

Visual DialogScript

Visual DialogScript (VDS) is a programming language—and a visual development environment—used for the rapid development of simple Windows-based applications. Although less complex than larger and more familiar programming languages (such as Pascal, C++, or Visual Basic), VDS supports a broad range of useful commands and functions. A complete interactive development environment—consisting of an assortment of useful development tools—is also available to users of the language. Unlike many other programming languages and development environments, VDS provides its users with the ability to create and test a variety of application-types in only minutes, rather than hours. Developed by programmer Julian Moss—using the Delphi development environment—VDS was eventually acquired by S.A.D.E. s.a.r.l (a French software development company). Both 16- and 32-bit versions of VDS are available, although only the 32-bit version continues to be developed and updated.

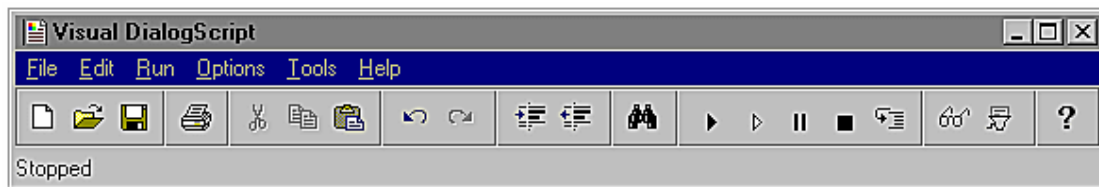
Commands and Functions

For the development of software applications (programs), VDS supports many useful *commands* (recognized instructions) and *functions* (recognized operations) that can be incorporated within *scripts* (the overall source-code of applications). Unlike the syntax of most other programming languages, these commands and functions are relatively easy to learn.

Interactive Development Environment

Visual DialogScript is equipped with a full *integrated development environment* (IDE) (see Fig. 1). Programmers use the IDE to develop applications that may or may not include *dialogs* (graphical-user interfaces known as "windows"). Although programmers must be familiar with the VDS language to manually enter program code (customized commands and functions) into scripts, many aspects of dialog development can be accomplished visually.

Figure 1. Main interface of the integrated development environment
(Visual DialogScript, 32-bit version).

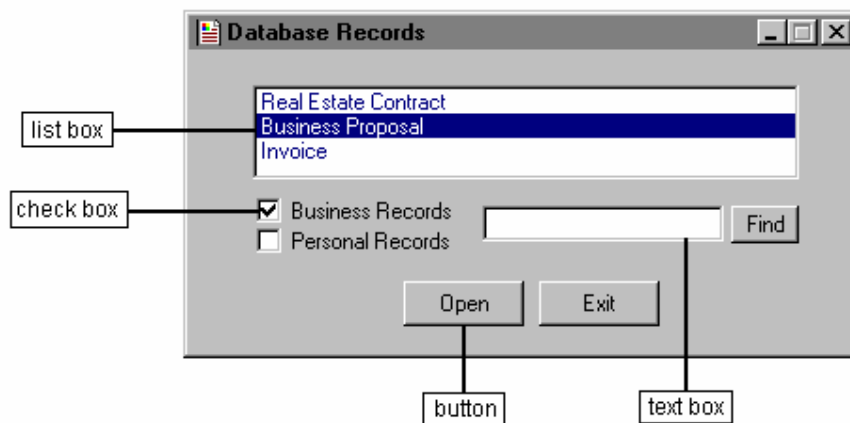


The IDE includes several useful development tools (all accessible from the main interface):

◆ **Dialog Designer/Editor**

Programmers use the *dialog designer/editor* to visually design dialogs that can be resized, as needed, using a mouse. Also, programmers can visually create application controls (such as *buttons*, *check boxes*, *text boxes*, *list boxes*, etc.) and then "drag and drop" them anywhere onto the dialogs. The dialog designer/editor automatically generates program code for the dialogs and inserts it into the dialogs' associated scripts. Because no manual programming is required, programmers save time. Figure 2 shows an example of a simple dialog for a small database application.

Figure 2. Simple Dialog



◆ **Dialog/Application Wizard**

Working in conjunction with the dialog designer/editor, the *dialog/application wizard* provides programmers with a step-by-step method for the creation of dialog-based scripts that serve as starting points for application development. Programmers may then modify and expand upon the scripts as desired.

◆ **Interactive Editor**

The *interactive editor* allows programmers to manually enter and edit program code within scripts. Unlike the dialog designer/editor, the interactive editor is not a visual-development tool.

◆ **Debugger**

Programmers use the included *debugger* to test scripts for errors.

◆ **Icon Designer/Editor**

The *icon designer/editor* allows programmers to create and edit icons (graphical images that users click to run programs) for VDS scripts.

◆ **Window Spy**

A simple utility, the *window spy* allows programmers to obtain information about other application windows. Programmers sometimes need this kind of information when developing certain types of applications (for example, applications that control other applications).

◆ **Compiler**

The *compiler* allows registered users (those who have purchased VDS) to compile their scripts into executable applications (programs that can run independently of the IDE—referred to as .EXE files). Registered users may then distribute their compiled applications royalty-free.

Uses for VDS

Programmers can use VDS to create a variety of application-types, including (but not limited to):

- simple utilities
- databases
- games
- system resource monitors
- multimedia applications, and
- front-ends (i.e. applications that control other programs—including programs running under the Disk Operating System (DOS) .

Available Versions of VDS

Created by programmer Julian Moss (using the Borland Delphi development environment), VDS was later sold to S.A.D.E s.a.r.l. (a French software development company). VDS is currently available in both 16- and 32-bit versions (16-bit applications are programs that run on Windows 3.x/higher, whereas 32-bit applications run on Windows 95/higher). Only the 32-bit version of VDS, however, continues to be developed and updated.

VDS v. 2.5 (16-Bit Version)

◆ Regular Edition

Available as shareware (software that you can try before you buy). Registered users can compile their scripts and distribute the resulting applications royalty-free.

◆ Personal Edition

Available as freeware (free software). This version of VDS is available to homeusers for noncommercial use. Users may run their scripts within the VDS integrated development environment, but the scripts cannot be compiled.

VDS v. 3.01 (32-Bit Version)

Considered the VDS standard, this version is available as shareware. Upon registration, users can compile their scripts and distribute the resulting applications royalty-free.