



ASCENT[®]

CENTER FOR TECHNICAL KNOWLEDGE

CATIA V5-6R2018

Visual Basic Automation

Learning Guide

1st Edition

Sample provided by ASCENT for review only
All copying and reuse strictly forbidden.

ASCENT - Center for Technical Knowledge®
CATIA V5-6R2018
Visual Basic Automation
1st Edition

Prepared and produced by:

ASCENT Center for Technical Knowledge
630 Peter Jefferson Parkway, Suite 175
Charlottesville, VA 22911

866-527-2368
www.ASCENTed.com

Lead Contributor: Scott Hendren



ASCENT - Center for Technical Knowledge is a division of Rand Worldwide, Inc., providing custom developed knowledge products and services for leading engineering software applications. ASCENT is focused on specializing in the creation of education programs that incorporate the best of classroom learning and technology-based training offerings.

We welcome any comments you may have regarding this guide, or any of our products. To contact us please email: feedback@ASCENTed.com.

© ASCENT - Center for Technical Knowledge, 2020

All rights reserved. No part of this guide may be reproduced in any form by any photographic, electronic, mechanical or other means or used in any information storage and retrieval system without the written permission of ASCENT, a division of Rand Worldwide, Inc.

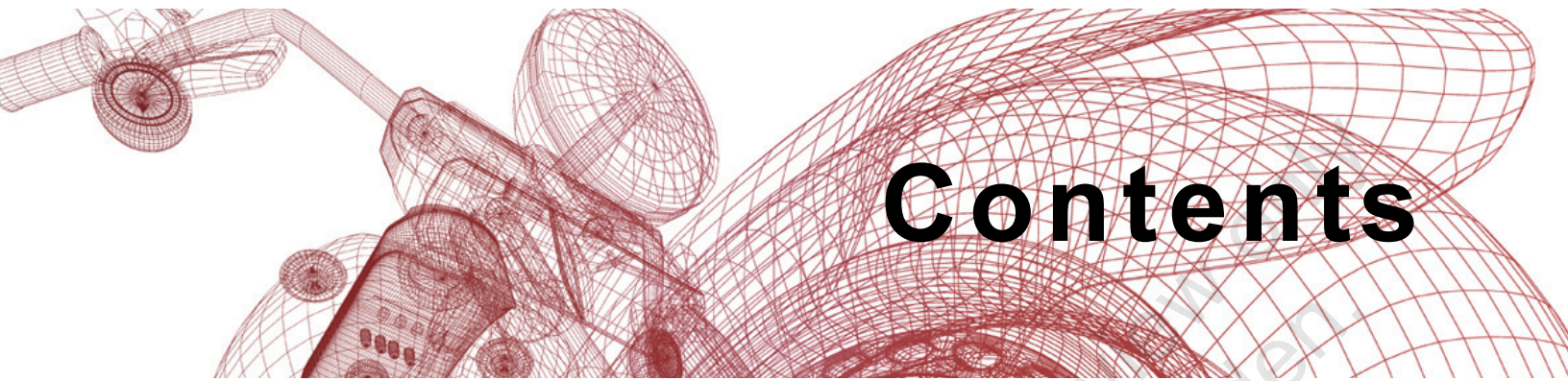
CATIA and ENOVIA are registered trademarks of Dassault Systèmes.

All other brand names, product names, or trademarks belong to their respective holders.

General Disclaimer:

Notwithstanding any language to the contrary, nothing contained herein constitutes nor is intended to constitute an offer, inducement, promise, or contract of any kind. The data contained herein is for informational purposes only and is not represented to be error free. ASCENT, its agents and employees, expressly disclaim any liability for any damages, losses or other expenses arising in connection with the use of its materials or in connection with any failure of performance, error, omission even if ASCENT, or its representatives, are advised of the possibility of such damages, losses or other expenses. No consequential damages can be sought against ASCENT or Rand Worldwide, Inc. for the use of these materials by any third parties or for any direct or indirect result of that use.

The information contained herein is intended to be of general interest to you and is provided "as is", and it does not address the circumstances of any particular individual or entity. Nothing herein constitutes professional advice, nor does it constitute a comprehensive or complete statement of the issues discussed thereto. ASCENT does not warrant that the document or information will be error free or will meet any particular criteria of performance or quality. In particular (but without limitation) information may be rendered inaccurate by changes made to the subject of the materials (i.e. applicable software). Rand Worldwide, Inc. specifically disclaims any warranty, either expressed or implied, including the warranty of fitness for a particular purpose.



Contents

Preface	vii
In this Guide	ix
Practice Files	xi
Chapter 1: Fundamentals.....	1-1
1.1 Visual Basic Evolution.....	1-2
CATIA Programming Platforms.....	1-2
1.2 VB6 Platform.....	1-3
Event-driven Programming	1-3
IDE (Integrated Development Environment)	1-4
GUI (Graphical User Interface)	1-5
COM (Component Object Model).....	1-5
1.3 VBA.....	1-7
1.4 VBScript and CATScript	1-8
1.5 VB.NET	1-9
1.6 In-process and Out-of-process Applications	1-10
In-process Applications	1-10
Out-of-process Applications	1-10
1.7 Registering CATIA Libraries	1-11
Chapter 2: CATIA Object Model	2-1
2.1 Objects Oriented Programming.....	2-2
Objects	2-2
Classes and Instances	2-3
CATIA Objects and Classes.....	2-4
Example of an Object in CATIA	2-6
Object Instantiation and Referencing in CATIA.....	2-6
2.2 Inheritance	2-8
Example of Inheritance in CATIA	2-8

2.3 Aggregation	2-10
Example of Aggregation in CATIA	2-10
2.4 Abstract Class	2-12
Example of Abstract Class in CATIA.....	2-13
2.5 Collections	2-14
2.6 CATIA Object Diagram.....	2-16
2.7 CATIA Automation Documentation	2-17
Practice 2a Navigating Automation Documentation.....	2-19
Chapter 3: Introduction to Macros	3-1
3.1 Introduction	3-2
3.2 Macro Libraries	3-3
3.3 Recording Macros.....	3-4
3.4 Viewing or Editing Macros	3-5
Interpreting Macro Recordings	3-6
Replaying a Macro	3-7
Executing a Macro on CATIA startup.....	3-7
Macros with Parameters	3-8
3.5 Assigning a Macro to a Toolbar.....	3-9
Practice 3a Square Pad with Parameters.....	3-10
Practice 3b Series of Round Pads.....	3-16
Chapter 4: Introduction to VBA.....	4-1
4.1 Overview	4-2
4.2 Event-driven Programming	4-3
4.3 Forms and Controls	4-4
4.4 Modules.....	4-5
4.5 Early Versus Late Binding.....	4-6
4.6 Type Libraries.....	4-7
4.7 Object Browser.....	4-8
4.8 Programming Tips	4-9
Explicit vs Implicit Variable Declaration.....	4-9
Running and Debugging	4-10
Starting a User Form from CATIA.....	4-11
4.9 Error Handling	4-12

4.10 Restricted CATIA Interfaces	4-13
Restricted Interface Error Example	4-14
Handling Restricted Interfaces	4-15
Practice 4a Simple User Form with Controls	4-17
Chapter 5: CATIA Infrastructure Automation	5-1
5.1 CATIA Root Object Diagram	5-2
5.2 Documents Collection	5-3
5.3 Windows Collection	5-4
5.4 SystemConfiguration Object	5-5
Practice 5a Creating and Saving a Part	5-6
Practice 5b Opening a Product	5-8
Chapter 6: Part Infrastructure Automation	6-1
6.1 Part Document Object Diagram	6-2
6.2 Bodies Collection	6-4
6.3 HybridBodies Collection	6-5
Practice 6a Working with Part Infrastructure	6-6
Chapter 7: Part Design Automation	7-1
7.1 Part Design Object Diagram	7-2
Types of Shape Objects.....	7-3
7.2 Sketch Object Diagram	7-4
7.3 Creating Sketches	7-5
Creating Sketch Geometry	7-5
Creating Sketch Constraints	7-6
7.4 Creating Solid Features	7-7
7.5 Reference Object	7-8
7.6 In Work Object and Part Update	7-9
Practice 7a Working with Sketcher and Part Design	7-10
Chapter 8: Shape Design Automation	8-1
8.1 Shape Design Object Diagram	8-2
Types of Hybrid Shape Objects	8-3
8.2 HybridBody Object	8-4
8.3 HybridShapeFactory Object	8-5
8.4 Creating Shape Features	8-6
Practice 8a Working with Shape Design	8-7

Chapter 9: Assembly Design Automation	9-1
9.1 Product Document Object Diagram	9-2
9.2 Product Properties	9-3
9.3 Reference Product	9-4
Building Product Structure	9-5
9.4 Positioning Assembly Components.....	9-6
Transformation Matrix	9-7
9.5 Assembly Constraints	9-8
Analyze Object and Bill of Materials.....	9-9
Practice 9a Working with Assembly Design.....	9-10
Chapter 10: Drafting Automation	10-1
10.1 Drawing Document Object Diagram.....	10-2
10.2 Sheets and Views	10-3
Sheets	10-3
Views.....	10-4
10.3 Drawing View Object Diagram	10-5
10.4 GenerativeBehaviour Object.....	10-6
10.5 Creating Generative Views.....	10-7
View Projection Plane and Direction	10-8
View Positioning.....	10-9
10.6 Creating Dimensions	10-10
10.7 Creating Annotations.....	10-11
Drawing Text Object.....	10-12
Drawing Table Object.....	10-13
10.8 Creating Non-Generative Geometry	10-15
10.9 Creating Frame and Title Block	10-16
Frame and Title Block Macros	10-16
Creating Frame and Title Block Macro.....	10-17
Practice 10a Creating Drawing Views	10-18
Practice 10b Creating Frame and Title Block.....	10-24
Chapter 11: Working with Graphic Properties	11-1
11.1 Selection Object.....	11-2
11.2 Working with Selections.....	11-3
Adding Elements to Selection	11-3
11.3 Modifying Graphic Properties	11-5
Practice 11a Working with Graphic Properties	11-6

Chapter 12: Interactive Selections	12-1
12.1 Interactive Selection Options.....	12-2
12.2 Selection Parsing Methods	12-3
Item Method	12-3
FindObject Method.....	12-4
12.3 Interactive Methods	12-5
SelectElement* Methods.....	12-5
Using SelectElement Methods	12-5
IndicateOrSelectElement* Methods	12-6
Practice 12a Parsing Pre-Selected Elements	12-7
Practice 12b Working with SelectElement2	12-10
Chapter 13: Cut, Copy, Paste, and Delete	13-1
13.1 Deleting CATIA Objects	13-2
13.2 Copying and Pasting CATIA Objects	13-3
13.3 Paste Options	13-4
Practice 13a Copying and Pasting between Parts	13-5
Chapter 14: Parameters and Formulas.....	14-1
14.1 Parameters and Relations Collections.....	14-2
14.2 Parameters Object Model	14-3
14.3 Creating Parameters	14-4
14.4 Valuating Parameters	14-5
14.5 Working with Units.....	14-6
14.6 Relations Object Model	14-7
14.7 Creating Relations	14-8
Practice 14a Measuring Spline Length using Formula.....	14-9
Chapter 15: Communication with Microsoft Office	15-1
15.1 Microsoft Automation	15-2
15.2 Microsoft Office Object Model	15-3
15.3 Registering Libraries	15-4
15.4 GetObject and CreateObject Methods	15-5
Handling Errors when Contacting Server.....	15-5
Practice 15a Exporting CATIA Data to Excel.....	15-7
Practice 15b Exporting Excel Data to CATIA.....	15-11

Appendix A: Completed Practices	A-1
Practice 1a Practice A1.....	A-2
Practice 1b Practice A2.....	A-6
Practice 1c Practice A3.....	A-10
Practice 1d Practice A4.....	A-11
Practice 1e Practice A5.....	A-13
Practice 1f Practice A6.....	A-15
Practice 1g Practice A7.....	A-17
Practice 1h Practice A8.....	A-19
Practice 1i Practice A9.....	A-23
Practice 1j Practice A10.....	A-27
Practice 1k Practice A11.....	A-31
Practice 1l Practice A12.....	A-35
Practice 1m Practice A13.....	A-37
Practice 1n Practice A14.....	A-38
Practice 1o Practice A15.....	A-40
Practice 1p Practice A16.....	A-42
Practice 1q Practice A17.....	A-44
Practice 1r Practice A18	A-46

Sample provided by ASCENT for review only
All copying and reuse strictly forbidden



Preface

The *CATIA V5-6R2018: Visual Basic Automation* learning guide provides you with a good understanding of the different ways to automate tasks using CATIA macros and Visual Basic programming. Using hands-on practices, you will use VB programming to work with parts, assemblies, drawings, selections, parameters and formulas, graphic properties, and to exchange data with Microsoft Excel.

Topics Covered:

- CATIA V5 Object Model
- Creating Part Design and Shape Design features
- Working with Product Structure and Assembly Design
- Scripting Drawing Views, Frames, and Title Blocks
- Deleting, Cutting, Copying, Pasting CATIA objects
- Interactive Selections
- Communication with MS Office

Note on Software Setup

This learning guide assumes a standard installation of the software using the default preferences during installation. Lectures and practices use the standard software templates and default options for the Content Libraries.

Lead Contributor: Scott Hendren

Scott Hendren has been a trainer and curriculum developer in the PLM industry for over 20 years, with experience on multiple CAD systems, including Pro/ENGINEER, Creo Parametric, and CATIA. Trained in Instructional Design, Scott uses his skills to develop instructor-led and web-based training products.

Scott has held training and development positions with several high profile PLM companies, and has been with the ASCENT team since 2013.

Scott holds a Bachelor of Mechanical Engineering Degree as well as a Bachelor of Science in Mathematics from Dalhousie University, Nova Scotia, Canada.

Scott Hendren has been the Lead Contributor for *CATIA: Visual Basic Automation* since 2013.

Sample provided by ASCENT for review only.
All copying and reuse strictly forbidden.



In this Guide


The following images highlight some of the features that can be found in this guide.

Link to the practice files

Practice Files

To download the practice files for this student guide, use the following steps:

1. Type the URL shown below into the address bar of your Internet browser. The URL must be typed **exactly as shown**. If you are using an ASCENT ebook, you can click on the link to download the file.



2. Press <Enter> to download the .ZIP file that contains the Practice Files.
3. Once the download is complete, unzip the file to a local folder. The unzipped file contains an .EXE file.
4. Double-click on the .EXE file and follow the instructions to automatically install the Practice Files on the C:\ drive of your computer.

Do not change the location in which the Practice Files folder is installed. Doing so can cause errors when completing the practices in this student guide.

<http://www.ASCENTed.com/getfile?id=xxxxxxx>

Stay Informed!
Interested in receiving information about upcoming promotional offers, educational events, invitations to complimentary webcasts, and discounts? If so, please visit: www.ASCENTed.com/updates/

Help us improve our product by completing the following survey:
www.ASCENTed.com/feedback
You can also contact us at: feedback@ASCENTed.com

Practice Files

The Practice Files page tells you how to download and install the practice files that are provided with this guide.

Learning Objectives for the chapter

Chapter 1

Getting Started

In this chapter you learn how to start the AutoCAD® software, become familiar with the basic layout of the AutoCAD screen, how to access commands, use your pointing device, and understand the AutoCAD Cartesian workspace. You also learn how to open an existing drawing, view a drawing by zooming and panning, and save your work in the AutoCAD software.

Learning Objectives in this Chapter

- Launch the AutoCAD software and complete a basic initial setup of the drawing environment.
- Identify the basic layout and features of AutoCAD interface including the Ribbon, Drawing Window, and Application Menu.
- Locate commands and launch them using the Ribbon, shortcut menus, Application Menu, and Quick Access Toolbar.
- Locate points in the AutoCAD Cartesian workspace.
- Open and close existing drawings and navigate to file locations.
- Move around a drawing using the mouse, the Zoom and Pan commands, and the Navigation Bar.
- Save drawings in various formats and set the automatic save options using the Save commands.

Chapters

Each chapter begins with a brief introduction and a list of the chapter's Learning Objectives.

Side notes

Side notes are hints or additional information for the current topic.

Practice Objectives

Getting Started

Starting Commands

You can also click (Customize) to display the Input Settings for the AutoComplete feature.

If you need to stop a command, press <Esc> to cancel. You might need to press <Esc> more than once.

1.3 Working with Commands

The main way to access commands in the AutoCAD software is to use the Ribbon. Several of the file commands are available in the Quick Access Toolbar or in the Application Menu. Some commands are available in the Status Bar or through shortcut menus. There are additional access methods, such as Tool Palettes. The names of all of the commands can also be typed in the Command Line. A table is included to help you to identify the various methods of accessing the commands.

When typing the name of a command in either the Command Line or Dynamic Input, the **AutoComplete** option automatically completes the entry when you pause as you type. It also supports mid-string search by displaying all of the commands that contain the word that you typed, as shown in Figure 1-12. You can then scroll through the list and select a command.




Figure 1-12

To set specific options for the **AutoComplete** feature, right-click on the Command Line, expand Input Settings, and select from the various options, such as the ability to search for system variables or to set the delay response time, as shown in Figure 1-13.

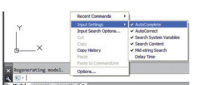


Figure 1-13

As you work in the AutoCAD software, the software prompts you for the information that is required to complete each command. These prompts are displayed in the drawing window near the cursor and in the Command Line. It is crucial that you read the command prompts as you work, as shown in Figure 1-14.

© 2015, ASCENT - Center for Technical Knowledge®
1-9

Instructional Content

Each chapter is split into a series of sections of instructional content on specific topics. These lectures include the descriptions, step-by-step procedures, figures, hints, and information you need to achieve the chapter's Learning Objectives.

Getting Started

Practice 1c

Estimated time for completion: under 5 minutes.

Saving a Drawing File

Practice Objectives

- Open and save a drawing.
- Modify the Automatic Saves option.

In this practice you will open a drawing, save it, and modify the **Automatic saves** option, as shown in Figure 1-51.

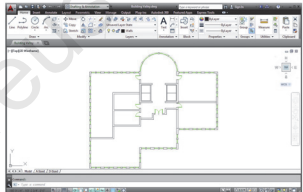


Figure 1-51

1. Open **Building Valley-M.dwg** from your class files folder.
2. In the Quick Access Toolbar, click (Save). In the Command Line, **QSAVE** displays indicating that the AutoCAD software has performed a quick save.
3. In the Application Menu, click to open the Options dialog box.
4. In the **Open and Save** tab, change the time for **Automatic save** to 15 minutes.

Practices

Practices enable you to use the software to perform a hands-on review of a topic.

Some practices require you to use prepared practice files, which can be downloaded from the link found on the Practice Files page.



Chapter

1

Fundamentals

Learning Topics in this Chapter

- Visual Basic Evolution
- VBA
- VBScript and CATScript
- VB.NET
- In-process and Out-of-process Applications
- Registering CATIA Libraries

1.1 Visual Basic Evolution

Each incarnation of Visual Basic has had a substantially different syntax, as shown in Figure 1–1

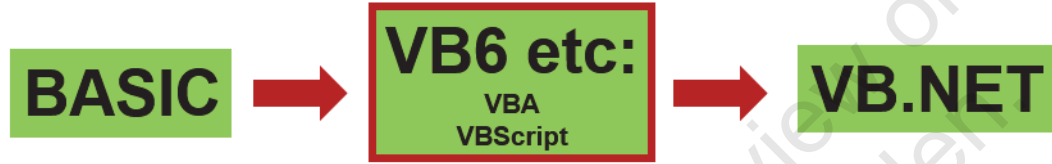


Figure 1–1

- The B.A.S.I.C. (Beginner's All-purpose Symbolic Instruction Code) is a programming language that was originally designed in 1963.
- Visual Basic is Microsoft's dialect of BASIC including controls, events, and IDE. It was very popular in 1980s and 1990s. CATIA V5 Automation was originally designed for VBScript, VBA, and VB6.
- VB.NET is a completely reworked version of VB. It replaces VB6, which Microsoft no longer officially supports. However, it does not replace VBA or VBScript, both of which are still supported by Microsoft. Therefore, VB.NET does not make VBA or VBScript scripts or macros obsolete.

**CATIA
Programming
Platforms**

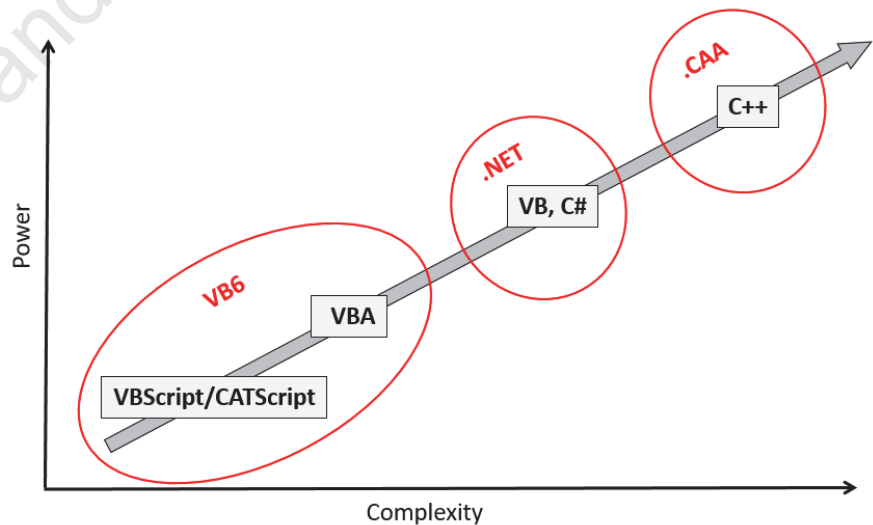


Figure 1–2

Event-driven Programming

1.2 VB6 Platform

Visual Basic 6 (VB6) is an event-driven programming platform and an associated integrated development environment (IDE) for building graphical user interfaces (GUI) in the Microsoft Component Object Model (COM) programming model.

In event-driven programming, the flow of the program is determined by user actions (i.e., mouse clicks, key presses, etc.), or messages from other programs, as shown in Figure 1–3.

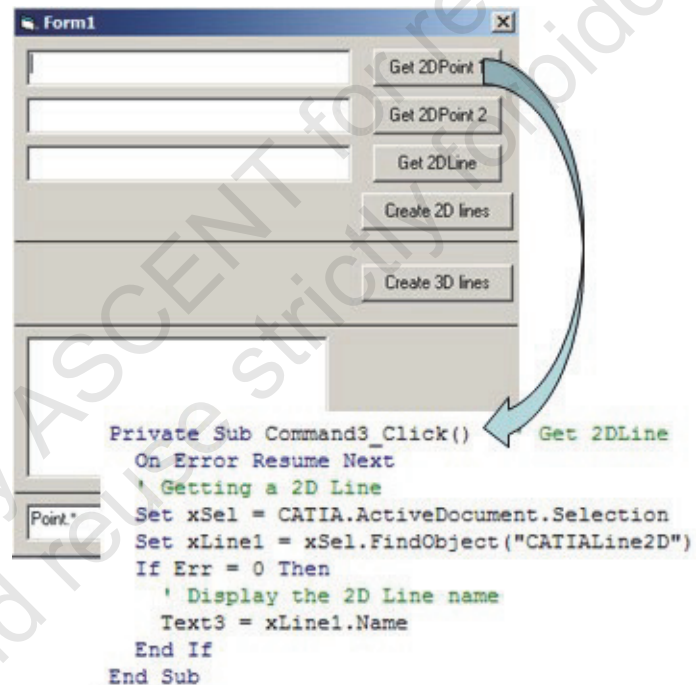


Figure 1–3

In contrast, in sequential programming the flow of the program is determined by the programmer using conditional statements (If...Else If ...Else, etc.), as shown in Figure 1–4.

```

Sub CATDrw_Create()

'MsgBox "Creating FTBs..."

'--- Init Drawing-scope public variables
On Error Resume Next
FTBIA_InitDocument
If Err.Number <> 0 Then Exit Sub
On Error Goto 0

'--- Set Drawing parameters
FTBIA_SetDrwParams

Dim DrwSheetCur As DrawingSheet
Set DrwSheetCur = poDrwSheets.ActiveSheet ' memorize last active sheet

'--- Loop over all sheets
For piSheet = 1 To pnSheet
'MsgBox "Creating FTB: Sheet " & piSheet & " of " & pnSheet
Set poDrwSheet = poDrwSheets.Item(piSheet)
poDrwSheet.Activate
FTBIA_CreateSheet
Next

DrwSheetCur.Activate ' return to the last active sheet

End Sub
    
```

Figure 1–4

IDE (Integrated Development Environment)

IDE (Integrated Development Environment) is a type of computer software that assists computer programmers in developing applications. It includes a source code editor, compiler or interpreter, build-automation tools, debugger, and object browser, as shown in Figure 1–5.

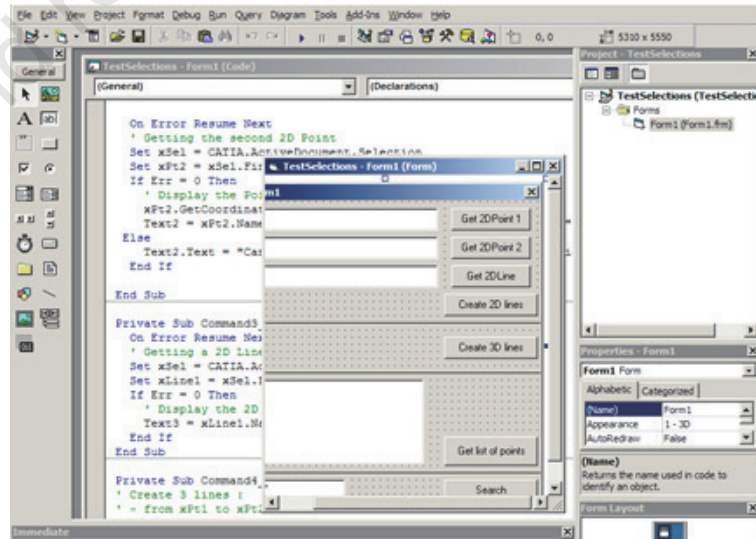


Figure 1–5

GUI (Graphical User Interface)

GUI (Graphical User Interface) is a type of user interface that enables you to interact with the computer using graphical elements (i.e., toolbars, icons, visual indicators, texts, labels, etc.), which represent the available information and actions. The actions are usually performed through direct manipulation of the graphical elements. CATIA software is GUI-enabled, as shown in Figure 1–6.

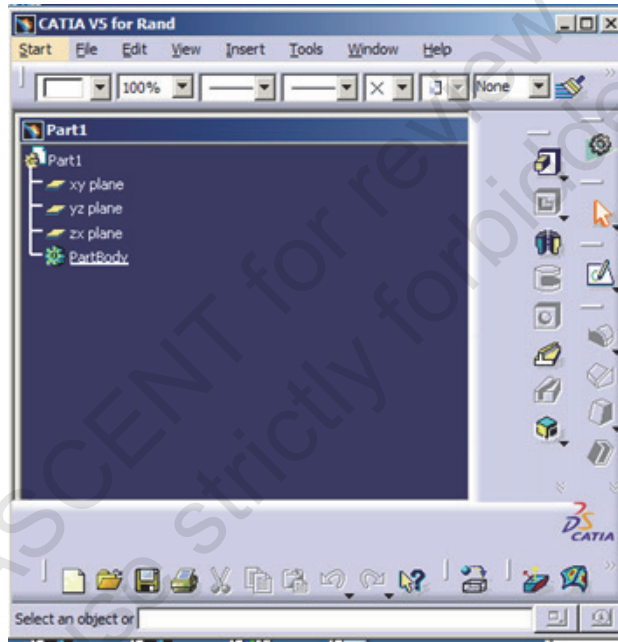


Figure 1–6

COM (Component Object Model)

COM (Component Object Model), shown in Figure 1–7, is a Microsoft technology designed to enable communication between software components written in any programming language. The term COM is often used as an umbrella term that encompasses OLE, Automation, ActiveX, COM+, and DCOM technologies.

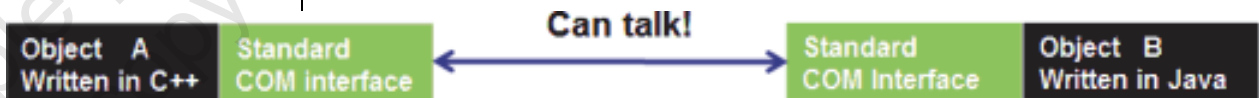


Figure 1–7

The essence of COM is a language-neutral way of implementing software components. COM standard permits the reuse of components with no knowledge of their internal implementation because it forces software programmers to provide standard binary interfaces that are separate from the implementation.

CATIA V5 is COM-enabled software. Codes for COM objects or components are stored in DLL libraries and registered in the Windows registry. Therefore, they can be called, initiated, or created at any time.

A software component can work as either a client, or a server. For example, if object A calls object B, object A is then the client and object B is the server (the one that provides services to the client). The client does not have to know the functions or methods (i.e., interface) that the server contains in advance. COM provides for a way to discover the server's interface when the program executes. This is called *late binding*. If the client is made aware of the server's interface before the program begins executing, this is called early binding.

Sample provided by ASCENT for educational use only.
All copying and reuse strictly forbidden.

1.3 VBA

The VB6 environment can be hosted in another application, such as Excel, Word, CATIA, etc. and is event-driven and GUI-oriented. It has full IDE (i.e., it contains an editor, debugger, type checker, object browser, etc.) as shown in Figure 1–8. However, it cannot run without the host application running. The source code can only be interpreted (i.e., executed line-by-line), and might be slower than a compiled VB application.

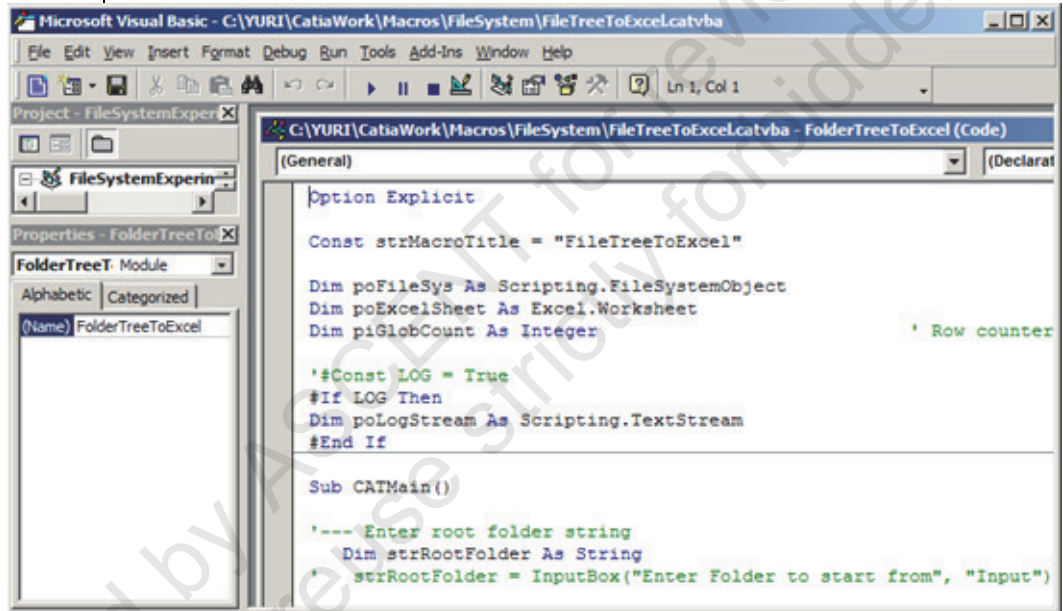
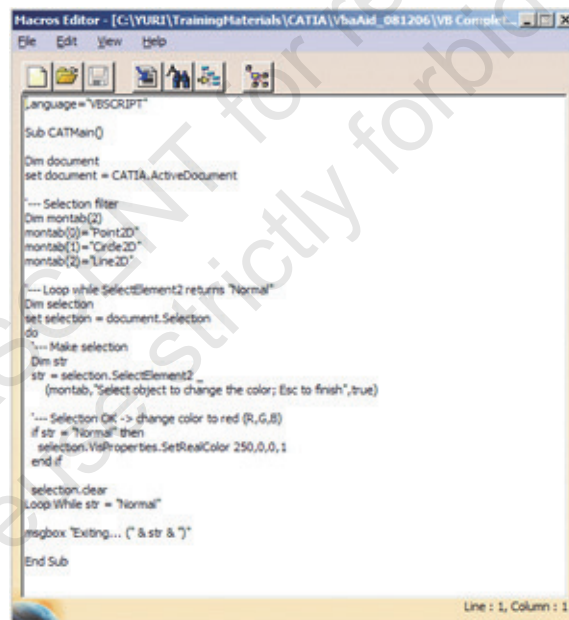


Figure 1–8

1.4 VBScript and CATScript

VBScript is a subset of full VB syntax (some programming features are omitted). It is designed to run in Web applications (i.e., Internet Explorer, etc.) and is an extremely simple and limited tool. It does not contain IDE, debugging etc., and regular text editors are used for coding. It is non-GUI-oriented and only a couple of very simple interactive dialog boxes are available. It is non-event-driven and its code represents a flow with a sequence of actions that is determined by the programmer, as shown in Figure 1–9.



```
Macros Editor - [C:\YURI\TrainingMaterials\CATIA\vb\Add_081206\VB Compil...
File Edit View Help
Language = VBSCRIPT
Sub CATMain()
Dim document
set document = CATIA.ActiveDocument

'--- Selection filter
Dim montab(2)
montab(0) = "Point2D"
montab(1) = "Circle2D"
montab(2) = "Line2D"

'--- Loop while SelectElement2 returns "Normal"
Dim selection
set selection = document.Selection
do
'--- Make selection
Dim str
str = selection.SelectElement2 _
(montab, "Select object to change the color; Esc to finish", true)

'--- Selection OK -> change color to red (R,G,B)
if str = "Normal" then
selection.VisProperties.SetRealColor 250,0,0,1
end if

selection.clear
Loop While str = "Normal"
msgbox "Exiting... (" & str & ")"
End Sub
Line : 1, Column : 1
```

Figure 1–9

CATScript is a portable version of VBScript. CATScript macros can run on Unix.

1.5 VB.NET

VB.NET is the latest incarnation of the popular Microsoft programming environment. It is also event-driven, has an integrated development environment (IDE), and is used for building graphical user interfaces (GUI). However, it is not based on COM (although it can call COM objects) and the syntax is quite different from VB6.

The following difficulties are normally encountered when switching from VB6 to VB.NET:

- New syntax
- New IDE
- New GUI controls
- New InstallShield

When using VB.NET for CATIA automation, CATIA V5 is still a COM application. There is no change to how CATIA objects are called and used.

1.6 In-process and Out-of-process Applications

There are two ways in which a VB application or macro can communicate with CATIA: as an in-process or an out-of-process application.

In-process Applications

The VB application runs within the CATIA process in the computer memory. No inter-process communication is involved. CATIA freezes while the VB application is running. The VB application's allocation memory is wiped out after each run. Therefore, passing data between two subsequent runs is not possible.

- Use the **Tools>Macros** interface (VBScript, CATScript, or VBA).

Out-of-process Applications

The VB application runs in its own process in the computer memory. It communicates with CATIA using inter-process messaging. CATIA is fully active while the VB application is running. For example, these applications could be VB.NET or C# programs, VBA macros in Excel, Word, etc., or Windows Explorer (i.e., embedded VBScript or Java scripts).

1.7 Registering CATIA Libraries

CATIA libraries are automatically registered in the Windows registry when CATIA is installed. The registration enables you to double-click on a CATPart or CATProduct to automatically start CATIA. A VB application or macro can also locate and communicate with CATIA objects.

Since CATIA libraries might change from release to release, a VB application that is developed or compiled with one CATIA release might or might not run correctly with another CATIA release. For example, a VB application developed for V5R21 might not work correctly with V5-6R2012, etc. Therefore, it is recommended that you have the correct CATIA release registered before the VB application is used.

If several CATIA releases are installed on a computer, registering a specific release can be done by executing the following command from the Command Prompt:

```
<catia_executable> /regserver -env <catia_envir_file>  
-direnv <catia_envir_directory>
```

For example, the following command registers CATIA V5-6R2018 libraries on a Windows 10 64-bit system:

```
"C:\Program Files\Dassault Systemes\B28\win_b64\code\  
bin\CNEXT.exe" /regserver -env CATIA.V5-6R2018.B28  
-direnv "C:\ProgramData\DassaultSystemes\CATEnv"
```