

Inhaltsverzeichnis

Vorwort des Herausgebers	15
1 Die VCL und CLX	17
1.1 Ein Euro-Umrechner	17
1.1.1 Umrechnungskurse und -regeln	18
1.1.2 Logik und Oberfläche trennen	20
1.1.3 Die Umrechnung implementieren	21
1.1.4 Control-Gruppen wiederverwenden mit Frames	23
1.1.5 Bilder aus Ressourcen laden	28
1.1.6 Komponenten im Quelltext erzeugen	31
1.1.7 Exkurs: Frames visuell bearbeiten	34
1.1.8 Umgerechnete Beträge anzeigen	35
1.1.9 Bessere Tastaturbedienung	37
1.2 Portierung nach Linux	38
2 Laufwerke, Verzeichnisse und Dateien	41
2.1 Standarddialoge	43
2.1.1 Der Standarddialog zum Öffnen von Dateien	43
2.1.2 Der nächste Standarddialog – Dateien speichern unter	50
2.1.3 TOpenPictureDialog, TSavePictureDialog	51
2.2 Datei- und Laufwerkskomponenten (File, Directory, Drive, Filter)	51
2.2.1 TDriveComboBox	52
2.2.2 TDirectoryListBox	56
2.2.3 TFileListBox	61
2.2.4 TFilterComboBox	62
2.3 Funktionen zum Bearbeiten und Verwalten von Laufwerken, Verzeichnissen und Dateien	70
2.3.1 Laufwerke	70
2.3.2 Verzeichnisse	75
2.3.3 Dateien	75
2.3.4 Arbeiten mit den Inhalten einer Datei	87
2.3.5 Fehlerbehandlung	106
2.3.6 Routinen für Textdateien	107
2.4 Dateihandles und Dateivariablen	108



2.5	Direkte API-Aufrufe	111
2.6	Suchen von Dateien mit API-Funktionen – Vergleich mit den Delphi-Funktionen	114
2.7	Rekursives Suchen im Dateibaum	117
2.7.1	Rekursive Dateibearbeitung – Ein etwas aufwendigeres Beispielprojekt ..	121
2.8	Diskussion der unterschiedlichen Funktionsgruppen	132
2.9	Shell-Programmierung	134
2.10	Des Rätsels Lösung	135
3	Docking	137
3.1	Die ersten Versuche	138
3.2	Docking Manager	139
3.3	Andocken unerwünscht	142
3.4	Ein Konzept mit Schwachstellen	143
4	TApplication, die etwas andere Komponente	145
4.1	Das Fundament im Bauwerk	146
4.2	Wozu TApplication?	147
4.3	Die Eigenschaften	150
4.3.1	Die wichtigsten Eigenschaften im Überblick	151
4.4	Die Methoden	154
4.4.1	Die wichtigsten Methoden im Überblick	155
4.5	Ereignisse	158
4.5.1	Die wichtigsten Ereignisse im Überblick	159
4.6	TApplication im Griff	160
4.7	Angriffsziel Projektdatei	161
4.8	Das unsichtbare Programm	162
4.9	Ein Programm für die Taskleiste	163
4.10	Botschaften auf Irrwegen	164
4.11	Läuft sie schon?	167
4.12	Schongang	169
4.13	Freiräume	172
5	Exception Handling – Ausnahmebehandlung	175
5.1	Syntax des Exception Handling in Delphi	176
5.1.1	Der Stoff, aus dem die Ausnahmen gemacht sind	180
5.1.2	Die Reihenfolge der Bearbeitung	183
5.2	Spezifische Informationen über die Instanz der Ausnahme ...	186
5.2.1	Die »Stapelabwicklung«	187
5.2.2	Exit, Break, Continue	189
5.2.3	»At« – eine Rarität	195
5.3	Hilfsroutinen	200
5.3.1	Assertions	201



6	Ressourcen	203
6.1	Der Resource Workshop	205
6.1.1	Benutzerdefinierte Ressourcen	207
6.2	Mehrsprachige Programme	208
7	Die Registry	211
7.1	Vor- und Nachteile	212
7.2	Der Registrierungseditor	214
7.2.1	Struktur der Registry	215
7.2.2	Datentypen der Schlüsseleinträge	221
7.2.3	Die Dateien »System.dat« und »User.dat«	222
7.3	Registry-Funktionen in Win32	223
7.3.1	Unicode- und ANSI-Format	223
7.3.2	Verwendete Datenstrukturen	224
7.3.3	Einige wichtige Win32-Funktionen	225
7.4	Anwendungen und Beispiele	229
7.4.1	Durchzuführende Eintragungen	229
7.4.2	Dateinamenerweiterungen (Dateiextension)	231
7.4.3	DDE-Verknüpfung mit dem Explorer	237
7.4.4	Anwendungsverzeichnis	242
7.4.5	Automatischer Programmstart	243
7.5	Die Bibliothek »Registry.pas«	245
7.5.1	Verfügbare Klassen	245
7.5.2	Datenstrukturen	245
7.5.3	Ausnahmebehandlung	247
7.5.4	Die Klasse TRegistry	248
7.5.5	Die Klasse TRegIniFile	257
7.5.6	Die Klasse TRegistryIniFile	259
8	Hooks	265
8.1	Dynamic Link Libraries (DLLs)	266
8.1.1	Statisches und dynamisches Einbinden	267
8.1.2	Wohin mit der DLL?	269
8.1.3	Fehlersuche in DLLs	271
8.1.4	Erstellen von DLLs	274
8.1.5	Fenster in DLLs	277
8.1.6	Datenaustausch zwischen Anwendungen über DLLs	280
8.2	Eingehakt : Hooks	285
8.2.1	Die Parameter hmod und dwThreadld	287
8.2.2	Die Callback-Funktion	289
8.2.3	Auswerten der Parameter	290
8.3	Endspurt	292



9	Einführung in das Open Tools API	295
9.1	Hello World!	298
9.2	Verwendung von Services	301
9.3	Den Editor beeinflussen	304
9.4	Notifiers	308
9.5	Über die Experten hinaus...	310
9.6	Übersicht: IOTAMessageServices	311
9.7	Übersicht: IOT>ActionServices	314
9.8	Übersicht: IOTAPackageServices	314
9.9	Übersicht: IOTAToDoServices	315
9.10	Übersicht: IOTAWizardServices	316
9.11	Übersicht: IOTAEditorServices	317
9.12	Übersicht: IOTADebuggerServices	318
9.12.1	Referenzen, Quellen und weitere nützliche Ressourcen	319
10	Prozeßverwaltung	321
10.1	Prozesse	321
10.1.1	Kernel-Objekte	324
10.1.2	Prozeßerzeugung	325
10.1.3	Prozeßverwaltung	331
10.2	Threads	332
10.2.1	Threaderzeugung	335
10.2.2	Threadverwaltung	337
10.2.3	Stack eines Threads	339
10.2.4	Deadlocks	341
10.2.5	Prioritäten	344
10.3	Synchronisation	348
10.3.1	Kritische Bereiche	353
10.3.2	Mutexe	356
10.3.3	Semaphoren	358
10.3.4	Ereignisse	362
10.3.5	Named Pipes	365
10.3.6	Verzögerung von Threads	372
10.3.7	Synchrone Zähler	374
10.3.8	Synchronisation in verteilten Anwendungen	376
10.3.9	Beeinflussung anderer Prozesse	376
10.4	Die Klasse TThread	380
10.4.1	Eigenschaften (Properties)	382
10.4.2	Methoden	384
10.5	Synchronisation	386
10.5.1	Die Klasse TCriticalSection	386
10.5.2	Die Klasse TEvent	386
10.5.3	Klasse TMultiReadExclusiveWriteSynchronizer	387
10.5.4	Anmerkungen	388
10.6	Entwurfsmuster	388
10.7	Beispiele	390
10.7.1	Beispiel 1: Lagersimulation	390



10.8	Beispiel 2: Producer/Consumer	392
10.9	Beispiel 3: Lagersimulation	394
10.10	Beispiel 4: Interlocked-Zähler	397
10.11	Beispiel 5: Supermarkt	399
11	Drucken	405
11.1	Die Unit Printers	405
11.2	Dialoge für den Anwender	411
11.3	Drucken mit Hindernissen	413
11.4	Der Weg zur Unabhängigkeit	414
11.5	Bis zum Äußersten	416
11.6	Eine Komponente für die Druckerausgabe	418
11.7	XPrint in der Praxis	427
11.8	Offene Wünsche	442
12	Die serielle Schnittstelle	443
12.1	Der Anschluß, das Kabel	443
12.2	Technische Betrachtungen	449
12.3	Handshake	452
12.4	Die Schnittstellen im PC	453
12.5	Serielle Datenkommunikation mit dem Windows-API	454
12.6	Ein Konfigurationsdialog	469
12.7	Die serielle Schnittstelle als I/O-Port	471
12.8	Praktische Beispiele	473
12.9	Mit Komponenten wird alles einfach	474
12.10	Ohne Zukunft?	490
13	Multimedia	491
13.1	MIDI-Sounds	491
13.2	Mit Wave-Dateien arbeiten	523
13.3	DirectX	560
13.3.1	DirectSound	560
13.3.2	DirectMusic	602
13.4	DirectDraw	635
13.5	Transformieren mit DirectX Media	657
13.6	3D-Programmierung mit Delphi und DirectX 8	681
14	Internet-Programmierung	707
14.1	Die Komponentenseite »Internet«	707
14.1.1	Allgemeines	707
14.1.2	Das Intranet	710
14.1.3	Computernetzwerke	712
14.1.4	IP-Adressen und Domain-Namen	717
14.1.5	Die Internetkomponenten	719



14.1.6	Verteilte Anwendungen	721
14.1.7	Webserver-Anwendungen	721
14.1.8	CGI-Webserver-Anwendungen	738
14.1.9	Antwortbotschaften generieren	743
14.1.10	Einbindung von Datenbank-Informationen	751
14.1.11	Socket-Anwendungen	756
14.2	Die Indy-Komponenten	770
14.2.1	Die drei Indy-Registerkarten: Clients, Server und Vermischtes	770
14.2.2	Erste Schritte mit Indy	779
14.2.3	TCP-Programmieren à la Indy	784
14.2.4	Netzwerk-Diagnose mit Hilfe von Servern (Projekte: IndyDiagServer, IndyEchoClient)	801
14.2.5	Weitere Diagnose-Instrumente: Ping, Traceroute, Finger und Whois	803
14.2.6	File Transfer Protocol (FTP)	812
14.2.7	Post im Internet (Projekt: IndyMail)	819
14.2.8	Hypertext Transfer Protocol (HTTP)	829
14.2.9	Komponenten für spezielle Bedürfnisse	842

15 Seifenoper – SOAP **851**

15.1	Was ist WSDL?	853
15.2	Web Services in Delphi 6	854
15.3	Erzeugen eines SOAP-Clients	854
15.4	Entwickeln von SOAP-Servern	859
15.4.1	Das Interface des SOAP-Servers	861
15.4.2	Ein SOAP-Session-Manager-Client	867
15.5	Zusammenfassung	870
15.6	Quellen	871

16 Installation eigener Programme **873**

16.1	InstallShield	873
16.1.1	Die »Demo«	875
16.1.2	Die Bedienung von InstallShield	876
16.2	Nützliche Verpackungen	878
16.2.1	Magnetisch, optisch, virtuell	878
16.2.2	»Zip« und Konsorten	879
16.2.3	Schrumpfen mit »UPX«	880
16.2.4	Installations-Martyrium	880
16.2.5	Schnippchen schlagen	881
16.2.6	Die Installation der Installation	883
16.2.7	Die dicke Ressource	884
16.2.8	»OneFile« muß konfiguriert werden	887
16.2.9	Der erste Versuch	888
16.2.10	»OneFile« im Detail	889
16.2.11	»Resource Workshop« eingebaut?	890
16.2.12	Möglichkeiten	891



17	Die Windows-Hilfe	893
17.1	Einführung	894
17.1.1	Steuercodes	897
17.1.2	Kunst in der Hilfe – Bitmaps verwenden	899
17.1.3	Heiße Punkte in der Bitmap – Die Hotspots	900

Die folgenden Bonus-Kapitel befinden sich auf der CD:

18	Mit dem Prozessor per Du (k18.pdf)	1
18.1	Aufwendig, umständlich, fehlerträchtig?	2
18.2	Aufrufkonventionen	4
18.3	Typen gibt's!	7
18.4	Globales und lokales, Konstanten und Variablen	8
18.5	Der kleine Unterschied	11
18.6	Optimal für die Optimierung?	13
18.7	Ein bißchen Zeigerarithmetik	16
18.8	Assembler ruft Pascal	17
18.9	Fit für den Pentium	19
18.10	Debugging	20
18.11	Zum Ausprobieren	21
19	Komponenten, ActiveX und Packages (k19.pdf)	23
19.1	Wozu brauche ich eine eigene Komponente?	24
19.2	Die erste Komponente – ein Koordinatensystem	25
19.3	Die Komponente vererben	26
19.3.1	VCL-Basisklassen	26
19.3.2	Los geht's!	29
19.4	Keine geschlossene Veranstaltung: die Schnittstelle	30
19.4.1	Eigenschaften definieren	32
19.4.2	Methoden mit Methode	35
19.4.3	Hier ist was los – Ereignisse	38
19.5	Erst testen – dann installieren!	41
19.6	Was steckt hinter Packages?	46
19.6.1	Die Grundausstattung – Mitgelieferte Packages	47
19.6.2	Packages in der IDE	51
19.6.3	Eigene Packages entwickeln	52
19.6.4	Feineinstellung des Compilers	55
19.6.5	Fehler in Packages	57
19.7	ActiveVCL – ActiveX-Programmierung	59
19.7.1	Delphi und ActiveX	59
19.7.2	Ein ActiveX erzeugen	60



19.7.3	ActiveX-Controls verwenden	63
19.7.4	Selbst Hand anlegen – Nachbearbeitung mit dem Typbibliotheks-Editor	65
19.7.5	Das ActiveX im Test	73
19.7.6	Ich hasse VB!	75
19.7.7	Zwischenstand	76
19.7.8	Wir werten das ActiveX auf	77
19.7.9	Das Web mit Windows vereinnahmen...	80
19.8	Nicht »einfache« Eigenschaften	84
19.8.1	Komponentenzeiger	84
19.8.2	Array-Eigenschaften	86
19.8.3	Mengen-Eigenschaften	88
19.8.4	Records	88
19.8.5	Objekt-Eigenschaften	88
19.9	Näher dran – Messages behandeln	90
19.9.1	Fenster ziehen	91
19.9.2	Bilder gucken	93
19.9.3	Nicht-visuelle Messages	95
19.10	Die besondere Komponente TForm	99
20	DLLs und API-Programmierung (k20.pdf)	101
20.1	Statisches und dynamisches Linken	101
20.2	Beschränkte DLLs	103
20.3	Warum dynamisch?	104
20.4	DLLs einbinden	104
20.4.1	Ablauf eines Windows-API-Programms	105
20.5	Probleme mit Datentypen	106
20.6	Rückrufe richtig aufrufen	109
20.6.1	Funktionen einbinden	111
20.7	Der Rest des Programms...	112
20.8	DLLs programmieren	115
20.9	Aufbau einer DLL	116
20.10	Funktionen für die Bibliothek	118
20.11	Dynamischer als dynamisch	121
20.12	Vom Eingang zum Ausgang	123
20.13	Komponenten und DLLs	125



21	Kommandozeile und Konsole (k21.pdf)	127
21.1	Bessere Kommandozeilen	128
21.2	Projekt: Suchpfad in der Registry	129
21.2.1	Der Schlüssel »App Paths«	129
21.2.2	Überblick	130
21.2.3	Programmierung	131
21.2.4	Verarbeitung der Kommandozeile	132
21.2.5	Arbeit an der Registry	135
21.2.6	Anwendung	138
21.3	Optionen verarbeiten mit getopt	138
21.3.1	Kurzreferenz	139
21.3.2	Beispiel: Message-Box für Skripte	140
21.3.3	RegPath mit getopt	143
21.4	Das Konsolen-API	144
21.4.1	Vor- und Nachteile	144
21.4.2	Kurzer Entwurf	145
21.4.3	Initialisierung	146
21.4.4	VCL im Textmodus	147
21.4.5	Low-Level-Ausgabe	149
21.4.6	Eingabe mit Tastatur und Maus	150
21.4.7	Signalbehandlung	153
	Stichwortverzeichnis	907

