on January 2000
	Pulsar (Pulsar.exe)	Sourced compiled: <b>35.400</b> lines. Connect to data by: Database Born in 1996 (Delphi 2.0).

## Other applications

	IPC Label	Born in 1998 Connect to data by: database client server
	IPC EasyLabel	Born in 2000 Connect to data by: database client server
	IPC TEA	Born in 2001 Connect to data by: database client server
	IPC Clean	Born in 2004 Connect to data by: database client server
	Plan MP	Born in 2004 Connect to data by: database client server
	IPC Stampi	Born in 2000 Connect to data by: database client server
	IPC data	Born in 1998 Connect to data by: database client server
	BB UserAdmin	Born in 2000 Connect to data by: database client server
	IPC Configurator	Born in 1997 Connect to data by: database client server

## Device drivers for IPC Server

Drivers	Description
PFMA1010.exe	Born in 1997
Int15.exe	Born in 1998
Bitbus.exe	Born in 2000
BB56driver.exe	Born in 2001
BB57driver.exe	Born in 2003
BB IntManual	Born in 2003

## Applications was written with OOP technologies.

For example see the header below:

```
unit kInt15;

// "driver for Sandretto Serie 7"

interface
uses sysutils, classes, kInt15Drv, vacomm, vaclasses, Virtual15TTY_TLB, extctrls, kint15decode;

type

TXWriteLog=procedure (level:integer;msg:string) of object;
TXInt15Operation=
(Int15none, Int15Polling, int15SelectingChecked, int15SelectingOneWay, int15Broadcast);
TXInt15RxMaster=
(Int15MRxNull, Int15MRxAck, Int15MRxNack, Int15MRxBusy, Int15MRxNoMsg, Int15MrxMsg);
TXInt15RXErr  =(Int15ERxNull, Int15ERxTimeout, Int15ERxSyntax, Int15ErXmlNode, Int15ErxToggle);

TXInt15=class
private
    forcedTimeout:boolean;
    function GetKMsgCount: integer;
    function GetKMsg: TXSelecMsgNode;
public
    LasttimeRxData:tdatetime;

    Macchina:string;
    ProtVersion:integer;
    messaggiNonPervenuti:integer;
    IPCComNotify:IUNKNOWN;
    Decoder:TXInt15Decoder;
    lASTCOUNT:integer;
    fFlowControl:boolean;
    fWaiting:boolean;
    fWaitingTime:tdatetime;
    fWaitingOperation:TXInt15Operation;
    LastttyRequest:byte;
    pendingmsgtour:shortstring;
    MsgPostSuccess:boolean;

    //timeout interni
    FtimeOutIgnoreData,
    FtimeoutTimeBlock,
    FTimeOutMax:double;
    ftimeoutNearRx:double;
    //momenti t
    FTimePostRequest:tdatetime;
    FTimeFirstRxData:tdatetime;
    FTimeLastRXData:Tdatetime;

    FBindingComm:boolean;
    FReadBufSize, FWriteBufSize:integer;
    FReadBuf, FWriteBuf:pointer; //buffer di rx e tx
    FreadBytes, FwriteBytes:integer;

    FComm:TVaComm;

    //eventi
    FOnRxData:tnotifyevent; //rx data x monitor
    FOnTxData:tnotifyEvent; //tx data x monitor
    FOnRxDataExt:tnotifyEvent; //altri gestori
    FOnTxDataExt:tnotifyEvent;
    FOnwriteLog:TXWriteLog;

    //virtual form debuggers
    FtimerVirTTY:TTimer;
    FTimer:TTimer;
    FVirTTY:IComInt15VirtualMachine;
    FtimerDoDecode:TTimer;

    //toggles
    Ftoggles:array[0..255] of boolean;
    fTogglesIgnore:boolean;

    //errori in rx dopo tx
    ErrorTimeout:boolean;
    ErrorSyntax:boolean;
```

```

ErrorToggle:boolean;
ErrorBusy:boolean;
//errori logici
ErrorNack:boolean;

CounterErrorTimeout,
CounterErrorSyntax,
CounterErrorToggle,
CounterErrorBusy,
CounterErrorNack:integer;

repeatTimeOut:integer;

//flag acquisizione x selecting checked e polling
fmsgIndex:integer;

//id messaggio
fMsgId:double;
foperazione:string;

//grouping
fTxRunGroup:boolean;
fTXGroupMax:integer;
fTXGroupCurrent:integer;
fTxGroupRepeatitem:integer;
fTXGroupMsg:TXSelecMsgNode;
fTxGroupMaxRepeatItem:integer;

constructor create;
destructor destroy;override;
function ReadMessageFromString(msg:string):integer;
// Legge un messaggio
function ReadMessageFromBuf(Start:pchar;Size:integer;var CountMessages:integer;Var
root:TXSelecMsgNode):integer;
property OnRxData:Tnotifyevent read FOnRxData write fOnRxData;
property OnTxData:Tnotifyevent read FOnTxData write fOnTxData;
property Comm:TVaComm read FComm write FComm;

property KMsgCount:integer read GetKMsgCount;
property KMsg:TXSelecMsgNode read GetKMsg;

procedure DoAfterprocessMessage(success:boolean;sembra vivo:boolean);

//funzioni di lettura e scrittura su seriale
procedure SetCommDefault;
procedure SetCommHandlers;
procedure ResetCommHandlers;
procedure CommRxChar(Sender: TObject; Count: Integer);
procedure CommRxFlag(Sender: TObject);
procedure CommTxEmpty(Sender: TObject);
procedure CommRlsd(Sender: TObject);
procedure CommRing(Sender: TObject);
procedure CommError(Sender: TObject; Errors: Integer);
procedure CommDsr(Sender: TObject);
procedure CommCts(Sender: TObject);
procedure CommBreak(Sender: TObject);

//Scrive caratteri in seriale generico
function CommWrite(var Buf; count:integer):integer;
//scrive in seriale predefinite
function CommWritePolling(tty:integer):integer;
function CommWriteSelectingChecked(tty:integer;Msg:TXSelecMsgNode):integer;
function CommWriteSelectingCheckedGroup(tty:integer;Msg:TXSelecMsgNode):integer;
function CommWriteSelectingOneWay(tty:integer;Msg:TXSelecMsgNode):integer;
function CommWriteAck(tty:integer):integer;
function CommWriteNack(tty:integer):integer;
//buffer scrive sui buffers, eliminando dati precedenti!
function BufRxWrite(var Buf;count:integer):integer;
function BufTxWrite(var Buf;count:integer):integer;
procedure ResetRxBuf;
procedure ResetTxBuf;
function AddRxBuf(p:pchar;count:integer):integer;
function GetRxBufRelPointer(var maxSize:integer):pointer;

//toggles
function Toggle(i:byte):byte;
procedure InvToggle(i:byte);
procedure InvertiToggle;

//virtual machine
procedure CreateVirtualMachine(when:string);
procedure DoOnTimerVirtualMachine(Sender:tobject);

//log files

```

```
procedure WriteLog(level:integer;msg:string);

//errori
procedure ResetErrors;
procedure DoOnErrorBusy;

//altro
Procedure DoRxCharFromSerial(Sender:tobject;count:integer);
Procedure DoRxCharFromVirtualTTY(Sender:Tobject;count:integer);
Procedure ActiveRequestTimeout;
procedure DoOnTimeout(Sender:tobject);
procedure DoOnDecode(Sender:tobject);
procedure PostMessage(msg:TXSelecMsgNode;Err:TXInt15RxErr;typeMsg:TXInt15RXMaster);

//messaggi

end;
```