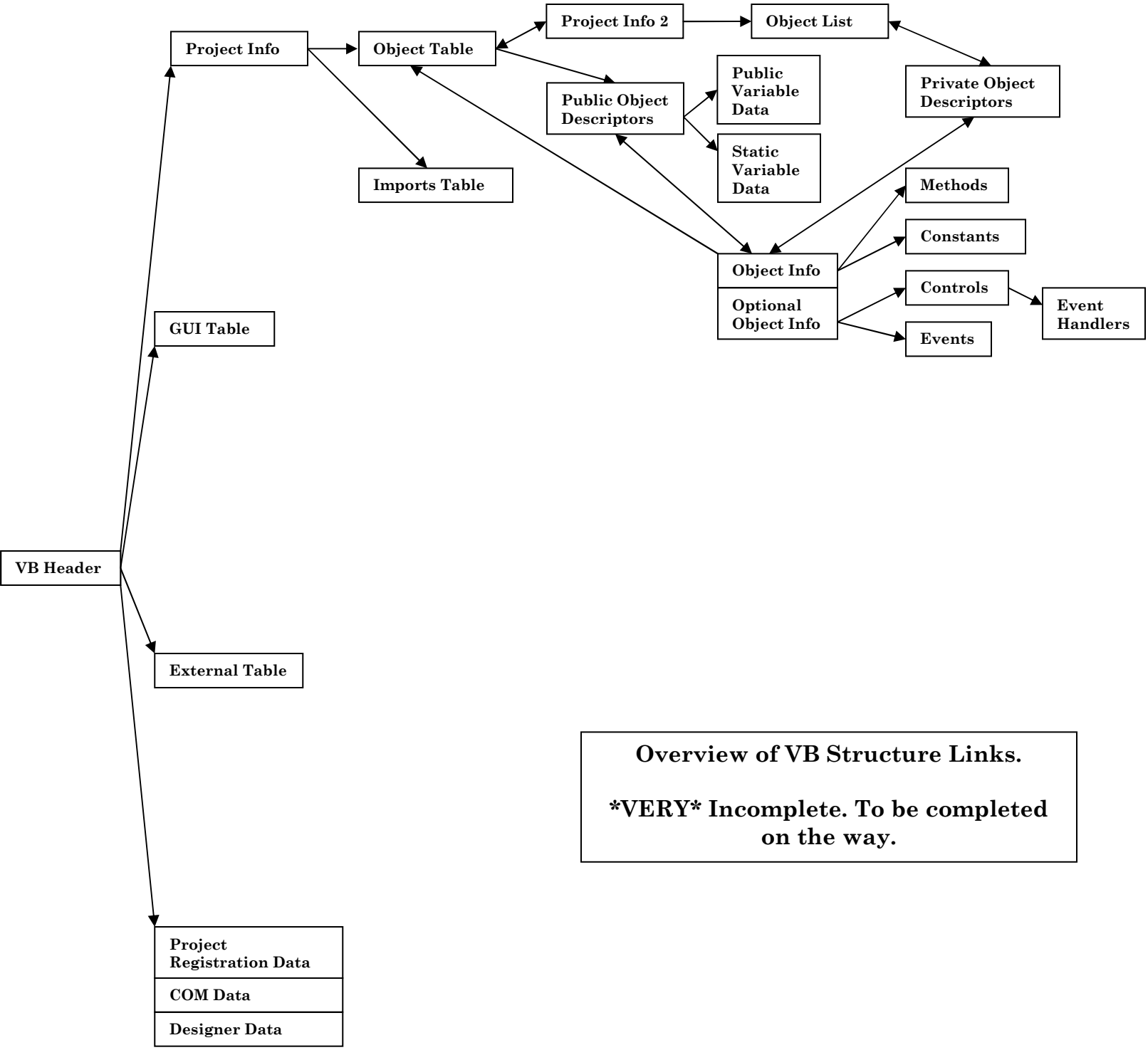
Visual Basic Image Internal Structure Format

Visual Basic Image Internal Structure Format

© 2004 Alex Ionescu
Relsoft Technologies
http://www.relsoft.net

Table of Contents



Structure Relationship Diagram

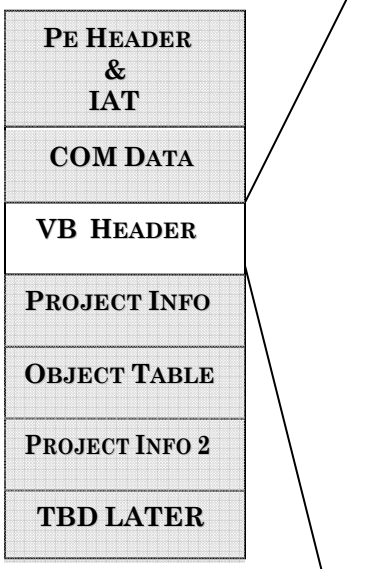
29/08/2004 More Structures, Some Fixes

This page is a work in progress. I hope you can find it useful, I will add more as I research. You're free to use this for whatever reason you want, I'm simply asking you to credit the original source (Alex Ionescu or www.relsoft.net)

1. The VB Header.

Structure name: EXEPROJECTINFO. Size: 0x68 bytes.

The VB Header is the main descriptor of any VB-compiled image file. It links into all the other structures contained in the file, and provides important language and program information. Here's an at-a-glance view:



Offset	*Name*	*Description*
0x0	szVbMagic	"VB5!" String
0x4	wRuntimeBuild	Build of the VB6 Runtime
0x6	szLangDll	Language Extension DLL
0x14	szSecLangDll	2nd Language Extension DLL
0x22	wRuntimeRevision	Internal Runtime Revision
0x24	dwLCID	LCID of Language DLL
0x28	dwSecLCID	LCID of 2nd Language DLL
0x2C	lpSubMain	Pointer to Sub Main Code
0x30	lpProjectData	Pointer to Project Data
0x34	fMdlIntCtls	VB Control Flags for IDs < 32
0x38	fMdlIntCtls2	VB Control Flags for IDs > 32
0x3C	dwThreadFlags	Threading Mode
0x40	dwThreadCount	Threads to support in pool
0x44	wFormCount	Number of forms present
0x46	wExternalCount	Number of external controls
0x48	dwThunkCount	Number of thunks to create
0x4C	lpGuiTable	Pointer to GUI Table
0x50	lpExternalTable	Pointer to External Table
0x54	lpComRegisterData	Pointer to COM Information
0x58	bSZProjectDescription	Offset to Project Description
0x5C	bSZProjectExeName	Offset to Project EXE Name
0x60	bSZProjectHelpFile	Offset to Project Help File
0x64	bSZProjectName	Offset to Project Name

Furthermore, the following flags are defined:

Thread Flags

Value	*Name*	*Description*
0x1	ApartmentModel	Specifies multi-threading using an apartment model.
0x2	RequireLicense	Specifies to do license validation (OCX only).
0x4	Unattended	Specifies that no GUI elements should be initialized.
0x8	SingleThreaded	Specifies that the image is single-threaded.
0x10	Retained	Specifies to keep the file in memory (Unattended only)

ex: A value of 0x15 specifies a multi-threaded, memory-resident ActiveX Object with no GUI.

MDL Internal Control Flags

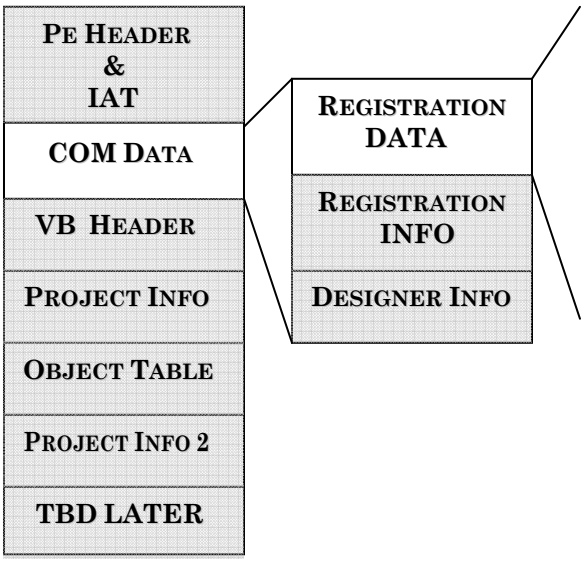
Control ID	*Value*	*Name*
0x0	0x1	PictureBox Object
0x1	0x2	Label Object
0x2	0x4	TextBox Object
0x3	0x8	Frame Object
0x4	0x10	CommandButton Object
0x5	0x20	CheckBox Object
0x6	0x40	OptionButton Object
0x7	0x80	ComboBox Object
0x8	0x100	ListBox Object
0x9	0x200	HScrollBar Object
0xA	0x400	VScrollBar Object
0xB	0x800	Timer Object
0xC	0x1000	Print Object
0xD	0x2000	Form Object
0xE	0x4000	Screen Object
0xF	0x8000	Clipboard Object
0x10	0x10000	Drive Object
0x11	0x20000	Dir Object
0x12	0x40000	FileListBox Object
0x13	0x80000	Menu Object
0x14	0x100000	MDIForm Object
0x15	0x200000	App Object
0x16	0x400000	Shape Object
0x17	0x800000	Line Object
0x18	0x1000000	Image Object
0x19	0x2000000	Unsupported
0x1A	0x4000000	Unsupported
0x1B	0x8000000	Unsupported
0x1C	0x10000000	Unsupported
0x1D	0x20000000	Unsupported
0x1E	0x40000000	Unsupported
0x1F	0x80000000	Unsupported
2nd Flag Zone	**2nd Flag Zone**	**2nd Flag Zone**
0x20	0x1	Unsupported
0x21	0x2	Unsupported
0x22	0x4	Unsupported
0x23	0x8	Unsupported
0x24	0x10	Unsupported
0x25	0x20	DataQuery Object
0x26	0x40	OLE Object
0x27	0x80	Unsupported
0x28	0x100	UserControl Object
0x29	0x200	PropertyPage Object
0x2A	0x400	Document Object
0x2B	0x800	Unsupported

ex: A value of 0x30F000 (the so called "static binary constant on most sites") actually means to initialize the Print, Form, Screen, ClipBoard Objects (0xF000) as well as the Drive/Dir Objects (0x30000). This is default on VB projects because those objects can always be accesses from a module *(ie, they are not graphic, except Forms, which can always be created).*

2. The COM Registration Data

Structure name: tagREGDATA. Size: 0x2A bytes.

The COM Registration Data contains information used if the image file is ActiveX, and contains valuable COM Registration data such as Typelib information, Designer data and Interface CLSIDs. Here's an at-a-glance view:



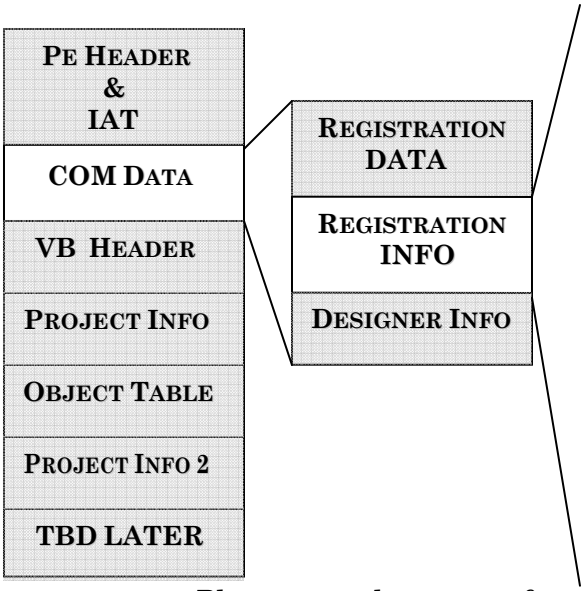
Offset	*Name*	*Description*
0x0	bRegInfo	Offset to COM Interfaces Info
0x4	bSZProjectName	Offset to Project/Typelib Name
0x8	bSZHelpDirectory	Offset to Help Directory
0xC	bSZProjectDescription	Offset to Project Description
0x10	uuidProjectClsId	CLSID of Project/Typelib
0x20	dwTlbLcid	LCID of Type Library
0x24	wUnknown	Might be something. Must check
0x26	wTlbVerMajor	Typelib Major Version
0x28	wTlbVerMinor	Typelib Minor Version

2.1 The COM Registration Info.

Structure name: tagRegInfo. Size: 0x44 bytes.

If a valid Object needs to be registered, then RegData->bRegInfo will point to the following structure, for each valid Object:

NB: All offsets are relative to tagREGDATA above.



Offset	*Name*	*Description*
0x0	bNextObject	Offset to COM Interfaces Info
0x4	bObjectName	Offset to Object Name
0x8	bObjectDescription	Offset to Object Description
0xC	dwInstancing	Instancing Mode
0x10	dwObjectId	Current Object ID in the Project
0x14	uuidObject	CLSID of Object
0x24	fIsInterface	Specifies if the next CLSID is valid
0x28	bUuidObjectIFace	Offset to CLSID of Object Interface
0x2C	bUuidEventsIFace	Offset to CLSID of Events Interface
0x30	fHasEvents	Specifies if the CLSID above is valid
0x34	dwMiscStatus	OLEMISC Flags (see MSDN docs)
0x38	fClassType	Class Type
0x39	fObjectType	Flag identifying the Object Type
0x3A	wToolboxBitmap32	Control Bitmap ID in Toolbox
0x3C	wDefaultIcon	Minimized Icon of Control Window
0x3E	fIsDesigner	Specifies whether this is a Designer
0x40	bDesignerData	Offset to Designer Data

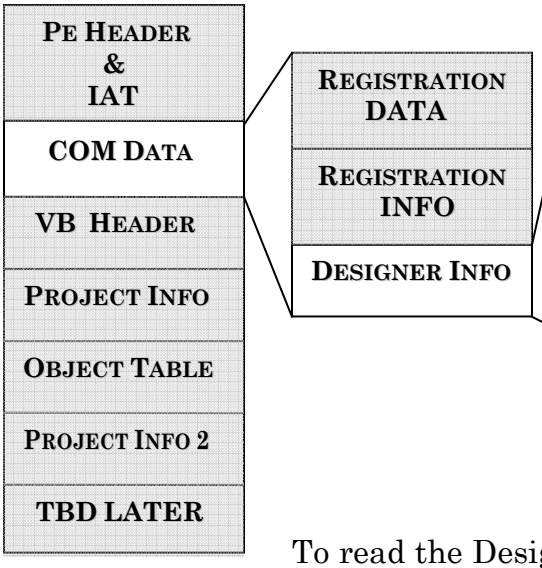
Please note that some of these values are only valid depending on the Object Type (see Flags below)

2.2 The Designer Info.

Structure name: NONE. Size: Variable.

If the Object happens to be a Designer (used for Add-Ins), then the tagRegInfo structure is augmented by the Designer Data structure, shown below:

NB: A BSTR contains the string length as a long, followed by the string itself.



Offset	*Name*	*Description*
0x0	uuidDesigner	CLSID of the Addin/Designer
0x10	cbStructSize	Total Size of the next fields.
0x14	bstrAddinRegKey	Registry Key of the Addin
VAR	bstrAddinName	Friendly Name of the Addin
VAR	bstrAddinDescription	Description of Addin
VAR	dwLoadBehaviour	CLSID of Object
VAR	bstrSatelliteDll	Satellite DLL, if specified
VAR	bstrAdditionalRegKey	Extra Registry Key, if specified
VAR	dwCommandLineSafe	Specifies a GUI-less Addin if 1.

To read the Designer Info structure, simply read the first long value at 0x14, which is the length of the Add-In Registry Key. Then add that number to the current offset, and you get the offset of Add-In Name's Length. Add that number to your new offset, and you get to the Add-In Description. Next up is the Load Behavior, a long value, followed by another length, this time the Satellite DLL's Name. If this is 0, it's the Additional Registry Key Name. If this is 0, then you arrive at dwCommandLineSafe.

Finally, here are the flags for RegData->ObjectType

Object Types

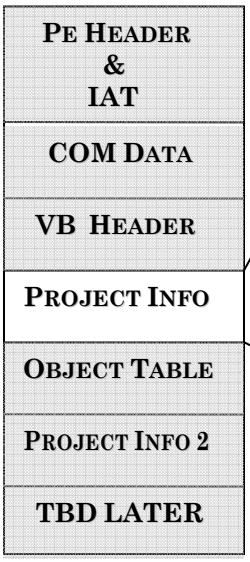
Value	*Name*	*Description*
0x2	Designer	A Visual Basic Designer for an Add-In
0x10	Class Module	A Visual Basic Class
0x20	User Control	A Visual Basic Active X User Control (OCX)
0x80	User Document	A Visual Basic User Document

nb: Other values may exist to define VB Objects, but they aren't used in this structure.

3. The Project Information

Structure name: none. Size: 0x23C bytes.

The Project Information structure is pointed by the VB Header. It contains user information about the project as well as critical information (such as a pointer to the Object Table). It is also heavily used for compilation statistics. Here's an at-a-glance view:



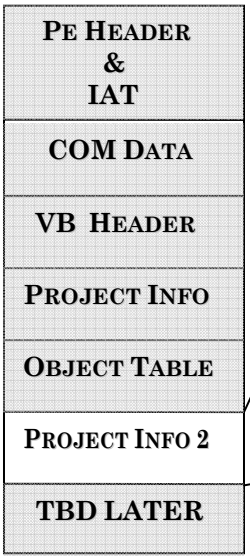
Offset	*Name*	*Description*
0x0	dwVersion	5.00 in Hex (0x1F4). Version.
0x4	lpObjectTable	Pointer to the Object Table
0x8	dwNull	Unused value after compilation.
0xC	lpCodeStart	Points to start of code. Unused.
0x10	lpCodeEnd	Points to end of code. Unused.
0x14	dwDataSize	Size of VB Object Structures. Unused.
0x18	lpThreadSpace	Pointer to Pointer to Thread Object.
0x1C	lpVbaSeh	Pointer to VBA Exception Handler
0x20	lpNativeCode	Pointer to .DATA section.
0x24	szPathInformation	Contains Path and ID string. < SP6
0x234	lpExternalTable	Pointer to External Table.
0x238	dwExternalCount	Objects in the External Table.

A great majority of these values are only used for compilation are leftovers of statistical data. These include the path information, code pointers, and data size.

4. The Secondary Project Information

Structure name: none. Size: 0x28 bytes.

This Secondary structure, pointed by the Object Table contains mostly data used when compiling the project. It does also however pave the way to the Form List (To be described later) and gives the elusive Help Context ID.

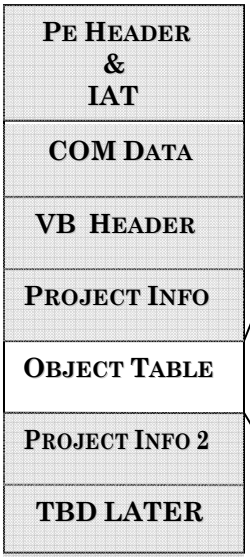


Offset	*Name*	*Description*
0x0	lpHeapLink	Unused after compilation, always 0.
0x4	lpObjectTable	Back-Pointer to the Object Table.
0x8	dwReserved	Always set to -1 after compiling. Unused
0xC	dwUnused	Not written or read in any case.
0x10	lpObjectList	Pointer to Object Descriptor Pointers.
0x14	dwUnused2	Not written or read in any case.
0x18	szProjectDescription	Pointer to Project Description
0x1C	szProjectHelpFile	Pointer to Project Help File
0x20	dwReserved2	Always set to -1 after compiling. Unused
0x24	dwHelpContextId	Help Context ID set in Project Settings.

5. The Object Table

Structure name: none. Size: 0x54 bytes.

The Object Table structure is pointed by the Project Info Structure. It contains points to the Object Array, as well as more repeated Project Data (presumably so it can be read quickly from here). Some values are also only used when running the project in memory (in the IDE). Here's an at-a-glance view:

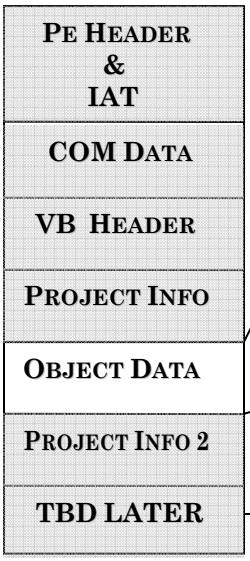


Offset	*Name*	*Description*
0x0	lpHeapLink	Unused after compilation, always 0.
0x4	lpExecProj	Pointer to VB Project Exec COM Object.
0x8	lpProjectInfo2	Secondary Project Information.
0xC	dwReserved	Always set to -1 after compiling. Unused
0x10	dwNull	Not used in compiled mode.
0x14	lpProjectObject	Pointer to in-memory Project Data.
0x18	uuidObject	GUID of the Object Table.
0x28	fCompileState	Internal flag used during compilation.
0x2A	dwTotalObjects	Total objects present in Project.
0x2C	dwCompiledObjects	Equal to above after compiling.
0x2E	dwObjectsInUse	Usually equal to above after compile.
0x30	lpObjectArray	Pointer to Object Descriptors
0x34	fIdeFlag	Flag/Pointer used in IDE only.
0x38	lpIdeData	Flag/Pointer used in IDE only.
0x3C	lpIdeData2	Flag/Pointer used in IDE only.
0x40	lpszProjectName	Pointer to Project Name.
0x44	dwLcid	LCID of Project.
0x48	dwLcid2	Alternate LCID of Project.
0x4C	lpIdeData3	Flag/Pointer used in IDE only.
0x50	dwIdentifier	Template Version of Structure.

6. The Private Object Descriptor

Structure name: none. Size: 0x40 bytes.

The Private Object Descriptor Table is pointed by an array defined in the Object List Pointer in the Secondary Project Information. The whole structure can be deleted after compilation. Here's an at-a-glance view:

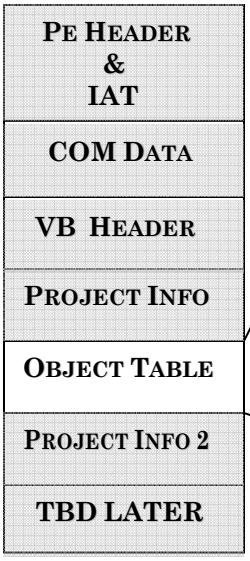


Offset	*Name*	*Description*
0x0	lpHeapLink	Unused after compilation, always 0.
0x4	lpObjectInfo	Pointer to the Object Info for this Object.
0x8	dwReserved	Always set to -1 after compiling.
0xC	dwIdeData[3]	Not valid after compilation.
0x18	lpObjectList	Points to the Parent Structure (Array)
0x1C	dwIdeData2	Not valid after compilation.
0x20	lpObjectList2[3]	Points to the Parent Structure (Array).
0x2C	dwIdeData3[3]	Not valid after compilation.
0x38	dwObjectType	Type of the Object described.
0x3C	dwIdentifier	Template Version of Structure.

7. The Public Object Descriptor

Structure name: none. Size: 0x30 bytes.

The Public Object Descriptor Table is pointed by the Array lpObjectArray in the Object Table. Each Object in the project will have its own. Unlike the private structure, this one is actually used by VB for a variety of tasks. Here's an at-a-glance view:

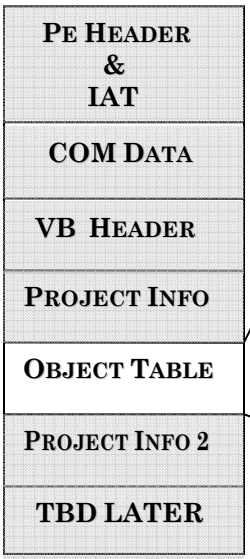


Offset	*Name*	*Description*
0x0	lpObjectInfo	Pointer to the Object Info for this Object.
0x4	dwReserved	Always set to -1 after compiling.
0x8	lpPublicBytes	Pointer to Public Variable Size integers.
0xC	lpStaticBytes	Pointer to Static Variable Size integers.
0x10	lpModulePublic	Pointer to Public Variables in DATA section
0x14	lpModuleStatic	Pointer to Static Variables in DATA section
0x18	lpszObjectName	Name of the Object.
0x1C	dwMethodCount	Number of Methods in Object.
0x20	lpMethodNames	If present, pointer to Method names array.
0x24	bStaticVars	Offset to where to copy Static Variables.
0x28	fObjectType	Flags defining the Object Type.
0x2C	dwNull	Not valid after compilation.

8. The Object Info

Structure name: none. Size: 0x38 bytes.

The Object Information structure defines an Object and provides various information to its methods and constants (in Pseudo Code). Here's an at-a-glance view:

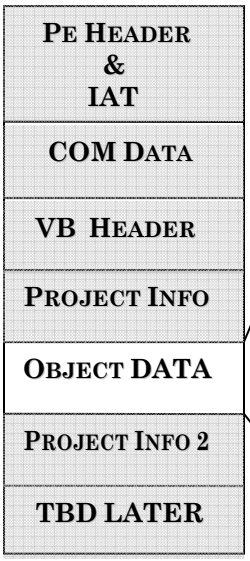


Offset	*Name*	*Description*
0x0	wRefCount	Always 1 after compilation.
0x2	wObjectIndex	Index of this Object.
0x4	lpObjectTable	Pointer to the Object Table
0x8	lpIdeData	Zero after compilation. Used in IDE only.
0xC	lpPrivateObject	Pointer to Private Object Descriptor.
0x10	dwReserved	Always -1 after compilation.
0x14	dwNull	Unused.
0x18	lpObject	Back-Pointer to Public Object Descriptor.
0x1C	lpProjectData	Pointer to in-memory Project Object.
0x20	wMethodCount	Number of Methods
0x22	wMethodCount2	Zeroed out after compilation. IDE only.
0x24	lpMethods	Pointer to Array of Methods.
0x28	wConstants	Number of Constants in Constant Pool.
0x2A	wMaxConstants	Constants to allocate in Constant Pool.
0x2C	lpIdeData2	Valid in IDE only.
0x30	lpIdeData3	Valid in IDE only.
0x34	lpConstants	Pointer to Constants Pool.

9. The Optional Object Info

Structure name: none. Size: 0x40 bytes.

The Optional Object Information structure, present only for COM Objects (anything but a Module) defines some GUIDs as well as other useful Form Information, and points to Controls. Here's an at-a-glance view:

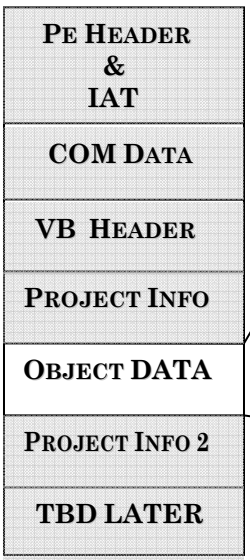


Offset	*Name*	*Description*
0x0	dwObjectGuids	How many GUIDs to Register. 2 = Designer
0x4	lpObjectGuid	Unique GUID of the Object *VERIFY*
0x8	dwNull	Unused.
0xC	lpuuidObjectTypes	Pointer to Array of Object Interface GUIDs
0x10	dwObjectTypeGuids	How many GUIDs in the Array above.
0x14	lpControls2	Usually the same as lpControls.
0x18	dwNull2	Unused.
0x1C	lpObjectGuid2	Pointer to Array of Object GUIDs.
0x20	dwControlCount	Number of Controls in array below.
0x24	lpControls	Pointer to Controls Array.
0x28	wEventCount	Number of Events in Event Array.
0x2A	wPCodeCount	Number of P-Codes used by this Object.
0x2C	bWInitializeEvent	Offset to Initialize Event from Event Table.
0x2E	bWTerminateEvent	Offset to Terminate Event in Event Table.
0x30	lpEvents	Pointer to Events Array.
0x34	lpBasicClassObject	Pointer to in-memory Class Objects.
0x38	dwNull3	Unused.
0x3C	lpIdeData	Only valid in IDE.

10. The Control Info

Structure name: none. Size: 0x28 bytes.

The Control Information Structure contains data about each control on the Form, and points to the Event Handler Table for this Control. Here's an at-a-glance view:



Offset	*Name*	*Description*
0x0	fControlType	Type of control.
0x4	wEventcount	Number of Event Handlers supported.
0x6	bWEventsOffset	Offset in to Memory struct to copy Events.
0x8	lpGuid	Pointer to GUID of this Control.
0xC	dwIndex	Index ID of this Control.
0x10	dwNull	Unused.
0x14	dwNull2	Unused.
0x18	lpEventTable	Pointer to Event Handler Table.
0x1C	lpIdeData	Valid in IDE only.
0x20	lpszName	Name of this Control.
0x24	dwIndexCopy	Secondary Index ID of this Control.