

CA-Visual Objects

Effective Programming

Bearbeitet von
Ivo Wessel, Gunnar Bless

1. Auflage 2003. Buch. 848 S. Hardcover
ISBN 978 3 446 22205 2
Format (B x L): 18,1 x 24,5 cm
Gewicht: 1647 g

Zu [Leseprobe](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](#) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

CARL HANSER VERLAG

Ivo Wessel, Gunnar Bless

CA-Visual Objects
Effective Programming

3-446-22205-7

www.hanser.de



Contents

Installation, Configuration & Tools.....	1
1 Installation, Configuration & Tools.....	3
1.1 This chapter ...	3
1.2 Outside CA-Visual Objects.....	3
1.2.1 Environment variables.....	4
1.2.2 Neat directory structure	4
1.3 Inside CA-Visual Objects.....	4
1.3.1 VO settings “View Options”.....	5
1.3.2 VO settings “File Setup”	5
1.3.3 Further VO settings.....	6
1.3.4 Painter file CAVOWED.INF	6
1.3.5 Registry entries	7
1.4 Strategy	8
1.4.1 What means “Programming with VO ...?”.....	8
1.4.2 Class Designer vs. Class Consumer.....	9
1.4.3 What is an application ...?.....	9
1.4.4 A bit of necessary theory: Object Oriented Programming	9
1.4.5 Event driven programming	10
1.4.6 OOP & Events	12
1.5 Programming of elements in three phases.....	12
1.5.1 1. Phase: Designing an element.....	12
1.5.2 2nd Phase: Implementation of necessary properties	12
1.5.3 3rd Phase: Using an element.....	13
1.6 Naming conventions and division in modules.....	13
1.6.1 Separating generated from manual written code	13
1.6.2 Module name = class name.....	14
1.6.3 Tokens, habits and other rules.....	14
1.7 Repository Explorer	14
1.7.1 Modules.....	14
1.7.2 Module names.....	14
1.7.3 Keyboard and mouse functions.....	16
1.8 Source code editor.....	17
1.8.1 Instance variables visibilities	17
1.8.2 Instance variables prefixes.....	17
1.8.3 Sorting of ASCII entities	18

1.8.4	Strong typing.....	19
1.8.5	ONLYEARLY pragma.....	19
1.8.6	Hungarian Notation.....	20
1.8.7	Scoping.....	21
1.8.8	Internal system constants.....	22
1.8.9	“Matching tokens”.....	23
1.8.10	Macros.....	23
1.8.11	Keyboard and mouse handling.....	28
1.9	Menu editor.....	29
1.9.1	Useful event names.....	29
1.9.2	Creating toolbars dynamically.....	29
1.9.3	Keyboard and mouse handling.....	29
1.10	Window editor.....	30
1.10.1	Control names.....	30
1.10.2	Spacing.....	31
1.10.3	Keyboard and mouse handling.....	33
	Windows & Controls.....	35
2	Windows & Controls.....	37
2.1	This chapter.....	37
2.2	Praeludium: The new spring colors of the season.....	37
2.2.1	A breath of Windows XP – also without XP.....	37
2.2.2	Color assignments for windows and Controls.....	38
2.2.3	“Global” brush object in its own App class.....	38
2.2.4	Definition of a “global” Assign interface in the CAVOWED.INF.....	39
2.2.5	Two new Assign statements.....	40
2.2.6	TabControl background.....	41
2.3	Resizable windows with control adjustment.....	41
2.3.1	Overview.....	42
2.3.2	Adjustment variants.....	43
2.3.3	Differences of SplitWindows.....	43
2.3.4	Problems & work minimisations.....	43
2.3.5	New! Multiple Inheritance for VO!.....	44
2.3.6	Conceptional basis.....	44
2.3.7	A new base window class.....	44
2.3.8	...respectively naturally two to three.....	44
2.3.9	The class dlgResizable.....	45
2.3.10	Constructor of the class dlgResizable.....	45
2.3.11	Clear the control arrays.....	46
2.3.12	Changing the window size.....	46
2.3.13	The function LockWindowUpdate ().....	48
2.3.14	In a loop through the controls.....	48
2.3.15	Four methods – for all control classes...?.....	48
2.3.16	Registration of dynamic controls.....	50
2.3.17	First (suboptimal) possibility: Manual work – only an attempt.....	50

2.3.18	New Assign interface for the class Control.....	51
2.3.19	Second (ditto) possibility: Ditto – also still not perfect.....	52
2.3.20	Better: Leave it to the Window editor	52
2.3.21	Adjustment of the Window editor	52
2.3.22	The file CAVOWED.INF.....	53
2.3.23	We are almost there – like Columbo: “There is still one little thing I forgot...”.....	54
2.3.24	I sew solve: A completely good solution	54
2.3.25	A new Assign interface in the Window editor.....	56
2.3.26	Description of the implementation of the three sections	56
2.3.27	A word about the eight values	57
2.3.28	Beauty tips to finish with.....	58
2.3.29	dlgSizeable as standard dialog	58
2.3.30	It will not get smaller respectively a small one is enough	58
2.4	SplitWindows	59
2.4.1	ListView controls.....	60
2.4.2	DataBrowser on DialogWindows.....	61
2.4.3	TreeView controls.....	61
2.4.4	The example application.....	61
2.4.5	The example data bases.....	61
2.4.6	SplitWindows	66
2.4.7	Calling a SplitWindow	66
2.4.8	The method Init ().....	66
2.4.9	Layout of panes.....	67
2.4.10	More properties of Init ().....	67
2.4.11	The Timer method of reaction.....	68
2.4.12	Instance variables.....	68
2.4.13	The desired pane layout.....	69
2.4.14	Creation of the pane controls	69
2.4.15	Size of the panes	70
2.4.16	Program controlled size change	72
2.4.17	Keyboard handling for the panes.....	73
2.4.18	Creating a FixedText control	74
2.4.19	Creating a TreeView control.....	74
2.4.20	Creating a ListView control.....	74
2.4.21	Creating ListView columns	75
2.5	Filling ListView and TreeView controls.....	76
2.5.1	Filling ListView controls.....	76
2.5.2	Filling TreeView controls.....	77
2.5.3	Important: Unique symbolic item names	77
2.5.4	Practical and nice-to-have: A/B/C trees.....	77
2.5.5	Rather rare: Immediate filling of the complete tree.....	78
2.6	Dynamic filling of ListView controls.....	79
2.6.1	Alternative – virtual ListView ...?.....	80
2.6.2	The method FillListView ().....	80
2.6.3	Time controlled dynamic reloading.....	83
2.7	Mark & Select.....	85

2.7.1	Select ListViewItems.....	85
2.7.2	Select TreeViewItems.....	85
2.7.3	Often forgotten: SetItemAttributes ().....	86
2.7.4	Fast update methods for ListViews.....	86
2.7.5	Other acceleration tips for speeding up.....	88
2.8	Sort ListView columns	88
2.8.1	Sorting nearly all rows.....	89
2.9	Displaying elements	91
2.9.1	Assigning ListViewItems.....	91
2.9.2	Displaying TreeViewItems	91
2.10	Event methods.....	92
2.10.1	Event methods for ListViews	92
2.10.2	Event methods for TreeViews.....	93
2.10.3	Event method ListViewItemChanged ().....	93
2.10.4	Event method ListViewKeyDown ()	94
2.10.5	Event method ListViewMouseButtonDown ()	95
2.10.6	Event method ListViewColumnClick ().....	95
2.11	Keyboard handling	96
2.11.1	The annoying Return key.....	96
2.11.2	Use of context menu key.....	96
2.12	Marking & Focussing	97
2.12.1	Methods for marking	97
2.13	Column gadgets.....	98
2.13.1	Right aligned label column	98
2.13.2	Column width in pixels.....	99
2.13.3	Editing the label column	99
2.14	SingleLineEdits for every ListView column	101
2.14.1	Column definitions	102
2.14.2	Creation of the dynamic SLEs.....	104
2.14.3	Consideration of the Tab order.....	104
2.14.4	Filling of the SLE controls.....	105
2.15	TreeView controls	106
2.15.1	Event method TreeViewSelectionChanged ()	106
2.15.2	About collapsing and expanding nodes.....	106
2.15.3	The common recursive function ExpandCollapseAll ()	108
2.15.4	Marking & Focussing	108
2.15.5	Dynamic filling of TreeView controls	110
2.15.6	A special class as a value for oTVI:value.....	111
2.15.7	A useful test for unique TVI names	112
2.16	Editing a TreeViewItem.....	113
2.16.1	Special controls inside the TreeView.....	113
2.16.2	Mouse click on empty space	115
2.16.3	SingleLineEdit class sleTreeView	116
2.16.4	Write back SLE control into the TreeViewItem	116
2.16.5	Trigger of the SLE creation.....	117
2.16.6	Finally: Create a SLE on a TreeView control.....	117

2.17	Programming a Wheel mouse.....	118
2.18	Right aligned SingleLineEdit controls.....	119
2.19	SLE with ListView.....	120
2.19.1	A “Smart Control”	120
2.19.2	SingleLineEdit class sleListView.....	121
2.19.3	Keyboard handling for sleListView.....	122
2.19.4	Creating pushbuttons.....	122
2.19.5	Pushbutton class pshSleListView.....	123
2.19.6	Keyboard handling for pshSleListView.....	123
2.19.7	Creating ListView controls.....	124
2.19.8	ListView class livSleListView.....	124
2.19.9	Keyboard handling for livSleListView.....	125
2.19.10	Hiding covered controls.....	126
2.19.11	Conclusion.....	126
2.20	Image lists	126
2.20.1	Overlay images for TreeView, ListView & TabControls.....	127
2.20.2	Code – generated or handwritten?.....	127
2.20.3	Icon names – please sensible	128
2.21	Bitmaps next to menu items.....	128
2.21.1	Bitmap? Why?	128
2.21.2	Trick: Bitmap = checkmark	129
2.21.3	API function SetMenuItemBitmaps ().....	129
2.21.4	Ingredients: 1 menu.....	130
2.21.5	And: 2 bitmaps respectively 1	130
2.21.6	Take care: loss of resources... ..	131
2.21.7	The destructor in the life of an object.....	131
2.21.8	Format of bitmaps	132
2.21.9	Renouncement of generated resources.....	134
2.21.10	Variable resources path	134
2.21.11	Parameter uPosition	135
2.21.12	Parameter uFlags.....	135
2.21.13	Function call in the window.....	135
2.21.14	About the Init method	136
2.21.15	Alternative: Call inside the menu.....	137
2.21.16	Calling Axit () manually.....	138
2.21.17	Conclusion: “MS Office 2000 look”? No problem	138
2.22	Define standard menu options.....	138
2.22.1	An example.....	138
2.22.2	API function SetMenuDefaultItem ()	139
2.22.3	An example call	140
2.22.4	Access to submenus	140
2.23	Menu options with “Radiobuttons”	140
2.23.1	Again: Starting with an example	140
2.23.2	API function CheckMenuItem ().....	141
2.24	More menu utilities	143
2.24.1	Refresh menu bar	143

2.24.2	Delete menu entries.....	143
2.24.3	Bitmap as menu item.....	144
2.24.4	Integration in an own menu class.....	145
2.25	Toolbar controls with keyboard handling.....	145
2.25.1	Toolbars & bands.....	145
2.25.2	Focusing with a trick.....	145
2.25.3	Tab order in the toolbar ...?.....	145
2.25.4	The basic principle.....	146
2.25.5	The class sleToolbar.....	146
2.25.6	Enhancements of the Window class.....	147
2.25.7	Instance variable _aTbControls.....	147
2.25.8	Register and unregister controls.....	148
2.25.9	Focusing and defocusing.....	148
2.25.10	Another useful method.....	149
2.26	TabControls.....	149
2.26.1	“Page on demand”.....	150
2.26.2	The class tabStandard.....	150
2.26.3	First selection = create window.....	150
2.26.4	All kinds of help methods.....	151
2.26.5	Disable a tab.....	151
2.26.6	Delete tabs & change labels.....	151
2.26.7	Configuration? Dynamical	152
2.26.8	... or using INI files.....	152
2.26.9	Load tabs without interruptions... ..	154
2.26.10	tabStandard in dlgSizeable.....	155
2.26.11	Some help methods.....	156
2.26.12	Keyboard handling.....	156
2.27	Windows API functions.....	158
2.27.1	Simulate keyboard action.....	158
2.27.2	Create events manually.....	158
2.27.3	Subclassing for Comboboxes.....	159
2.27.4	Set a window into foreground.....	160
2.27.5	Lock controls.....	162
2.27.6	Waiting status.....	162
2.27.7	Debugging.....	163
2.27.8	Palette styled window caption.....	164
2.28	String tables.....	165
	Printing with ReportPro.....	167
3	Printing with ReportPro.....	169
3.1	This chapter.....	169
3.2	Alternatives.....	169
3.2.1	HTML output.....	169
3.2.2	VO’s Printer class.....	169
3.2.3	Report Pro.....	169

3.2.4	Using ReportPro's painter	170
3.2.5	Programming reports.....	170
3.2.6	A meaningful compromise: Configuration of lists in an INI file	171
3.3	Printing with ReportPro.....	171
3.3.1	Installation of the library	171
3.3.2	Using common directories	172
3.3.3	Using the painter.....	172
3.3.4	Variation 1: Calling of Reports	173
3.3.5	Variation 2: Path independent data bases and indexes.....	174
3.3.6	Variation 3: Passing a predefined server object.....	174
3.4	Manual programming.....	175
3.4.1	Advantages & differences.....	175
3.4.2	Class RpPrinter	176
3.4.3	Three important help files.....	177
3.5	Advantage compared to the painter	177
3.5.1	Any and as many Instance variables.....	177
3.5.2	Inheritance of own base classes	178
3.5.3	Base classes for a "CI" (Corporate Identity)	179
3.5.4	No more external layout files *.RPT	179
3.5.5	Higher programming efforts, higher flexibility	180
3.6	Important attributes of RpPrinter	180
3.6.1	Dimensioning interfaces.....	180
3.6.2	Positioning.....	181
3.6.3	Measurement methods.....	181
3.6.4	Creating printable elements	181
3.6.5	Selecting the printer	182
3.7	Callback methods.....	182
3.7.1	Return value of callback methods	182
3.7.2	The parameter oReport	183
3.7.3	Other parameters.....	183
3.7.4	Order of the callback methods	183
3.7.5	Method PrintStart (oReport)	183
3.7.6	PrintPageBody (oReport).....	183
3.8	Example 1 (rptOne): At first sample: simple & minimalistic	184
3.8.1	Class rptOne	184
3.8.2	Callback methods.....	185
3.8.3	Calling the Report.....	186
3.8.4	Using class rptOne	187
3.8.5	Number of pages? – Indefinitely.....	187
3.9	Example 2 (rptTwo): Server objects, headers and footers, fonts	187
3.9.1	Class rptTwo.....	188
3.9.2	Instance variables.....	188
3.9.3	Method Init ().....	189
3.9.4	Server _oDbCustomer, _oDbOrder	189
3.9.5	Arrays _aFieldCustomer, aFieldOrder.....	190
3.9.6	Further Instance variables.....	190

3.9.7	Further callback methods.....	190
3.10	Page treatment.....	192
3.10.1	Paging within PrintPageBody ().....	192
3.10.2	Callback method PrintSetPage ().....	192
3.10.3	Own page variables	193
3.10.4	Calculation of the total number of pages.....	193
3.10.5	Method CountPages ()	193
3.11	Example 3 (rptThree): Author and Title Data.....	194
3.12	Example 4: Base class rptStandard.....	194
3.12.1	Instance variables & interfaces.....	194
3.12.2	General methods.....	195
3.13	Classes rptCustomer and rptOrder	196
3.14	Example 5: rptIni {}	196
3.14.1	Properties of rptIni.....	196
3.14.2	Columns.....	198
3.14.3	Methods of rptIni {}	199
3.14.4	Using class rptIni {}	199
	Columbo – The VO Code & Repository Inspector	201
4	Columbo – The VO Code & Repository Inspector.....	203
4.1	This chapter	203
4.1.1	Overview	204
4.1.2	What VO is missing.....	204
4.1.3	What gets on your nerves in VO.....	206
4.1.4	Interesting programming details.....	206
4.2	Installation and organisation	212
4.2.1	Special arrangement in VO 2.6.....	212
4.2.2	The Columbo files.....	213
4.2.3	The directory structure.....	213
4.2.4	The configuration file COLUMBO.INI	214
4.2.5	Registration of Columbo as a “Toy”	214
4.2.6	Changes in CAVOWED.INF	214
4.2.7	Creating EXE and DLL	215
4.3	The pane controls in detail.....	216
4.3.1	TreeView “Applications”.....	216
4.3.2	Listview “Modules”.....	219
4.3.3	Listview “Entities”.....	219
4.3.4	Listview “Dependants”.....	220
4.3.5	Listview “Implicants”.....	221
4.3.6	TreeView “Classes”.....	221
4.3.7	RichEdit “Sourcecode”.....	222
4.3.8	Reports	224
4.3.9	Miscellaneous	227
4.3.10	Help and “About...” dialog.....	228
4.4	The modules of Columbo.....	229

4.4.1	Start-modules.....	229
4.4.2	VO base classes	231
4.4.3	Control base classes	232
4.4.4	Window classes	234
4.4.5	Pane controls	234
4.4.6	Zoom controls.....	236
4.4.7	RichEdit control for source code.....	238
4.4.8	Repository interfaces.....	241
4.4.9	Repository classes.....	243
4.4.10	Miscellaneous functions	253
4.4.11	INI classes	255
4.4.12	Bitmaps and icons	256
4.4.13	Menus.....	259
4.4.14	Custom controls	264
4.4.15	Dialogs	266
4.4.16	HTML class.....	285
4.4.17	Windows	288
4.5	Columbo PlugIns	292
4.5.1	Solution: PlugIns	292
4.5.2	Visible classes inside the plugin.....	292
4.5.3	An example: list of class-inheritances	293
4.5.4	Sample code for this.....	294
4.5.5	Requirements for plugin DLLs	295
4.5.6	Reading menu options	295
4.5.7	Choosing a plugin menu option	296
4.5.8	Loading a plugin DLL.....	296
4.5.9	Function <code>__LoadLibrary ()</code>	297
4.5.10	Alternatively: <code>__voDllClassInit ()</code> and <code>_voDllMain ()</code>	298
4.6	Future prospects.....	298
	Custom Controls	299
5	Custom Controls.....	301
5.1	This chapter... ..	301
5.2	Overview	302
5.2.1	Custom Controls – why?	302
5.2.2	Custom Controls – preferably also interactive	302
5.2.3	Custom drawing	303
5.2.4	... for TreeView controls.....	303
5.2.5	... for ListView controls.....	305
5.3	The fundamentals.....	305
5.3.1	Structures.....	305
5.3.2	Declaration of a structure variable.....	306
5.3.3	“As” versus “Is”	306
5.3.4	Using structures	306
5.3.5	Useful functions.....	306

5.3.6	Own functions with structures.....	307
5.3.7	Drawing elements.....	307
5.3.8	hDC – the Device Context	309
5.3.9	Useful and please use it: “Stock” objects	309
5.3.10	Return values of API functions	309
5.3.11	VO-Classes versus API.....	309
5.3.12	Likewise: VO methods versus API functions.....	310
5.3.13	Selecting drawing elements.....	310
5.3.14	Destroying drawing elements	311
5.3.15	All in one piece: a Setup class.....	311
5.3.16	Drawing functions	313
5.3.17	Point to point: Polygon drawing.....	314
5.4	Custom drawing controls	314
5.4.1	Custom drawing for ListView controls.....	314
5.4.2	Drawing with Dispatch () or ControlNotify ()	315
5.4.3	Custom drawing for TreeView controls.....	316
5.4.4	A useful combination: a TreeListView control.....	318
5.5	Custom Controls.....	321
5.5.1	The necessary minimum	322
5.5.2	Just in case: The base class ccStandard.....	322
5.5.3	CustomControl class ccListView	322
5.5.4	A test case: dlgListView_vo / dlgListView.....	323
5.6	Example 1: Scrolling text control ccMovieText	324
5.6.1	Class ccMovieText.....	324
5.6.2	Methods of ccMovieText.....	326
5.6.3	Drawing methods Draw... ().....	328
5.6.4	Using ccMovieText inside dlgMovieText.....	329
5.7	Example 2: LED control ccDisplay	330
5.7.1	Class ccDisplay	330
5.7.2	Setup class DisplaySetup	331
5.7.3	Methods DrawDigits () and DrawDigit ()	332
5.7.4	Methods to draw the segments	334
5.7.5	Drawing with numbers: API function Polygon ().....	334
5.7.6	Responding on size changes.....	336
5.8	Example 3: Gradient color header ccPaneCaption.....	337
5.8.1	Class ccPaneCaption	337
5.8.2	Methods of ccPaneCaption	339
	Drawing-methods	339
5.8.3	Event methods.....	341
5.8.4	Pane control classes.....	341
5.8.5	The pane client dialog for pane controls	342
5.8.6	dlgPane – rather universal.....	343
5.9	Example 4: List control ccList	345
5.9.1	Custom Controls – anything goes	345
5.9.2	Class ccList.....	345
5.9.3	Setup class ccListSetup.....	346

5.9.4	Class ccListRow	348
5.9.5	Interfaces of ccListRow.....	349
5.9.6	Methods of ccListRow	350
5.9.7	Drawing a line: Method Paint ().....	351
5.9.8	Vertical scrollbar for ccList	353
5.9.9	Drawing ccList.....	354
5.9.10	Keyboard handling.....	356
5.9.11	Mouse handling	358
5.9.12	And now – THE WHEEL.....	358
5.9.13	Skip methods	360
5.9.14	Fast display using scrolling functions.....	363
5.9.15	Using ccList.....	365
5.9.16	Bonus track: Specials with Subclassing.....	365
5.9.17	List with double-lined rows	366
5.9.18	Class ccListRowDetail	366
5.9.19	Determination of the clicked row	366
5.9.20	Drawing method Paint ()	366
5.10	Example 5: Calendar control ccCalendar.....	368
5.10.1	Drawing method DrawDays ().....	371
5.10.2	Mouse and keyboard handling	373
5.10.3	Bonus track: Lassoing	375
5.11	Example 6: “TreeListView” ccTreeView	378
5.11.1	Class ccTreeViewItem	379
5.11.2	Column class ccTreeViewItemColumn	379
5.11.3	Dynamic controls inside column cells	380
5.11.4	Defining a ccTreeView controls.....	381
5.11.5	Filling a ccTreeView controls.....	382
5.12	Outlook	383
	E-mails & Internet.....	385
6	E-mails & Internet.....	387
6.1	This chapter	387
6.2	Start an e-mail Client.....	387
6.3	E-mails.....	388
6.3.1	Sending e-mails	388
6.3.2	Once more: Sending e-mails.....	389
6.3.3	E-mailings	391
6.3.4	Main window dlgEmail	391
6.3.5	Method Send ()	392
6.3.6	Internet functions	392
6.3.7	Tab page 1: “E-mail”.....	393
6.3.8	“Drop” e-mail attachments	393
6.3.9	Measuring a file’s size.....	394
6.3.10	Change a FixedIcon.....	395
6.3.11	Tab 2 to 7: ListView plus RTF control.....	395

6.4	Displaying Web sites	395
6.4.1	Start the Internet Explorer	396
6.4.2	A Web control for files	396
6.4.3	A Web control as help system	397
6.4.4	Comboboxes for controlled data input	397
6.4.5	Creating a wrapper class	398
6.4.6	Class MicrosoftWebBrowser	399
6.4.7	Two ways to go	399
6.4.8	Suboptimal	399
6.4.9	... and better	399
6.4.10	OLE control inside a dummy control	400
6.4.11	Two callback event methods	400
6.4.12	Key F4 -> Address entry	401
6.5	Web links inside RTF controls	401
6.5.1	Some constants	401
6.5.2	Class rtfStandard	401
6.5.3	Dispatcher for Window class	401
6.5.4	Calling the Internet Explorer	402
	Web applications	403
7	Web applications	405
7.1	This chapter	405
7.2	Implementation	405
7.2.1	A VO application as basis	405
7.2.2	A first overview	406
7.2.3	The basic principle	407
7.2.4	The data base structure	407
7.2.5	Communication via CGI	408
7.2.6	JUNO inside VO's IDE	409
7.3	Client side: HTML pages	409
7.3.1	Formats as Cascading Style Sheets	409
7.3.2	The header	410
7.3.3	A Tab line as navigation bar	410
7.3.4	Communication with the VO application on the server side	412
7.3.5	"Global" variables across page borders	412
7.4	The tab pages	413
7.4.1	Tab page "Login"	413
7.4.2	Tab page "Search"	415
7.4.3	Tab page "Overview"	416
7.4.4	Tab page "List"	419
7.4.5	Tab page "Edit"	420
7.4.6	Tab page "New"	422
7.4.7	Tab page "About"	422
7.5	Server side: VO application JUNO.EXE	424
7.5.1	Internet server application: without any GUI	425

7.5.2	VO class LogFile	426
7.5.3	VO class HttpCgiContext	428
7.5.4	Class IniFile – managing INI files.....	428
7.5.5	Function Token () – “chopping” strings	429
7.5.6	Function Start () – launching the application	429
7.6	Class JunoHtmlForm – managing an HTML file	431
7.6.1	Example for using JunoHtmlForm	432
7.6.2	Replacing variables and fields	433
7.6.3	Macros – loops through the data area	433
7.6.4	“Cascading Style Sheets” for simplification	435
7.7	Class JunoHttpCgiContext – Lynchpin of the application	436
7.7.1	The method Init ()	436
7.7.2	Creating the data base.....	438
7.7.3	Opening the data base.....	439
7.7.4	Calling from function Start ().....	440
7.7.5	Method CheckParameter ().....	444
7.7.6	Calling tab page “Login”	444
7.7.7	Pressing “Go” button on “Login” page.....	445
7.7.8	Calling tab page “Search”.....	445
7.7.9	Pressing “Go” button on “Search” page.....	445
7.7.10	Calling tab page “List” respectively “Overview”	446
7.7.11	Pressing “Go” button on page “List” respectively ”Overview”.....	450
7.7.12	Calling tab page “Edit”.....	451
7.7.13	Pressing “Go” button on page “Edit”	451
7.7.14	Pressing “Save” button on page “Edit”.....	453
7.7.15	Pressing “Delete” button on page “Edit”	454
7.7.16	Calling tab page “New”.....	454
7.7.17	Pressing “Save” button on page “New”	455
7.7.18	Calling tab page “About”.....	456
7.7.19	Using HTML file MESSAGE.HTML	456
7.8	Installation.....	457
7.8.1	Necessary files.....	457
7.8.2	Directory structure in the development environment.....	458
7.8.3	Directory structure on the internet server.....	458
7.8.4	Set-up a local Webserver.....	459
7.8.5	Set-up the Personal Web Manager	460
7.8.6	Set-up the Internet Information Service.....	461
7.8.7	Starting Juno.Exe on the local internet server	461
7.8.8	Starting Juno.Exe on the internet	461
7.8.9	CGI – something ancient...?.....	462
7.8.10	Variables vs. Data records	463
7.8.11	SessionIDs – without ending	463
7.8.12	User management – SessionID data bases vs. Cookies.....	464
7.9	Internet applications with VO & DBF & CDX	464
7.10	Web application www.ivo-wessel.de.....	464

API-Programming.....	467
8 API-Programming.....	469
8.1 How to read C.....	470
8.1.1 C- Language Elements.....	474
8.1.2 Operators.....	487
8.1.3 Control-Structures.....	488
8.1.4 Functions.....	490
8.1.5 Header-Files.....	494
8.1.6 Structures.....	496
8.1.7 Unions.....	501
8.1.8 Resume.....	502
8.1.9 Continuing sources.....	505
8.2 Frequent tasks and problems in the API-programming.....	505
8.2.1 Utilization of API-functions.....	505
8.2.2 Memory Allocation.....	507
8.2.3 Usage of Structures.....	510
8.2.4 Sending and Processing of Messages.....	513
8.2.5 Usage of Function-Pointers.....	525
8.2.6 Resume.....	533
8.3 A Banner-Control.....	537
8.4 Continuing Sources.....	575
COM-Programming.....	579
9 COM-Programming.....	581
9.1 The Theory explained through an Example.....	583
9.1.1 Interface Considerations.....	589
9.1.2 Far and wide Dynamics.....	590
9.1.3 Inhabitants of Potemkin-like Villages (The Bridge Pattern).....	591
9.1.4 Interim Balance.....	593
9.1.5 New Interfaces.....	594
9.1.6 Inquiringly, but certain.....	596
9.1.7 Orthogonal Interfaces.....	597
9.1.8 How long does an Object live?.....	600
9.1.9 A Factory.....	602
9.1.10 An ID.....	604
9.1.11 What is won?.....	605
9.2 A SAX-Implementation in CA-Visual Objects.....	612
9.3 An Automation Server with CA-Visual Objects.....	619
9.4 Access to an Automation Server with CA-Visual Objects.....	624
9.5 Preparation of an Automation Server with VOCOM.....	626
9.6 The Microsoft Agent Control.....	630
9.6.1 Use of the ActiveX-Controls.....	630
9.7 More Sources.....	635

Enterprise-Programming.....	637
10 Enterprise-Programming.....	639
10.1 DCOM/COM+.....	639
10.2 Overview of the MTS.....	642
10.3 Programming of the MTS.....	644
10.4 Registering of a Component at the Component- Services.....	645
10.5 Using the MTS in CA-Visual Objects.....	649
10.6 Windows DNA.....	650
10.7 Outlook.....	651
10.8 Further sources.....	652
XML and Web-Services.....	653
11 XML and Web-Services.....	655
11.1 Technologies and concepts.....	656
11.1.1 Document Type Definition (DTD).....	662
11.1.2 XML-Schema.....	664
11.1.3 Namespaces.....	666
11.1.4 Displaying Documents (XSL).....	667
11.1.5 XML-Parser.....	669
11.2 Parsing XML-Documents.....	671
11.2.1 Using DOM.....	673
11.2.2 Using SAX.....	688
11.2.3 Using XPath.....	700
11.3 Web-Services.....	702
11.3.1 SOAP.....	703
11.3.2 WSDL.....	705
11.3.3 Access to a Web-Service.....	708
11.3.4 Accessing a Web-Service with a CA-Visual Objects-Program.....	709
11.3.5 Creating a Web-Service with CA-Visual Objects.....	714
11.4 For further Reference.....	719
Design Patterns.....	721
12 Design Patterns.....	723
12.1 A few common Design Patterns.....	726
12.1.1 Adapter.....	726
12.1.2 Abstract Factory.....	733
12.1.3 Bridge.....	739
12.1.4 Proxy.....	743
12.1.5 Strategy.....	748
12.1.6 Iterator.....	751
12.1.7 Singleton.....	759
12.2 Processing of Hierarchical Data.....	761

12.2.1	Composite.....	762
12.2.2	Visitor.....	766
12.3	Resume.....	771
Version Control.....		775
13	Version Control.....	777
13.1	A little Theory	777
13.1.1	General	777
13.1.2	The Operation of Version-Control Tools	778
13.2	Version-Control with CA-Visual Objects	781
13.2.1	Version-Control based on AEFs.....	782
13.2.2	Version-Control based on PRGs.....	784
13.2.3	Version-Control on basis of the SCC.....	786
13.3	Further Sources	801
Index		803