



**Computer & Literatur Verlag GmbH**



# LAZARUS

von Michaël Van Canneyt, Mattias Gärtner,  
Swen Heinig, Felipe Monteiro de Carvalho,  
Inoussa Ouedraogo, Marc Weustink und Jörg Braun



Bibliographische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Alle Rechte vorbehalten. Ohne ausdrückliche, schriftliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus in irgendeiner Form durch Fotokopie, Mikrofilm oder ein anderes Verfahren zu vervielfältigen oder zu verbreiten. Dasselbe gilt für das Recht der öffentlichen Wiedergabe.

Der Verlag macht darauf aufmerksam, daß die genannten Firmen- und Markenzeichen sowie Produktbezeichnungen in der Regel marken-, patent-, oder warenzeichenrechtlichem Schutz unterliegen.

Die Herausgeber übernehmen keine Gewähr für die Funktionsfähigkeit beschriebener Verfahren, Programme oder Schaltungen.

2. durchgesehene und aktualisierte Auflage 2011

© 2011 by C&L Computer und Literaturverlag  
Zavelsteiner Straße 20, 71034 Böblingen  
E-Mail: [info@cul.de](mailto:info@cul.de)  
WWW: <http://www.CuL.de>

Coverdesign: Hawa & Nöh, Neu-Eichenberg  
Satz: C&L-Verlag  
Druck: PUT i RB DROGOWIEC  
Printed in Poland

Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt

ISBN 978-3-936546-63-7

# INHALT

<b>Vorwort .....</b>	<b>13</b>
----------------------	-----------

---

<b>Kapitel 1</b>	
<b>Die Architektur von Lazarus .....</b>	<b>15</b>

von **Mattias Gärtner**

<b>1.1 Der Quelltexteditor .....</b>	<b>19</b>
1.1.1 CodeTools.....	20
1.1.2 Schnelle Navigation im Code .....	20
1.1.3 Automatische Vervollständigung .....	23
<b>1.2 Basiswissen zur Sprache .....</b>	<b>30</b>
1.2.1 Basistypen .....	33
1.2.2 Aufbau und Funktion von Units .....	39
1.2.3 Objektorientierung in Lazarus.....	42
1.2.4 Arrays in Free Pascal .....	45
1.2.5 Compilerdirektiven .....	50
<b>1.3 Projekte .....</b>	<b>52</b>
<b>1.4 Komponenten.....</b>	<b>59</b>
<b>1.5 Packages .....</b>	<b>60</b>
1.5.1 C-Bibliotheken in Free Pascal.....	62
1.5.2 Free-Pascal-Bibliotheken in C .....	64
1.5.3 Neue Komponente in der IDE installieren.....	65
1.5.4 Units in mehreren Projekten.....	68
1.5.5 Virtuelle Units .....	70
1.5.6 Units für verschiedene Plattformen .....	70

1.5.7 Packages suchen .....	72
1.5.8 Verzeichnisse und Suchpfade .....	74
1.5.9 Abhängigkeiten und Vererbung .....	75
1.5.10 Kompilierung .....	76
<b>1.6 Quelltextdokumentation .....</b>	<b>79</b>
<b>1.7 Unicode .....</b>	<b>81</b>

## Kapitel 2

### Lazarus installieren ..... 85

von Jörg Braun, Swen Heinig und Felipe Monteiro de Carvalho

<b>2.1 Download aus der Versionsverwaltung.....</b>	<b>87</b>
2.1.1 Die Installation von TortoiseSVN .....	87
2.1.2 Das Programmpaket Subversion.....	88
2.1.3 SVN-Verzeichnisse von Lazarus und FPC .....	91
2.1.4 Basisarbeiten auf der Kommandozeile .....	93
2.1.5 Auschecken mit TortoiseSVN.....	95
<b>2.2 Installieren auf Windows .....</b>	<b>98</b>
<b>2.3 Installieren auf Linux .....</b>	<b>101</b>
<b>2.4 Installieren auf FreeBSD .....</b>	<b>105</b>
<b>2.5 Installieren auf MacOS X .....</b>	<b>114</b>

## Kapitel 3

### Die IDE ..... 115

von Swen Heinig und Marc Weustink

<b>3.1 Das Lazarus-Menü .....</b>	<b>116</b>
3.1.1 Datei.....	116
3.1.2 Bearbeiten.....	118
3.1.3 Suchen.....	125
3.1.4 Ansicht .....	130
3.1.5 Projekt.....	138

---

3.1.6	Compilereinstellungen.....	147
3.1.7	Start .....	159
3.1.8	Package .....	161
3.1.9	Werkzeuge.....	176
3.1.10	Einstellungen.....	183
3.1.11	Fenster.....	217
3.1.12	Hilfe.....	217
<b>3.2</b>	<b>Tools .....</b>	<b>218</b>
3.2.1	Objektinspektor.....	218
3.2.2	Quelltexteditor .....	223
3.2.3	Quelltext vervollständigen.....	228
3.2.4	Message Composer.....	231
3.2.5	Debugger.....	232
<b>3.3</b>	<b>IDE rekompilieren .....</b>	<b>239</b>

---

## Kapitel 4

### Projekte..... 245

von Felipe Monteiro de Carvalho

<b>4.1</b>	<b>GUI-Anwendungen .....</b>	<b>248</b>
<b>4.2</b>	<b>Konsolenanwendungen .....</b>	<b>253</b>
<b>4.3</b>	<b>DLLs und Shared Objects.....</b>	<b>258</b>
4.3.1	Dynamische Bibliotheken.....	259
4.3.2	Bibliotheken in MacOS X .....	265
4.3.3	Systemsteuerungs-Applets für Windows.....	266
<b>4.4</b>	<b>Unit-Testing.....</b>	<b>275</b>
<b>4.5</b>	<b>Packages .....</b>	<b>276</b>
4.5.1	Komponenten einbinden .....	281
4.5.2	Komponenten registrieren .....	281
4.5.3	Eigenschafts-Editoren (Property-Editoren) .....	282
4.5.4	Komponenten-Editoren.....	286
<b>4.6</b>	<b>Dienste und Daemonen.....</b>	<b>289</b>

## **Kapitel 5 Zielplattformen ..... 293**

von Felipe Monteiro de Carvalho

<b>5.1 Plattformspezifisches .....</b>	<b>297</b>
5.1.1 Das API von Windows32/64 .....	299
5.1.2 Spezialfall Windows CE .....	305
5.1.3 Linux, FreeBSD und andere Unix-Systeme .....	317
5.1.4 Die APIs von MacOS X.....	324
5.1.5 32 Bit und 64 Bit .....	338
<b>5.2 Konfigurationsdateien .....</b>	<b>339</b>
<b>5.3 Ressourcendateien .....</b>	<b>342</b>

## **Kapitel 6 Die Klassenbibliotheken..... 347**

von Michaël Van Canneyt

<b>6.1 Die RTL (FPC-Laufzeitbibliothek) .....</b>	<b>349</b>
6.1.1 Laden und Speichern mit Datenströmen.....	350
6.1.2 Nomenklatur der Klasseninstanzen .....	352
<b>6.2 Die FCL (Free Component Library) .....</b>	<b>354</b>
<b>6.3 Die LCL (Lazarus Component Library) .....</b>	<b>356</b>
6.3.1 Das Anwendungsobjekt .....	359
6.3.2 Bildschirmfenster .....	374
6.3.3 Arbeiten mit TForm.....	375
6.3.4 Die Eigenschaften von TForm.....	383
6.3.5 Spezialfenster .....	395
6.3.6 Die Fensterumgebung .....	402
6.3.7 Die Steuerelemente im Fenster .....	405
6.3.8 Layout und Programmwurf.....	418
6.3.9 Actions .....	428
6.3.10 Drag&Drop .....	434
6.3.11 Drag&Dock .....	445
<b>6.4 Die Elemente der Komponentenpalette .....</b>	<b>455</b>
6.4.1 Der Reiter Standard .....	459
6.4.2 Der Reiter Additional .....	480

---

6.4.3 Der Reiter Common Controls.....	491
6.4.4 Der Reiter Dialogs .....	511
6.4.5 Der Reiter Misc .....	516
6.4.6 Der Reiter Data Controls.....	521
6.4.7 Der Reiter Data Access .....	522
6.4.8 Der Reiter System .....	522
6.4.9 Der Reiter SynEdit .....	523

---

## Kapitel 7

### **Delphi-Komponenten portieren ..... 525**

von Michaël Van Canneyt und Mattias Gärtner

7.1 Die Architektur von Lazarus-Komponenten .....	525
7.1.1 Plattformunabhängige Schicht .....	526
7.1.2 Plattformabhängige Schicht .....	528
7.2 Komponentenmodelle .....	529
7.3 Praxis der Portierung .....	537
7.3.1 Von der Komponente zum Lazarus-Package .....	541
7.3.2 Die Komponentenpalette .....	543

---

## Kapitel 8

### **Dateien und Geräte ..... 545**

von Felipe Monteiro de Carvalho und Jörg Braun

8.1 Dateialoge in Lazarus.....	545
8.2 Arbeiten mit Dateien .....	554
8.3 Verzeichnisse durchsuchen .....	573
8.4 Kommunikation mit Geräten .....	579
8.4.1 Der Parallelport .....	580
8.4.2 Serielle Kommunikation.....	585
8.4.3 Der Drucker .....	590

## Kapitel 9 Grafikprogrammierung..... 595

von Felipe Monteiro de Carvalho

<b>9.1 Die Zeichenfläche.....</b>	<b>595</b>
9.1.1 Farben .....	598
9.1.2 TPen.....	600
9.1.3 TBrush.....	604
9.1.4 Fonts.....	607
9.1.5 Die wichtigsten Grafikroutinen .....	610
<b>9.2 Grafikkomponenten.....</b>	<b>615</b>
<b>9.3 Die Grafikformate.....</b>	<b>620</b>
9.3.1 TGraphic.....	620
9.3.2 TRasterImage .....	621
9.3.3 Bitmaps.....	624
9.3.4 TjpegImage .....	625
9.3.5 Icons .....	626

## Kapitel 10 Prozesse und Threads..... 631

von Felipe Monteiro de Carvalho

<b>10.1 Prozesse.....</b>	<b>631</b>
10.1.1 CreateProcess und die Komponente TProcess .....	632
10.1.2 ExecuteProcess .....	638
<b>10.2 Threads .....</b>	<b>639</b>

## Kapitel 11 Netzwerk-Programmierung ..... 647

von Inoussa Ouedraogo

<b>11.1 TCP/IP programmieren.....</b>	<b>647</b>
11.1.1 Das Clientprogramm.....	649



11.1.2 Das Serverprogramm .....	652
<b>11.2 Webservices .....</b>	<b>654</b>
11.2.1 Programmierung des Servers.....	656
11.2.2 Programmierung des Clients .....	672
11.2.3 Nachrichten-Logging .....	677
11.2.4 Objektpooling .....	678
11.2.5 Serviceerweiterungen .....	681

## Kapitel 12

### Der Zugriff auf Datenbanken ..... 683

von Michaël Van Canneyt

<b>12.1 Die Architektur.....</b>	<b>683</b>
12.1.1 Datenbankzugriff.....	683
12.1.2 Datenbank auswählen.....	685
12.1.3 Hilfsanwendung: Der Daten-Desktop .....	689
<b>12.2 Die Datenbank-Zugriffsklassen .....</b>	<b>690</b>
12.2.1 Das Dataset .....	690
12.2.2 Das Datenmodul .....	713
12.2.3 Visuelle datenbezogene Steuerelemente .....	715
<b>12.3 Ableitungen von TDataSet.....</b>	<b>717</b>
<b>12.4 Der Daten-Desktop .....</b>	<b>735</b>
12.4.1 Das Data Dictionary (Daten-Verzeichnis).....	737
12.4.2 Daten exportieren.....	742
12.4.3 Codegenerierung .....	744
<b>12.5 Crashkurs SQL .....</b>	<b>746</b>
<b>12.6 Berichte (Reports).....</b>	<b>750</b>
12.6.1 Bericht anlegen .....	752
12.6.2 Der Report-Designer.....	755

### Stichwortverzeichnis ..... 757



# VORWORT

Liebe Leser,

es gibt sicherlich gute Gründe, warum Sie dieses Buch, das sich nun schon in der zweiten Auflage befindet, in Händen halten. Vielleicht möchten Sie plattformunabhängige Programme entwickeln und suchen eine moderne und wirklich bedienbare IDE, wie es sie für Perl und Java nicht gibt. Oder Sie sind bereits passionierter Pascal-Programmierer und sind froh, eine Alternative zu Delphi gefunden zu haben, weil Sie damit schon länger nicht mehr zufrieden sind. Das Schreiben 64-bitiger Windows-Programme verwehrt Ihnen Embarcadero und Plattformen wie Unix oder Windows CE können Sie damit auch nicht bedienen. Außerdem haben Sie die Zwangsregistrierung satt.

Die Entwickler von Lazarus und Free Pascal kennen diese Probleme und haben einen technisch ausgereiften Compiler und eine zukunftsorientierte IDE gebaut, die nicht nur Delphi in vielerlei Hinsicht alt aussehen läßt. Und das sogar ohne Kosten für Anwender (auch nicht versteckte) und als Open Source!

Im C&L Verlag habe ich die Entwicklung dieser Projekte die letzten Jahre intensiv beobachtet und gefördert und mich oft gewundert, wie eine Handvoll Leute es schafft, in ihrer Freizeit solch umfassende und komplexe Software auf die Beine zu stellen. Heute freue ich mich, daß ich die mittlerweile große Lazarus-Community mit diesem Buch unterstützen kann und hoffe, daß viele neue Entwickler aufgrund dieses Lehr- und Nachschlagebuchs zu Lazarus wechseln werden. Die, die bereits damit arbeiten, werden darin neue Anregungen und Informationen finden.

An diesen beiden Entwicklergruppen haben wir das Buch auch ausgerichtet: Es hilft sowohl beim Einstieg in Lazarus, weil es sein Konzept vermittelt und beschreibt, wie man mit der IDE arbeitet. Derjenige, der noch nie mit einer Pascal-RAD zu tun hatte oder nur wenig Erfahrung damit hat, wird darin lernen, wie er seine eigenen Programme entwerfen und mit den Elementen der Klassenbibliothek ausstatten muß. Die Umsteiger von Delphi lesen in diesem Buch, wie sie plattformunabhängige Programme schreiben und bereits vorhandene Delphi-Projekte umziehen. Eigentlich ist es nicht erstaunlich, wie wenig dabei zu beachten ist, schließlich wurde Lazarus von vornherein auf Plattformunabhängigkeit ausgelegt. Weil auch an den entsprechen-

den Stellen auf die Unterschiede zwischen Lazarus und Delphi hingewiesen wird, werden sich Umsteiger schnell in Lazarus zurechtfinden.

Die Autoren dieses Buchs sind alle Mitglieder des Lazarus- beziehungsweise FreePascal-Teams. Ihre Namen werden einigern Lesern sicherlich von der Aboutbox von Komponenten oder Tools bekannt sein. Ich bin sehr stolz, daß ich solche Experten als die Verfasser dieses Buchs gewinnen konnte.

Der Belgier Michaël Van Canneyt ist Core-Mitglied in der FreePascal.Org und hat über die LCL und die Datenbankbindung geschrieben. Er hat das Buch mitkonzipiert und hat mich bei meinen technischen Fragen stets kompetent und geduldig beraten. Mattias Gärtner als einer der Haupt-Maintainer von Lazarus gibt sein profundes Wissen über die hinter Lazarus stehende Technik und Philosophie an die Leser weiter. Wenn ich mich hilfeschend an ihn wandte, wenn ich im Themendickicht des Manuskripts nicht mehr weiter wußte, hat er mir immer wieder klarsichtig den Weg gewiesen. Swen Heinig, der Betreuer der deutschen Lazarus-Wiki-Seiten, und neuer deutscher Übersetzer der IDE hat die Aufgabe auf sich genommen, die Entwicklungsumgebung zu beschreiben. Ein Unterfangen, das ständiges Am-Ball-Bleiben und Aktualisierungen erforderte, schließlich entwickelt sich Lazarus permanent weiter. Felipe Monteiro de Carvalho aus Brasilien zeichnet bei Lazarus unter anderem verantwortlich für die MacOS-Brücke, als Wanderer zwischen den Welten hat er hier unter anderem über die Plattformunabhängigkeit und Systemnahes geschrieben. Er und der Netzwerkspezialist Inoussa Ouedraogo, der in Mali wohnt, aus Burkina Faso stammt und über sein Fachthema in diesem Buch berichtet, haben mich in trüben Tagen mit den Wetterberichten aus ihrer Heimat bei Laune gehalten. Jörg Braun, der einen großen Teil der Lazarus-IDE ins Deutsche übersetzte, hat geschrieben, wie Lazarus auf seiner Lieblingsplattform FreeBSD installiert wird und hat an einigen Stellen Manuskripte ergänzt. Der Holländer Marc Weustink hat sein Wissen zum Debugger weitergeben. Ein Teil der Manuskripte wurde von Dr. Uwe Barlage und Wolfgang Soltendick vom Englischen ins Deutsche übersetzt.

Ihnen allen danke ich ganz herzlich für die viele Mühe und Arbeit, die sie auf sich genommen haben, um dieses einmalige Buch exklusiv für der C&L Verlag zu verfassen!

Ich wünsche auch Ihnen viel Spaß beim Schmökern und Entdecken der Fähigkeiten von Lazarus!



Rosa Riebl  
Verlegerin (info@cul.de)

# KAPITEL 1

## DIE ARCHITEKTUR VON LAZARUS

von *Mattias Gärtner*

Lazarus ist eine integrierte Entwicklungsumgebung für die Sprache Object Pascal. Das Backend von Lazarus bildet der Free Pascal Compiler. Diese Kombination aus Backend und integrierter Entwicklungsumgebung ist vergleichbar mit Visual Studio für C++ oder Borland Delphi. Das Ergebnis ist ein effizienter, hochoptimierter nativer Quelltext, ganz im Gegensatz zu Java- oder C#-Entwicklungsumgebungen, die Bytecode produzieren, der erst von einer Laufzeitumgebung interpretiert werden muß oder vom einem JIT-Compiler in wenig optimierten Code umgewandelt wird.

Eine Stärke der Lazarus-IDE sind ihre RAD-Fähigkeiten, mit denen grafische Programme sehr einfach und schnell geschrieben werden können. Sie ähneln in weiten Teilen Delphi. Während Delphi aber auf dem Delphi-Compiler basiert, ist bei Lazarus der Free Pascal Compiler FPC hinterlegt, der weitgehend kompatibel mit Delphi 2007 ist. Allerdings läuft Lazarus nicht nur unter Windows, sondern nativ auch auf Linux, MacOS X, FreeBSD und mehr Plattformen. Der Free Pascal Compiler ist selbst in FPC geschrieben (genau so wie Lazarus auch komplett in Lazarus geschrieben ist). FPC enthält auch Spracherweiterungen aus anderen Pascal-Dialekten, weshalb bereits vorhandener Quelltext einfach portiert werden kann. Da großer Wert auf die Plattformunabhängigkeit gelegt wird, wird Code, der möglicherweise nicht portabel ist, von FPC erkannt und der Compiler warnt davor.

Die grafische Komponentenbibliothek heißt in Lazarus »Lazarus Component Library«, LCL. Viele Units, Klassen und Bezeichner heißen und funktionieren wie bei Delphi, damit eine Portierung von Delphi-Code nach Lazarus möglich ist. Lazarus ist aber kein Delphi-Clone, sondern um Plattformunabhängigkeit erweitert, und zwar ohne Emulator, sondern mit nativen Anwendungen. Der Preis, den man dafür zahlen muß, ist, daß eine hundertprozentige Kompatibilität zur VCL nicht möglich ist. Die LCL verfügt über native Komponenten, die Applikationen sehen aus wie andere native Anwendungen und werden auch so bedient. Dies gelingt der LCL aufgrund nativer Widgetsets wie dem WinAPI unter Windows, dem Gtk 2 oder QT unter Linux/BSD und Carbon/Cocoa unter OS X.

Die wichtigsten Unterschiede zu Delphi sind, daß Lazarus komplett Open Source ist, auf inzwischen mehr als fünfzehn Plattformen läuft, auch für kommerzielle Anwender kostenlos ist und auf einer flexiblen Lizenz basiert. Durch die Open-Source-Gemeinschaft wurden inzwischen nicht nur viele Delphi-Bibliotheken auf Lazarus portiert und für die anderen Plattformen erweitert, sondern auch Bibliotheken geschrieben, die es exklusiv nur für Lazarus gibt. In zahlreichen Foren, Mailinglisten und Webseiten ist schnell und unkompliziert Hilfe zu finden.

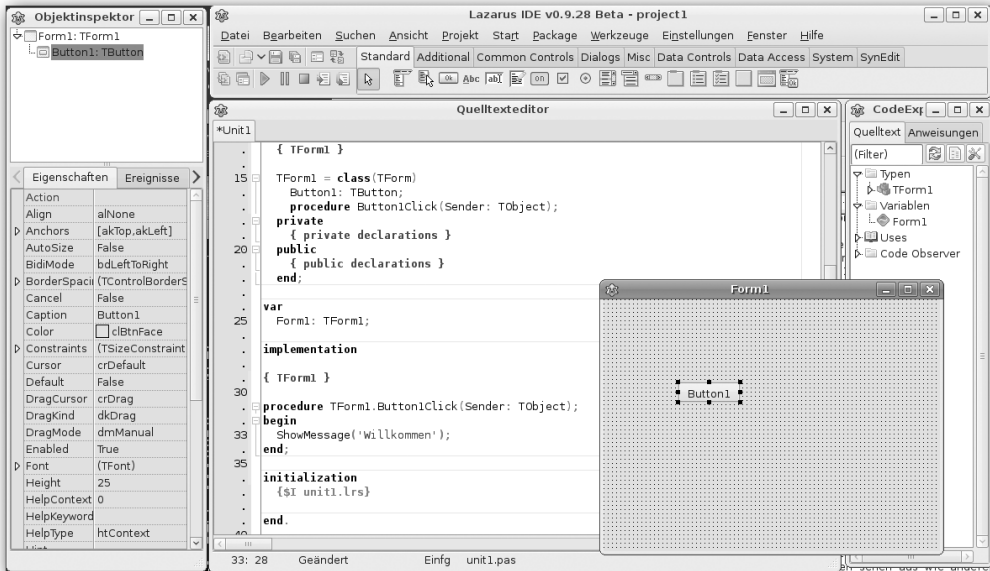


Bild 1.1: Delphi-Anwender werden vieles bereits kennen: Formulareditor, Objektinspektor, CodeExplorer

Der Free Pascal Compiler und seine mitgelieferten Bibliotheken sind das solide Fundament von Lazarus. Die beiden sind zwar eng miteinander verzahnt, aber dennoch zwei unterschiedliche Projekte. Die erste Version von Free Pascal entstand 1993, während die Arbeiten an Lazarus 1999 begannen.

Free Pascal ist ein 32- und 64-bittiger Compiler für Windows 32, Windows 64, WinCE, Linux, FreeBSD, MacOS X/Darwin, DOS, OS/2, Netware (libc und classic) und MorphOS, für die Architekturen Intel x86, AMD64/x86\_64, PowerPC, PowerPC64, Sparc und Arm. Wie Delphi ist der Compiler sehr schnell, er ist jedoch nicht so extrem schnell wie Delphi, weil FPC für leichte Erweiterbarkeit auf mehr als nur eine Plattform optimiert ist.

Der entstandene Code ist unter Windows in Größe und Geschwindigkeit vergleichbar mit Delphi (mal ist FPC besser, mal Delphi), unter Linux/MacOS X mit dem

GNU C/C++ Compiler. Es gibt derzeit keine .NET-Unterstützung und keine Packages wie in den bpl-Bibliotheken von Delphi. Normale Bibliotheken können kompiliert und eingebunden werden, und das auf allen Plattformen.

Mit Lazarus lassen sich sowohl native grafische Anwendungen als auch Windows-Dienste, Linux-Daemonen, Konsolenwerkzeuge, Bibliotheken, Plugins, Datenbank-anwendungen, Webanwendungen und vieles mehr programmieren. Die Funktionalität der IDE läßt sich durch Designtime-Packages erweitern.

Lazarus erlaubt Crosskompilierung, also zum Beispiel das Erzeugen von Windows-Programmen unter Linux oder WinCE-Anwendungen unter Windows. Die Codenavigation und Codevervollständigung funktionieren dann entsprechend. Umgeschaltet wird in der IDE, ohne daß sie neu gestartet werden muß.

Der Compiler kennt bedingte Kompilierung, mit der spezieller Code je nach Betriebssystem, CPU oder nach anderen Kriterien geschrieben werden kann. Die IDE erlaubt Makros in allen Suchpfaden, so daß zum Beispiel Units und Include-Dateien je nach Betriebssystem ausgewählt werden können, was gerade bei mehr als zwei Plattformen für eine saubere und übersichtliche Struktur sorgt; die LCL selbst basiert auf diesem Konzept. Dem Programmierer stellt sich die LCL wie eine plattform-unabhängige Sammlung von Komponenten dar, während durch Makros in den Suchpfaden je nach Betriebssystem maßgeschneiderte Backend-Units auf das native Widgetset zugreifen.

Durch das Package-Konzept können Units zu größeren Einheiten zusammengefaßt werden, die ihre eigenen CompilerEinstellungen besitzen. Das stellt sicher, daß Code sehr einfach auf andere Computer mit anderen Pfaden und anderen Betriebssystemen übertragen werden kann. Dazu müssen einfach die Dateien kopiert und das Package geöffnet werden. Es müssen keine Pfade angepaßt oder andere Schalter gepflegt werden, was das Aktualisieren von Packages erheblich erleichtert.

Die visuellen Komponenten der LCL sind ereignisgesteuert. Komplette Anwendungen müssen also nicht mehr Schritt für Schritt geschrieben werden, sondern man entwirft Methoden, in denen auf Ereignisse reagiert wird. Der Zustand der LCL-Komponenten wird komplett über Properties festgelegt. Diese können in nahezu beliebiger Reihenfolge verändert werden und in der IDE können Formulare komfortabel grafisch zusammengestellt werden. Sollte einmal eine Funktion fehlen, kann jede Komponentenklasse als Basis für eigene Klassen genommen und in der IDE wiederum grafisch benutzt werden. Auch eigene Komponentenbibliotheken können geschrieben werden.

In der IDE können Fenster grafisch editiert werden. Für jede Unit kann es höchstens ein Fenster geben. Fenster können auf anderen Fenstern basieren – Stichwort Vererbung – oder als Unterfenster ineinander verschachtelt werden (Frames). Die Eigenschaften werden in lfm-Dateien gespeichert. Natürlich können Fenster auch per Code erzeugt werden. Der Clou ist, daß der Formulareditor eng mit dem Quelltexteditor zusammenarbeitet. Wird eine neue Komponente auf das Fenster abgelegt,

wird eine neue Variable im Quelltext erzeugt. Wird das *OnClick*-Ereignis eines Knopfes gesetzt, wird eine neue Methode im Quelltext erzeugt. Weitere Schritte sind nicht nötig, das Programm kann sofort kompiliert und gestartet werden.

Neben einer einfachen Syntax muß eine Sprache auch ein Konzept für das Kombinieren von Dateien besitzen. Dies ist bei FPC sehr bequem gelöst. Bei anderen Sprachen wie C/C++ wird häufig anhand von Makedateien das Kompilieren von mehreren Dateien festgelegt. Da das Erzeugen und Pflegen guter Makedateien nicht leicht ist, gibt es unzählige Werkzeuge, die beim Erzeugen von Makedateien helfen.

Die gute Nachricht für FPC-Programmierer: Oft braucht man keine Makefiles. Die bessere: Natürlich kann man sie trotzdem nutzen. Die beste: Es gibt Werkzeuge, sie automatisch erzeugen zu lassen.

Die Aufgabe einer Makedatei ist es, den Kompilierungsprozeß zu vereinfachen und zu vereinheitlichen. Dabei sollen automatisch die veränderten Dateien in der richtigen Reihenfolge kompiliert und die nötigen Compilerparameter zusammengestellt werden.

Der Free Pascal Compiler kompiliert automatisch alle veränderten Units, macht dies erheblich schneller als `make` und arbeitet mit Prüfsummen. Die Prüfsummen sind insbesondere wichtig, wenn sich die Dateien in Netzwerkverzeichnis befinden und die Uhrzeiten zwischen Client und Server leicht abweichen. Manche C-Programmierer kennen die seltsamen Warnungen, daß etwas mit der Uhrzeit nicht stimmen würde. FPC hat damit weniger Probleme, weshalb man FPC nur die Hauptdatei übergibt und den Compiler den Rest selbst suchen läßt.

Beispielsweise ruft

```
fpc programm1.pas
```

automatisch den richtigen Compiler auf. Zum Crosskompilieren auf eine andere Prozessorarchitektur wird ein anderer Compiler benötigt. Der andere Compiler ist natürlich auch ein FPC, aber kompiliert für die Zielarchitektur. Der Compiler liest als nächstes die Quelldatei, und wenn er dabei auf eine *uses*-Klausel stößt, sucht er automatisch nach den entsprechenden Units. Findet er fertig kompilierte .ppu-Dateien, nutzt er diese. Findet er nur Quelldateien oder sind die Quelldateien neuer als die .ppu-Dateien, so kompiliert er diese. Benötigen auch diese weitere Units, werden diese rekursiv kompiliert. Dies alles geschieht rasend schnell, normalerweise werden zehn- bis hunderttausend Zeilen pro Sekunde übersetzt. Dabei sucht der Compiler automatisch die Dateien und kompiliert automatisch alles Veränderte in der richtigen Reihenfolge.

Die Compilerparameter sind im Gegensatz zu denen bei C-Compilern meist überschaubar. Das liegt zum einen daran, daß viele Einstellungen bereits im Quellcode stehen, und zum anderen, daß der Compiler und die Systembibliotheken auf allen

Plattformen aus den gleichen Quellen stammen. Dadurch verkürzen sich die Parameter meist auf die Angabe von Suchpfaden und das berühmte `./configure` zum Sammeln der Systemparameter entfällt. Daher das Motto: Write once, compile anywhere. Beispielsweise prüft der Aufruf von `configure` unter C unter anderem, welche Größe ein `int` hat. Bei Free Pascal sind die Typen direkt aus dem Code ersichtlich. Tests, welche `libc`-Version genutzt wird, sind unter Free Pascal unnötig, da die Runtime Library nur Kernelfunktionen nutzt, die sich erfahrungsgemäß nur sehr selten ändern. Ein Pascal-Programm läuft daher auf mehr Distributionen als ein vergleichbares C-Programm. Da Pascal die C-Headerdateien nicht direkt benötigt, sind genaue Tests, welche Bibliotheken genutzt werden, unnötig.

Lazarus erweitert das Unit-Konzept um Packages und Projekte. Ein Package ist eine Sammlung von Dateien mit der Information, wie man sie kompiliert und welche Packages dafür benötigt werden. Ein Package kann eine Bibliothek sein oder einfach nur eine logische Einheit innerhalb eines umfangreichen Projektes. Ein Lazarus-Projekt ist wie ein Package eine Sammlung von Dateien, allerdings mit zusätzlichen Informationen, die zum Bau eines Programms benötigt werden. Darüber hinaus speichert die IDE zu jedem Projekt, welche Dateien im Quelltexteditor geöffnet sind, Informationen zum Debugging und mehr. Man könnte also sagen, ein Lazarus-Projekt beziehungsweise Package ist eine Makedatei plus Zusätze.

Auf diese Weise lassen sich selbst umfangreiche und voneinander unabhängige Bibliotheken ohne Makefiles kompilieren. Zuständig ist das Kommandozeilenprogramm `lazbuild`.

Beispielsweise hat ein Projekt immer eine Datei `projektname.lpi`, in der alle Metainformationen stehen. Bei der `.lpi`-Datei handelt es sich um eine XML-Datei und alle Informationen darin können komfortabel über die IDE eingestellt werden. Ein manuelles Editieren ist also unnötig, außer man möchte Hunderte Dateien auf einmal umbenennen. Hinzu kommt typischerweise eine Hauptdatei, deren Namen und Endung beliebig ist. Die IDE nennt die Hauptquelldatei standardmäßig `projectname.lpr`.

In den letzten Abschnitten wurde grob die Konzepte erklärt, wie unter Lazarus Dateien zu größeren Einheiten, den Projekten und Packages verbunden werden. Durch die Unabhängigkeit lassen sich Packages von sehr unterschiedlichen Programmierern und Drittanbietern unter einen Hut bringen. Der nächste Abschnitt zeigt, wie der Lazarus-Quelltexteditor mit diesem Wissen beim Programmieren hilft.

## 1.1 DER QUELLTEXTEDITOR

Der Quelltexteditor ist das Paradebeispiel für das »Integrierte« in der IDE. Er basiert auf der SynEdit-Komponente und nutzt die CodeTools zum projektübergreifenden Navigieren und Ändern. Der Formulareditor, der Debugger und der `fpdoc`-Editor arbeiten ebenfalls eng mit dem Editor zusammen. Er ist eine der Stärken von Lazarus und geht in vielen Teilen über den Editor von Delphi hinaus.

Allen Funktionen können Tastenkürzel zugeordnet werden, die unter *Einstellungen* → *Einstellungen* → *Editor* → *Tastaturbelegung* definiert werden. Tastaturkürzel können aus ein oder zwei Tasten bestehen und die Modifizierertasten beliebig kombinieren. Unter Windows sind die Modifizierertasten [Strg], [Shift] und [Alt], unter MacOS X kommt [Cmd] hinzu.

### 1.1.1 CodeTools

Die IDE liest Quellcode nicht mit dem Compiler, sondern anhand der CodeTools. Im Vergleich mit einem Compiler besitzen sie folgende Vorteile:

- Sie sind fehlertoleranter, insbesondere gegenüber fehlenden Units.
- Sie können Delphi-, Kylix- und FPC-Code lesen.
- Sie können in jedem Verzeichnis mit eigenen Compilerparametern arbeiten.
- Cross Editing: Mit den CodeTools kann auf der einen Plattform Code einer anderen Plattform bearbeitet werden, so als ob auf einer anderen Plattform gearbeitet würde. Es kann also beispielsweise Windows-Code unter Linux auch ohne installiertem Cross-Compiler bearbeitet werden.
- Geschwindigkeit: Wird bei großen Projekten an einer Basis-Unit gearbeitet, muß ein Compiler alle abhängigen Units neu kompilieren. Die CodeTools sind optimiert und aktualisieren nur, was gerade benötigt wird.
- Die Codetools funktionieren auch in Kommentaren und in durch IFDEFs deaktiviertem Code. Allerdings ist die Syntaxprüfung in Kommentaren weniger sicher.

### 1.1.2 Schnelle Navigation im Code

Mit der Tastenkombination [Strg]+[Shift]+[↑⇌] oder [Strg]+[Shift]+[↓] kann zwischen der Deklaration einer Prozedur und der Implementation hin- und hergesprungen werden. Mit der Funktion *Suche Prozedurdeklaration/-körper* kann zwischen Methoden, Prozeduren im Interfaceteil und Forward-Prozeduren gesprungen werden, sie ist also fast überall anwendbar:

```
procedure MachWas(i: Integer);    // von hier ...
```

**implementation**

```
procedure MachWas(i: Integer);
begin
    // ... nach hier und umgekehrt
end;
```

Wird keine passende Prozedur gefunden, wird die gesucht, die am besten paßt, und der Cursor auf die erste Abweichung positioniert. Wurde zum Beispiel der obige Parameter *i* in der Deklaration auf *Parameter1* umbenannt, kann mit [Strg]+[Shift]+[↓] zur Implementation gesprungen werden. Die IDE setzt den Cursor auf die erste Abweichung, also auf das *i*. Dort kann dann die Variable umbenannt werden.



```
procedure MachWas(Parameter1: Integer); // von hier ...
```

**implementation**

```
procedure MachWas(i: Integer);
begin
end;
```

Mit der Funktion *Suche Deklaration* mit der Tastenkombination [Alt]+[↑] wird die Deklaration eines Bezeichners gesucht, es wird stets zur nächsten Deklaration gesprungen.

Ein Beispiel:

```
type
  TMyInteger = 1..2; // findet keinen weiteren
  TMyInteger = 1..3; // springt zum ersten TMyInteger

var
  i: TMyInteger; // springt zum zweiten TMyInteger
```

Bei Vorwärtsdeklarationen wird zur eigentlichen Deklaration gesprungen. Bei Bezeichnern, die eine Vorwärtsdeklaration nutzen, wird normalerweise zu dieser Vorwärtsdeklaration gesprungen, da sie am nächsten liegt. Soll die IDE direkt zur eigentlichen Deklaration springen, kann dies in den CodeTools eingestellt werden.

```
type
  THund = class; // springt nach unten zur eigentlichen Deklaration

  TPfote = class
  public
    Hund: THund; // springt nach oben
                // oder direkt nach unten, je nach Einstellung
  end;

  THund = class
  public
    Pfoten: array[1..4] of TPfote;
  end;
```

*Parameterhinweise* [Strg]+[Shift]+Leertaste:

Wenn der Cursor in einer Parameterliste steht, können mit Parameterhinweisen die Deklarationen eingeblendet werden.

Steht der Cursor im Quelltext

```
fs := TFileStream.Create( );
```

innerhalb der Klammern der Konstruktorparameterliste, zeigen die Parameterhinweise alle Konstruktoren der Klasse *TFileStream* an, einschließlich der Parameter-

listen. Der aktuelle Parameter unter dem Cursor ist jeweils hervorgehoben. Die Hinweise verschwinden mit der Escape-Taste oder wenn der Cursor die Klammern verläßt.

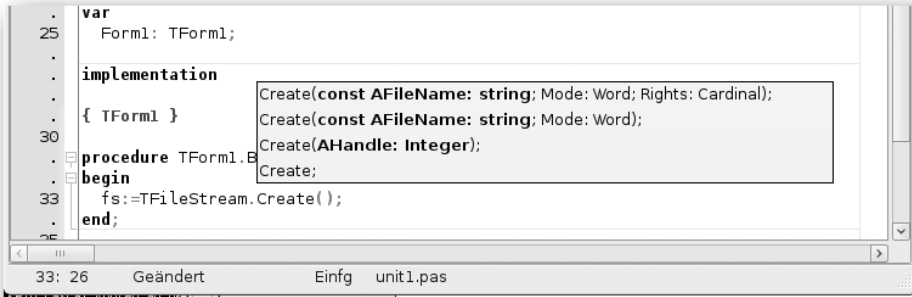


Bild 1.2: Mit [Strg]+[Shift]+ [Leertaste] lassen sich die Parameterhinweise einblenden

### Code Explorer:

Der Code Explorer ist ein separates Fenster und zeigt wie unter Delphi alle Pascal-Bezeichner der aktuellen Unit an. Diese werden entweder alphabetisch oder nach Kategorien sortiert. Zusätzlich wird auf einer zweiten Seite noch ein Direktiven-Browser ausgegeben, der einen Überblick über alle Direktiven im Quelltext liefert. Das erleichtert es, konditionalen Code zu durchsuchen oder include-Direktiven zu finden.

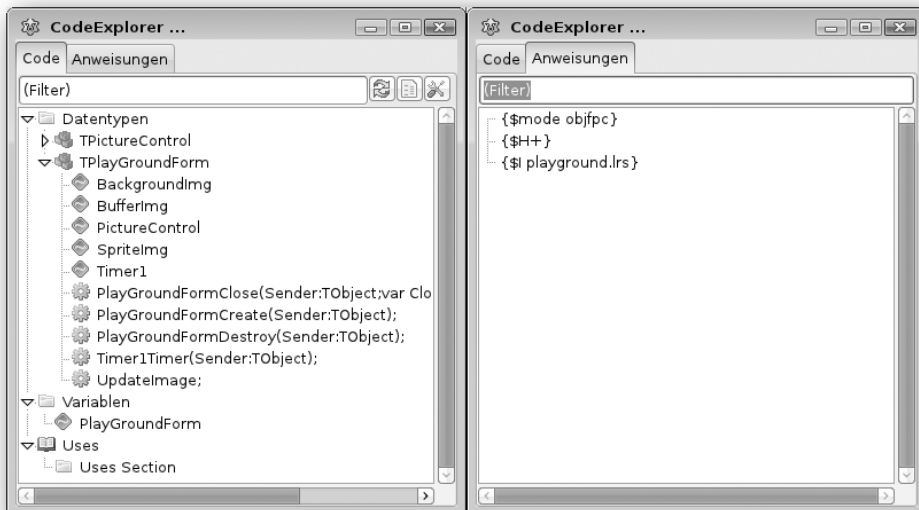


Bild 1.3: Die beiden Seiten des Code-Explorers

Seit Lazarus 0.9.28 gibt es im CodeExplorer einen neuen Unterpunkt namens *Code-Observer*. In ihm werden Codestellen aufgelistet, die möglicherweise unsauber programmiert wurden.

### 1.1.3 Automatische Vervollständigung

*Bezeichner-Vervollständigung:*

Es wird eine Liste von Methoden oder Eigenschaften der Klasse angezeigt. Die Liste wird während des Eintippens der Buchstaben des Methodennamens gefiltert. Der Anwender kann dann mit der Maus oder den Pfeiltasten der Tastatur durch die Liste wandern und die richtige Methode aussuchen. Diese Funktionalität wird mit dem Betätigen der [Strg]- und Leertaste eingeschaltet.

Steht im folgenden Quelltext

```
var
  L : TStrings;

begin
  for I:=0 to L.
```

der Cursor auf dem ».« nach dem Bezeichner *L*, wird nach dem Betätigen der [Strg]- und Leertaste eine Liste von Eigenschaften und Methoden der Klasse *TStrings* angezeigt (die Variable *L* ist vom Typ *TStrings*).

*Bearbeiten* → *Quelltext vervollständigen* [Strg]+[Shift]+[C]:

Die Funktionalitäten der Codevervollständigung hängen von der Position des Cursors ab. Befindet sich der Cursor im Deklarationsabschnitt einer Klasse, ist die Klassenvervollständigung aktiviert. Die IDE versucht dann, die Klassendeklaration zu vervollständigen und leere Implementierungen für alle Prozeduren in der Deklaration zu erzeugen. Dabei erkennt sie Property-Getter/-Setters und errät aufgrund intelligenter Heuristiken, was in Felder oder Methoden konvertiert werden soll.

Wird beispielsweise in der Deklaration

```
type
  TMyComponent = class(TComponent)
  public
    property Active: Boolean Read FActive Write SetActive;
  end;
```

die Codevervollständigung mit der Tastenkombination [Strg]+[Shift]+[C] angeschaltet, ändert die IDE den Quelltext folgendermaßen:

```
TMyComponent = class(TComponent)
private
  FActive: Boolean;
```

```

    procedure SetActive(const AValue: Boolean);
public
    property Active: Boolean Read FActive Write SetActive;
end;

{ TMyComponent }
procedure TMyComponent.SetActive(const AValue: Boolean);
begin
    if FActive = AValue then Exit;
    FActive := AValue;
end;

```

Sie erkennt also, daß *SetActive* eine Methode und *FActive* ein Feld ist, das es voreingestellt im *private*-Abschnitt der Klassendeklaration erzeugt. Sie legt auch den Körper der Methode *SetActive* an und füllt ihn mit einem Anfangscode. Sie macht das aber natürlich nur für Property Setters, also Methoden, die hinter dem Schlüsselwort *write* stehen. Obwohl das Beispiel nur eine einzige Eigenschaftendefinition enthält, macht die IDE das gleiche mit jeder beliebigen unvollständigen Methode oder Eigenschaft, auf die sie stößt.

Auch die umgekehrte Operation ist möglich. Wird der Implementation der Klasse folgende Prozedur hinzugefügt

```

procedure TMyComponent.CheckActive;
begin
    if not Active then RaiseException;
end;

```

sorgt das Betätigen der Taste für die Codevervollständigung mit dem Cursor an einer beliebigen Stelle zwischen den Schlüsselworten *procedure* und *end* dafür, daß der Klassendeklaration die Prozedurdeklaration hinzugefügt wird:

```

TMyComponent = class(Tcomponent)
private
    FActive: Boolean;
    procedure CheckActive;
    procedure SetActive(const AValue: Boolean);
public
    property Active: Boolean Read FActive Write SetActive;
end;

```

Eine weitere Art der Codevervollständigung ist die Variablenvervollständigung. Steht im Quelltext

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    for i := 0 to L.Count - 1 do
end;

```

# STICHWORTVERZEICHNIS

## \$

\$(LazarusDir) .....	54
\$(TargetOS) .....	295
\$apptype console/gui .....	254
\$ifdef ... \$endif .....	294
\$ifdef MSWindows .....	60
\$ifdef-Konstruktionen vermeiden .....	28
\$include .....	294
.compiled-Datei .....	252
.conf-Datei .....	344
.cpl-Datei .....	266
.dpr-Datei .....	53
.exe-Datei .....	140
.ico-Datei .....	250
.inc-Datei .....	50
.ini-Datei .....	344
.lfm-Datei .....	17, 139, 218, 60, 250, 350
.lpi-Datei .....	19, 53, 140, 249, 250
.lpk-Datei .....	54, 72, 278
.lpr-Datei .....	19, 53, 139, 146, 249f, 359
.lrs-Datei .....	139, 250, 252
.manifest-Datei .....	250
.o-Datei/..or-Datei .....	252
.pas-Datei .....	139, 250
.po-Datei .....	54, 146, 77
.ppu-Datei .....	54, 139, 252
.rc-Datei .....	250, 252
.res-Datei .....	252
.rst-Datei .....	54, 77
.xsd-Datei .....	673
32 Bit   64 Bit, Prozessor .....	338
32-Bit-Programm .....	299
64-Bit-Betriebssystem, RAM .....	299
64-Bit-Lazarus .....	100
8-Bit-ISO-Zeichen .....	83

## A

Abbrechen-Button .....	465
—, Ergebnis .....	379
Abfragedialog, benutzerdef. ....	400
Abfrageelemente .....	416
Abfragen, tabellenübergreifende ..	686
Abstrakte Deklaration/Methode ..	43f.
Actionklassen .....	431
Actionlist .....	429, 430, 480
Actions .....	428, 430f.
ActiveControl .....	373

ActiveSync .....	307
—, Debuggen .....	308
Addon-Installer .....	309
Advantage Datenbank .....	684, 718
Akkumulator-Register .....	259
Aktuelles Fenster schließen .....	385
Align .....	421
Alignment .....	711
Alle-Button, Ergebnis .....	379
Alles ablehnen/bestätigen-Button, Ergebnis .....	379
Ankreuzfeld .....	417, 473
ANSI .....	122
Ansicht v. Fenstern .....	130
Ansi-String .....	34, 150, 258, 302
AnsiToUTF8 .....	82
Anwender-Actions zentral sammeln .....	429
Anwendereingabe, Reaktion auf ..	370
Anwendung anhalten bei Exception .....	371
Anwendung geht in Idle .....	372
Anwendung wiederherstellen .....	372
Anwendung wird minimiert .....	372
Anwendungsfenster duplizieren ..	385
Anwendungs-Icon .....	530
Anwendungsobjekt .....	371
Anwendungstitel .....	146, 371
API-Fkt. i. Pascal suchen .....	301
Application Bundle .....	325
Application.CreateForm .....	146
Application.Initialize .....	363
Application.Run .....	379
Application.Title .....	146
ApplicationType .....	313
Applikationsserver, Webservice einbetten in .....	658
ARM-Prozessor .....	309
Array .....	45
—, dynamisches .....	48
—, offenes .....	49
ASCII-Dateien, Zugriff auf .....	717
Assembler .....	31, 150
assert-Anweisungen .....	149
AsXYZ-Eigenschaften .....	696
AT&T-Syntax, Assembler .....	151
Aufrufkonvention .....	51, 321
— Register .....	64

— wechseln .....	51
Aufrufmechanismen .....	260
Aufrufstack .....	308
Aufzählungstypen .....	35
Aufzählungstypen-Editor .....	283
Ausdrücke auswerten .....	237
Ausführbare Datei .....	140
Ausnahme anzeigen .....	372
Ausnahmebehandlung bei DB .....	704
Ausnahmebehandlung, Unit .....	40
Ausnahme in Hauptschleife beh ..	371
Auswahlfeld .....	417, 474
Auszuführendes Programm, Name ..	633

## B

Barcodes auf Bericht .....	750
Basisklasse d. Bildschirmfenster ..	357
Basisklasse der Klassen .....	348
Basisklasse der Komponenten .....	357
Basisklasse der visuellen Komp. ..	357
Basispakete der IDE .....	280
Baumansicht .....	493
— durchlaufen .....	497
—, Eigenschaften .....	492
—, Eintrag auswählen .....	492
—, Knoten editieren .....	500
—, Knoten hinzufügen .....	494
—, Knotenbeschriftungen .....	494
Baumknoten, automat. expand .....	492
Bedingte Definitionen .....	296
Before-Konstanten .....	707
Benachrichtigungs-Popup .....	511
Bereichstypen .....	35
Bericht .....	750
Bericht anlegen .....	750ff.
Berichtskomponente .....	750
Berichtsvariablen .....	754
Beschriftungen, Text von .....	81
Bestandteile von Lazarus .....	243
Bestätigung-Button, Ergebnis .....	379
Betriebssystem-Definitionen .....	296
Betriebssystem-Exceptions .....	236
Betriebssystemunabhäng. Konst. ..	298
Bevel, Caption .....	482
Bezeichner autom. umbenenn. .....	28, 129
Bezeichner autom. anlegen .....	179
Bezeichner der aktuellen Unit anz. ..	22
Bezeichner, deklariert wo .....	223

Bezeichnerdeklaration ..... 36  
 — suchen ..... 21  
 Bezeichner-Mehrdeutigkeit ..... 32  
 Bezeichnernamen ..... 41  
 Bezeichnerreferenzen suchen ..... 129  
 Bibliothek ..... 19  
 — aus Speicher entfernen ..... 262  
 — in Mac OS X ..... 265  
 — statisch binden ..... 62  
 —, dynamisch/statisch ..... 259  
 —, nicht threadsich.aufrufen ..... 640  
 —, Suchpfad ..... 264  
 Bibliotheks-Versionsnummer ..... 263  
 Bild anzeigen ..... 481  
 Bild auf Canvas zeichnen ..... 597  
 Bild skalieren ..... 597, 621  
 Bild, Puffer anlegen für ..... 597  
 Bildbetrachter ..... 615  
 Bildelement ..... 481  
 Bilder anzeigen ..... 716  
 Bilder laden/speichern ..... 546  
 Bildindex in Bilderliste ..... 430  
 Bildklassen, Basisklasse ..... 621  
 Bildschirmauflösung ..... 373, 403  
 Bildschirmfenster ..... 374  
 — als oberstes setzen ..... 382  
 — autom. v. IDE anlegen lassen ..... 392  
 — im Textmodus öffnen ..... 408  
 — manuell anlegen ..... 392  
 — schließen ..... 379  
 — teilen ..... 418  
 —, b. Programmstart gleichzeitig  
 zwei öffnen ..... 377  
 Bildspeicher ..... 615  
 Binärdatei d. Programms, Ordner ..... 371  
 Binärdatei, Zugriff auf ..... 556  
 Binderdatei, Webservice ..... 658  
 Binding ..... 656  
 Bitbutton ..... 481  
 Bitmap-Format ..... 624  
 Bitmap-Muster ..... 606  
 BLOB-Feld ..... 690  
 Block einrücken ..... 119  
 Blockread/Blockwrite ..... 563  
 Blockvervollständigung ..... 203  
 Bluetooth ..... 306  
 BOF (»Begin Of File«) ..... 692  
 Bookmark ..... 700  
 Boolean ..... 34  
 Borland Database Engine ..... 725  
 Browser f. bedingte Eigenschaft ..... 133  
 Build ..... 147

**C**

C-/C++-Compiler, Aufrufkonv. .... 264  
 C++-artiges Inline ..... 150  
 C++-Schnittstellendatei zu Qt ..... 321  
 Cancel ..... 701  
 Canvas ..... 595  
 —, Drucken auf ..... 591  
 Caption ..... 373, 478

Carbon ..... 241, 325, 348  
 C-artige Operatoren ..... 149  
 C-Bibliothek ..... 32  
 — in Free Pascal ..... 62  
 — in Programm einbinden ..... 33  
 CDecl ..... 48, 62, 260  
 Char34  
 Checkbox ..... 473, 716  
 —, grayed ..... 473  
 —, undefinierter Status ..... 473  
 —, Zustände ..... 474  
 Checkgroup ..... 478  
 Class ..... 44  
 Clientprozeß ..... 647  
 Close ..... 379, 694  
 Cocoa ..... 148, 241, 325, 348  
 —, Datentypen ..... 335  
 —, Framework, Anbindung an ..... 329  
 —, Klassen instantiiieren ..... 332  
 —, Klassen ..... 330  
 —, Sub-Frameworks ..... 325  
 Code Explorer ..... 22  
 Codeblöcke vervollständigen ..... 27  
 Code-Explorer ..... 132  
 Codefaltung ..... 203  
 Codegröße ..... 152  
 Codepage ..... 81, 83, 610  
 Code-Schablonen ..... 26, 212  
 CodeTools ..... 20, 204  
 —, Werte anzeigen ..... 138  
 Combobox ..... 477, 716  
 —, AutoComplete ..... 476  
 —, Einträge in ..... 476  
 —, Pfeiltasten zum Wechseln ..... 476  
 Compiled Source Path ..... 138  
 Compilerausgaben ..... 153  
 Compilerdefinitionen ..... 295  
 Compilerdirektiven ..... 50  
 Compilereinstellungen ..... 147  
 — in Datei speichern/laden von ..... 157  
 — vererben ..... 54  
 —, Menge von ..... 56  
 —, Speicherung ..... 53  
 —, von Packages geerbt ..... 156  
 Compiler-Konfigurationsdatei ..... 136, 154  
 Compiler-Metadaten ..... 351  
 Compiler-Suchpfade ..... 136  
 COM-Port ..... 585  
 ConsoleApplication ..... 371  
 Container für Steuerelem. .... 479  
 Controls, datenbezogene ..... 683, 715  
 ConvertEncoding ..... 83  
 CPU-Definitionen ..... 295  
 CPU-Zeit verteilen ..... 631  
 CreateForm ..... 363  
 Critical Section ..... 644  
 Crosscompiler-Version ..... 311  
 Crosskompilieren ..... 17, 151, 314  
 CSV-Datei ..... 684  
 — anlegen ..... 723  
 —, Feldtrenner ..... 720

C-Typen ..... 34  
 Ctypes ..... 34  
 Currency ..... 33  
 Cursor, momentaner ..... 403  
 Cursorposition im Dataset ..... 692  
 C-Werkzeuge ..... 33

**D**

Daemon ..... 289  
 — registrieren ..... 292  
 — starten ..... 290  
 Daemon-Klassen ..... 291  
 Data Dictionary ..... 735, 737  
 — in IDE einbinden ..... 741  
 — verwalten ..... 735  
 — zur Laufzeit einbinden ..... 740  
 —, neues anlegen ..... 739  
 —, Speicherort ..... 736  
 —, SQL-Anweisung generieren ..... 739  
 Database ..... 713  
 Data-Cursor setzen ..... 692  
 DataField ..... 715  
 Dataset ..... 713  
 — mit DB verbinden ..... 713  
 — prüfen auf Edit-Modus ..... 700  
 —, Browse-Modus ..... 701  
 —, Daten im aktualisieren ..... 694  
 —, Eigenschaften ..... 713  
 —, erstes Ereignis ..... 707  
 —, Komponenten, Container f. .... 691  
 —, leeres ..... 693  
 —, Post-Methode ..... 703  
 DataSource ..... 715  
 DataType ..... 695  
 Datei abschneiden ..... 560  
 Datei auswählen, mehr als eine ..... 515  
 Datei erzeugen, Fehler beim ..... 557  
 Datei kopieren ..... 558, 563  
 Datei öffnen ..... 554, 559  
 Datei öffnen mit TFileStream ..... 564  
 Datei schließen ..... 559  
 Datei sichern/speichern ..... 513, 554  
 Datei z. Lesen/Schreib. öffn. .... 556, 560  
 Datei, blockweise Daten  
 lesen/schreiben ..... 561  
 Datei, lesen aus ..... 562  
 Datei, Moduszuweisung ..... 556  
 Datei, neue erzeug./überschreib. .... 557  
 Datei, sukzessive lesen aus ..... 557  
 Datei, ungeöffnete löschen ..... 560  
 Datei, vorhd. öffnen ..... 557  
 Datei, Werte lesen aus ..... 560  
 Datei-/Verzeichnisdialogkomp.,  
 Eigenschaften ..... 546  
 Dateianfang ..... 557  
 Dateianzeige, Eigenschaften ..... 520  
 Dateiattribut ermitteln/setzen ..... 578  
 Dateiauswahlfeld ..... 519  
 Dateidatum konvertieren ..... 579  
 Dateidatum lesen/schreiben ..... 578  
 Dateidialog, Eventhandler ..... 546



- Dateialloge.....545
  - des Betriebssystems.....547
- Dateien a. Fenster fallenlassen.....372
- Dateien a. Fenster zieh., Ereignis..374
- Dateien im Quelltexteditor,
  - geöffnete.....52
- Dateien in Verzeichnis, Details zu .500
- Dateien laden/speichern.....545
- Dateien ziehen und ablegen.....434
- Dateien, versteckte anzeigen.....515
- Dateien/Verzeichnisse anzeigen...545
- Dateiende, – erreicht?.....557, 559
- Dateiendung, suchen nach.....513
- Dateierweiterungen v. Lazarus.....100
- Dateifilter.....545
- Dateigröße in Byte.....557
- Dateigröße in Records.....558
- Dateihandle zuweisen/freigeben..562
- Dateihandle, ungenutzt/ungültig...298
- Datei-Maske.....141
- Dateimaske f. alle Dateien in einem
  - Verzeichnis.....298
- Dateimaske für Suche.....129
- Dateiname ändern.....559f.
- Dateiname einstellen.....192
- Dateiname konvertieren.....562
- Dateiname, Groß-/Kleinschreib.....298
- Dateiname, langer erlaubt?.....298
- Dateinamenauswahl.....418
- Dateinamenslänge, max.....298
- Datei-öffnen-Dialog schließen.....546
  - , Titelzeile.....545
  - , Autostart-Programmgruppe.....551
  - , Callback.....549
  - , Einträge im Startmenü zeigen..549
  - , Fenstergr.....545
  - , Netzwerkordner anzeigen.....548
  - , Papierkorb.....551
  - , Programmgruppe auswählen...550
  - , Schriften.....551
  - , Startverz.....545
  - , Statusber.....548
  - , Systemst.....551
  - , Unicode-Namen.....547
  - , Verzeichnisse anzeigen.....548
  - , voreingestellte Dateiendung...545
- Datei-öffnen, Menüeintrag.....513
- Dateioperationen.....559
- Dateiposition setzen.....560
- Datei-Suchmaske.....577
- Dateivergleich.....179
- Dateivorschau.....515
- Daten auf Oberfläche anzeigen.....715
- Daten aus Datensatz
  - anzeigen/verändern in GUI...715
- Daten e. Datensatzes auf Konsole
  - ausgeben.....696
- Daten einfügen, neue.....702
- Daten im Speicher bearbeit. 564, 701
- Daten in Objekt erhalten.....571
- Daten in String speichern.....564
- Daten komprimieren.....568
- Daten zw. Prozessen austausch..632
- Datenanzeige in Baumstruktur....418
- Datenbank-Controls.....521
  - , Eigenschaften.....715
- Datenbank-Engine.....685
- Datenbank-Komponenten.....356
- Datenbankname.....730
- Datenbank-Navigationsmethod...716
- Datenbankschnittstelle.....684
- Datenbanktyp.....684
  - ändern.....729
- Datenbank-Zugriff, Exception.....239
- Datenbankzugriffsschichten von
  - Drittanbietern.....684
- Datenbezogene Komponenten.....349
- Datencursor.....690
- Datendateien, Speicherort.....343
- Daten-Desktop.....735
  - , Daten export.....743
  - , Datenbank-Abfr.....741
  - , DB-Verbindung.....741
  - , Exportformate.....743
  - , Funktionen.....736
- Datenmodul.....247, 691
- Datensatz.....686
  - als letzten anfügen.....701
  - ändern.....700
  - anlegen, neuen.....708
  - aus DB löschen.....701, 703
  - geöffnet/geschlossen.....692ff.
  - in DB einfügen.....703
  - wird abgefragt.....692, 693
  - wird editiert.....692, 693
  - , Änderungen verwerfen.....701
  - , Bearbeitung abbrechen.....704
  - , bestehenden suchen.....699
  - , Cursor bewegen auf.....707
  - , Editiermodus.....701
  - , erster/letzter, Cursor auf....694
  - , jeden zweiten löschen.....703
  - , nächster, Cursor auf.....694
  - , navigieren auf nächsten.....692
  - , neuen einfügen.....692, 693
  - , springen auf.....694
  - , vorheriger, Cursor auf.....694
- Datensatz-Änderungen abbrech..706
- Datensätze filtern.....693
- Datensätze im Dataset, Anzahl v. 692
- Datensätze nur i. Speicher halten 693
- Datensätze sortieren.....725
- Datensätze, alle löschen.....703
- Datensatz-Status.....692, 693
- Datentypen.....33
- Datum und Zeit, Unit.....40
- Datum-/Kalenderanzeige.....416
- DB, nativer Zugriff auf.....717
- dBase, Indexdaten.....723
- dBase, Zeichensatz.....726
- dBase-Datei.....723
  - anlegen.....727
  - öffnen.....724f.
  - , Daten speichern.....724
  - , Filter-Ausdruck.....724
  - , Indizes.....724
  - , Quelle f. Master-Detail-Bez..724
  - , UTF-8.....725
  - , Zugriff auf.....717
- dBase-Tabelle/-Datei, Zugriff auf..724
- DB-Benutzer, Paßwort.....730
- DBF-Datei browsen.....725
- DBF-Datei, Zugriff auf.....725
- DBF-Dateien.....684
- DB-Unit.....688
- DB-Verbindungskomp. Proxy f....729
- DB-Zugriff, Klassen für.....688
- DCR-Dateien bearbeiten.....530, 531
- Debuggen über ActiveSync.....308
- Debugger.....33, 152, 232
  - , Unterschiede zu Delphi.....237
- Debuggerausgaben.....238
- Debugger-Einstellungen.....233
- Debuggersymbole entfernen.....153
- Debuggertyp.....233
- Debugging-Prozeß beenden.....239
- Define Templates.....204
- Delphi-Komponente.....529
  - portieren.....60
- Delphi-Modus des Compilers.....533
- Delphi-Packages.....530
  - in Laz-Package konvertieren...55
- Delphi-Projekt konvertieren.....53
- Dereferenzierungsoperator.....46
- Desktop, anderer.....160
- Desktop, Infos über.....402
- Dialog.....390, 395
  - anzeigen.....512
  - m. benutzerdefin. Titelzeile...398
  - mit Icon u. Text.....397
  - mit Warnung.....399
  - , Farben in.....600
  - , Ganzzahlen/Gleitkomma-  
werte bearbeiten.....518
  - , Position a. Bildschirm.....399
  - , Warnungen ausgeben.....400
  - , Widgetset.....526
- Dialog-Icon.....399
- Dialog-Komp., Rückgabewerte...512
- Dialogtypen.....397
- Dienst.....289
  - , Datenbankzugriff.....713
- Dienstanbieter.....655
- Dienstbenutzer.....655
- DirectorySeparator.....298
- DLL.....247, 267
- DMA-Modus.....579
- Dockfähigkeit, Eigenschaften....445
- Docking.....445
- Docking-Operation, Ereignis.....413
- Docking-Ziel.....446
- Dockmanager.....453
- Dockposition speichern.....453

Docksite, mehr. Elemente in ..... 453  
 Dokumentation von Pascal-Code ..... 54  
 DOS-Zeichensatz ..... 726  
 Double ..... 33  
 Drag&Dock ..... 445  
 —, manuelles ..... 451  
 —, Operation ..... 447  
 Drag&Drop ..... 434  
 —, Ausgangspunkt ..... 439  
 —, Eigenschaften u. Ereignisse ..... 435  
 Drag-Objekt, eigenes ..... 438  
 Drag-Quelle in anderem Fenster ..... 438  
 DriveSeparator ..... 298  
 Drop-down-Auswahlliste anbieten ..... 283  
 Drucken ..... 225, 590ff.  
 Druckerauswahl ..... 418, 591  
 Druckjob senden ..... 591  
 Druckparameter einstellen ..... 593  
 Duplicate identifier ..... 36  
 dwarf ..... 31  
 Dynamic Link Libraries ..... 263

**E**

EDatabaseError ..... 704  
 Edit ..... 701  
 Edit-Elemente ..... 517  
 Edit-Methode, Dialog anzeigen ..... 284  
 Editor-Steuerlem. - Hierarchie ..... 469  
 Eigene Zeichenoperationen ..... 370  
 Eigenschaft/Ereignis zur  
 Entwicklungszeit ändern ..... 115  
 Eigenschaft auf Ursprungswert  
 zurücksetzen ..... 284  
 Eigenschaft setzen ..... 43  
 Eigenschaft published definieren ..... 526  
 Eigenschaften e. Komponente  
 veröff./ändern ..... 218, 351  
 Eigenschaften, auf and. Komp.  
 referenzieren ..... 284  
 Eigenschaften, mehrauswahlfäh. ..... 284  
 Eigenschaften, untergeordnete ..... 283  
 — nur lesen ..... 284  
 Eigenschafteneditor ..... 282  
 — für Klassen ..... 285  
 —, Grundfunktionen ..... 286  
 —, Name ..... 282  
 —, TFont ..... 284  
 Eigentümerfenster, Position auf ..... 385  
 Eigenschaften-Mehrfachauswahl,  
 Kriterien ..... 411  
 Ein-/Ausgabe, erweiterte, Unit ..... 40  
 Ein-/Ausgabedatenströme ..... 349  
 Ein-/Ausgabeoperationen,  
 Prioritätsgrad ..... 642  
 Einfache Fenstertypen ..... 395  
 Eingabeelemente ..... 417  
 Eingabefeld ..... 418  
 — für Zahlenwerte m. Schalter ..... 418  
 —, schreibgeschütztes ..... 468  
 —, Text markieren ..... 468  
 —, Textauswahl ..... 470

Eingabefenster nimmt allen Platz  
 im Fenster ein ..... 388  
 Eingabemaske ..... 483  
 Eingabewert, fragen nach ..... 401  
 Einrückung ..... 193  
 Einsprungsroutine ..... 258  
 Einträge zw. Listbx. hin-/herbew. ..... 435  
 Eintrittspunkt, Name d. Bibliotheks- ..... 62  
 Einzelfeld im Datensatz ..... 690  
 Elemente e. graf. Oberfläche ..... 348  
 Ellipse zeichnen ..... 597  
 Endekennung einer Datei ..... 298  
 Entfernen, Lazarus ..... 101  
 EOF (End Of File) ..... 692  
 Ereignis ..... 59, 366  
 Ereignisbehandlungsroutine ..... 367, 370  
 Ereignis-Objekte ..... 632  
 Ereignisse, verknüpfte anzeigen ..... 221  
 Ereignissteuerung ..... 17  
 Erstellmakros ..... 57, 154  
 Erstellmodi ..... 56, 148  
 — speichern ..... 56  
 Eventhandler ..... 429  
 — (Definition) ..... 367  
 — auswählen ..... 368  
 — erzeugen ..... 25  
 — schreiben ..... 369  
 examples-Unterverzeichnis ..... 243  
 Exports ..... 64, 259

**F**

Farbauswahl ..... 418, 490, 598ff.  
 Farbschalter ..... 519  
 FBLib ..... 684, 718  
 FCL ..... 59, 347, 354, 355  
 FCL, nicht-visuelle Komponenten ..... 355  
 Fehlerprüfung ..... 707  
 —, After-Ereignis ..... 708  
 —, Before-Ereignis ..... 708  
 Feld ..... 686  
 — als benötigt kennzeichnen ..... 711  
 — enthält falsche Datentypen ..... 705  
 — in aktuellem Datensatz ..... 692  
 — in Klassendeklar. e. Fensters ..... 353  
 — löschen ..... 702  
 — mit Index 0, Zugriff auf ..... 695  
 — nur lesbar ..... 695  
 —, alternat. Anzeige-/Edit-Text ..... 712  
 —, Alternativwert ..... 712  
 —, Bearbeitungsmaske ..... 711  
 —, berechnetes ..... 695  
 —, boolesches anzeigen ..... 716  
 —, Ereignisse ..... 712  
 —, erschein. als Komponente ..... 712  
 —, Formatstring angeben ..... 711  
 —, nur lesbar ..... 711  
 —, persistent ..... 708, 712  
 —, prüfen auf leeres ..... 699  
 —, Standardwert einfügen ..... 711  
 —, Wert muß stehen in ..... 695  
 —, Zugriff auf ..... 691

Feldanzeige, Breite ..... 711  
 Feld-Anzeigeeigenschaften per  
 Default setzen ..... 709  
 Feldattribute, erweiterte ..... 737  
 Felddaten konvertieren ..... 696  
 Feld-Definition ..... 693  
 Feld-Eigenschaften, Platzhalter ..... 709  
 Feldinhalt, Anzeige steuern ..... 711  
 Feldname ..... 353, 695, 711  
 Feldtext anzeigen ..... 716  
 Feldwert abfragen/setzen ..... 690, 697  
 Feldwert ändern, Ereignis ..... 712  
 Feldwert auswählen ..... 716  
 Feldwert in Datensatz, Zugriff auf ..... 695  
 Feldwert löschen ..... 701  
 Feldwert, Anzeige eines steuern ..... 712  
 Feldwert, BCD ..... 698  
 Feldwert, Bedingung prüfen ..... 711  
 Feldwert, BLOBs ..... 698  
 Feldwert, Integer ..... 698  
 Feldwert, String ..... 698  
 Feldwert, Typen ..... 696  
 Feldzugriff, Eigenschaften ..... 696  
 Fenster ..... 17  
 — ableiten ..... 247, 353  
 — aktualisieren ..... 577  
 — als Child z. Systemsteuer. .... 268  
 — als Symbol in Taskleiste ..... 390  
 — anzeigen ..... 373, 376  
 — auf welchem Monitor? ..... 373  
 — aufzählen ..... 404  
 — aus Speicher entfernen ..... 380  
 — ausblenden ..... 381  
 — bei Programmstart,  
 welches erzeugen? ..... 365  
 — drängelt sich in Liste  
 immer nach vorn ..... 390  
 — dynamisch anlegen ..... 391  
 — erhält Fokus ..... 404  
 — für Programmstart ..... 390  
 — hat sehr kleine Titelleiste  
 (Toolwindow) ..... 389  
 — holen ..... 404  
 — ikonisieren ..... 381  
 — in Desktopmitte anzeigen ..... 385  
 — instantieren ..... 370  
 — größenveränderbar ..... 389  
 — kann kein Menü enthalten ..... 389  
 — max./minimieren ..... 390  
 — mit Memo u. Button ..... 422  
 — nicht autom. v. IDE erzeugt ..... 392  
 — niemals regulär schließen ..... 380  
 — öffnen, Position ..... 385  
 — ohne Rahmen u. Titelleiste ..... 389  
 — Rollbalken anzeigen ..... 373  
 — schließen ..... 370, 379  
 — sichtbar/unsichtbar machen ..... 374  
 — umschalten ..... 382  
 — verstecken ..... 381  
 — wird aktiviert/deaktiv ..... 374  
 — wird geschlossen, Ereignis ..... 374

— zerstören..... 370, 381  
 —, autom./nicht automat. einger...365  
 —, Dateien fallenlassen auf.....373  
 —, Erzeugen des.....360  
 —, Fokus setzen auf.....404  
 —, Mehrfachinstanzen.....392  
 —, Position u. Ursprungsgröße...385  
 —, Rahmenbreite/-höhe.....388  
 —, Ressourcendatei.....375  
 —, Unit.....374  
 Fensterart.....390  
 Fenster-Basisklasse.....527  
 Fensterbeschriftung abfragen.....401  
 Fenstereigenschaften,  
   gespeichert in.....17  
 Fenstereigentümer setzen.....386  
 Fenstergröße.....383  
 Fenstericons.....388  
 Fensterinstanz erzeugen, Ereignis 374  
 Fensterinstanz wird zerstört.....374  
 Fensterinstanz, neue anlegen.....386  
 Fensterinstanzen freigeben.....375  
 Fensterliste.....390, 393, 404  
 Fensterliste des Taskbars.....386  
 Fenster-Menü.....373, 404  
 Fenstermenü-Eintrag anlegen.....404  
 Fenstername.....375  
 Fensterposition.....373, 383  
 Fensterrahmen.....386  
 Fenstersystem fährt Anwendung  
   herunter.....372  
 Fenstertitel.....373, 386, 391, 399  
 Fenstertyp.....373  
 Fensterzugriff, Formvariable für...375  
 Fensterzustand speichern.....565  
 Fensterzustand, anfänglicher.....373  
 Filter.....693  
 Filterauswahlbox.....521  
 FindFirst/FindNext/FindClose.....573  
 Firebird.....684  
 Firebird/Interbase.....729  
 Fläche füllen.....597  
 Flächen zeichnen.....604  
 Flat Files.....685, 723  
 Fontauswahl.....373, 607ff.  
 Form.....247  
 Forminstanz speichern/zerstören.394  
 Formular anlegen.....116  
 Formular, Layoutinformationen.....139  
 Formular-Designer, Mausclick.....286  
 Formulareditor.....188  
 Fortschrittsanzeige.....416, 492  
 FPC installieren.....99  
 FPC, mehrere V. parallel install.....319  
 fpc.cfg.....136, 154  
 FPDoc-Editor.....19, 79, 133  
 fpGUI.....241, 248, 348  
 Frame.....17, 247, 479  
 Frames schachteln.....479  
 Free Pascal Compiler.....15  
 Free Pascal kompilieren.....106

Free Vision.....253  
 FreeLibrary.....262  
 Free-Pascal-Anwend. beginnen...247  
 Free-Pascal-Bibliotheken in C.....64  
 Fremdsprachige Zeichen.....81  
 FTP-Server, Pfadtrenner.....297  
 Füllmuster, eigene.....605  
 Funktion/Proz. exportieren.....64, 259  
 Funktionsergebnis, Rückgabe.....259

## G

gdb.....33, 232  
 Geometrische Einfassungen.....416  
 Geometrische Figur anzeigen.....481  
 Gerätenamen in /dev.....317  
 Gitter anzeigen.....418  
 Gitter mit Datenbankanbindung...483  
 Gitterelemente.....483  
 Gitterstruktur.....716  
 Gleitkommatypen.....33  
 Globales Anwendungsobjekt aller  
   GUI-Applikationen.....357  
 Glyph.....186, 462, 481  
   — auf Schaltflächen.....371  
   — in Menü.....371  
   —, Bitmaps in.....466  
 GNU Assembler.....33  
 GNU Binutils.....311  
 GNU Debugger.....31, 232  
 GNU Linker.....33  
 GNU make.....108  
 GNU-Übersetzungsprojekt.....77  
 GOTO erlauben.....150  
 gprof.....33  
 Grafikformate.....546, 620  
 Grafikkomponenten.....615  
 Grid.....716  
   —, Eigenschaften.....486  
   —, fixierte Zellen.....487  
   —, Spalte automatisch anpassen485  
   —, Spalten.....487  
   —, Zeile in Eingabemodus.....487  
   —, Zeilenfarbe.....487  
 Groß-/Kleinschreibung.....36, 208  
 Groupbox.....478  
 Gruppe v. Steuerelementen anz...416  
 Gtk.....59, 241, 248, 293, 320, 348  
 Gtk 1, Theme.....320  
 Gtk 2.....197, 241  
 GUI-Anwendung beginnen.....247  
 GUI-Bibliotheken.....248  
 GUI-Element, Funktionalitäten.....411  
 GUI-Programme, Aufbau.....358  
 GUI-Toolkits.....320  
 GUI-Widget-Sammlung.....347

## H

h2pas.....33, 64  
 h2paswizard.....64  
 Haltepunkte.....237  
 Handler in Idle-Schleife.....430

Hardlinks.....319  
 Hardwareschnittstelle.....579ff.  
 Haupt-/Kontextmenü.....459  
 Hauptfenster blockieren.....378  
 Hauptfenster d. Anwendung.....249  
 Hauptfenster erzeugen.....363  
 Hauptfenster explizit anzeigen.....371  
 Hauptfenster schließen.....375  
 Hauptfenster, zentriert auf.....385  
 Hauptmenü anlegen.....462  
 Hauptmenü, neuer Eintrag.....404  
 Hauptprogramm, Verwaltung d.....360  
 Hauptprogrammquelltext.....250  
 Hauptprogrammumschleife.....249  
 Hauptprojektdatei.....249  
 Hauptprojektsymbol.....250  
 Hauptquelldatei, Name.....19  
 Hauptquelltextdatei.....139  
 Hauptroutine.....258  
 Hauptthread.....640  
 Headerdateien.....39  
 HFS-Dateisystem.....327  
 Hide.....381  
 Hilfesystem aufrufen.....372  
 Hilfetooltip aufpoppen.....371  
 Hilfslinien.....190  
 Hilfsprogramme.....100  
 Hint anzeigen.....371  
 Hint in Statusbar anzeigen.....506  
 Hintergrundfarbe e. Controls.....458  
 Hinweis in Popup-Fenster.....458  
 Hinweis anzeigen.....416, 478  
 Hinweisertext anzeigen.....372  
 Histogramme anzeigen.....416  
 Hostname d. DB.....730  
 HTTP-Applikationsserver.....663  
 HTTP-Servermodul.....655  
 HTTP-Versand.....656

## I

I/O-Überprüfung ein-/ausschalt...557  
 i18n.....77, 173  
 Icon.....373  
   — in Titelzeile e. Fensters.....388  
   — laden.....627  
   —, Bestandteile.....626  
   —, Ressourcen-ID.....272  
   —, Systemschrift.....403  
 Icon-Definitionsdatei.....267  
 Iconformat.....626  
 Icons f. Speedbuttons.....481  
 ICS.....656  
 IDE erweitern.....246  
 IDE fährt nicht hoch.....244  
 IDE findet Package nicht.....280  
 IDE für and. Widgetset kompil.....239  
 IDE in and. Verzeichnis kompil.....244  
 IDE kompilieren, Einstellungen.....243  
 IDE neu kompilieren.....239, 277  
 IDE, Hauptfenster.....115  
 IDE, Makros.....57

IDE, Sicherungsdateien .....	191
IDE, Tastaturbelegung .....	198
IDE-Fenster, Größe .....	187
IDE-Schnittstelle .....	243
IDE-Suchpfade .....	136
Idle-Aktionen sind ausgeführt .....	372
Implementationsdatei, Webserv. ..	658
Implementierungs-Abschnitt .....	39
Include-Datei für Linux/Win32 .....	295
Include-Dateien .....	28, 50, 294
Include-Suchpfad .....	295
Index .....	686
—, Feldname des .....	724
—, Name des .....	724
Indexdefinitionen e. Tabelle .....	688
Indy (Internet Direct) .....	656
Indy-basierter HTTP- TCP-Serv. ....	655
inherited Pfade .....	169
In-memory-Daten .....	719
InnoSetup .....	346
inout32.dll .....	581
In-Process-Transportprotokoll .....	656
Input .....	561
Input-/Output-Dateideskriptoren ..	635
InputQuery .....	401
Installationsprogramm .....	346
Integer, 64-Bit-System .....	339
Integer, Größe .....	269
Intel-Syntax, Assembler .....	151
Interbase .....	684
Interface-Definition .....	45
Interface-Abschnitt .....	39
Interface-Datei, Webservice .....	658
Interface-Units .....	536
Interrupt .....	260
Introspection .....	351
IP-Adresse .....	647

**J**

JPEG .....	625
------------	-----

**K**

Kalender .....	716
Kapselung d. vis. Schicht v. GUI ..	347
KDE .....	320
—, Window Man., Fenstericon ..	388
Kernelfunktionen .....	31
Kernel-Gerätetreiber .....	581
Kernelobjekte .....	632
Kindkomponenten .....	527
Klasse .....	35
— ableiten .....	44
— in Objektp. bearbeiten .....	349
Klassenbibliotheken .....	347
Klassendeklaration, Abschnitte ..	42
Klasseninstanzen, Namen .....	352
Klassenmethode .....	43
— deklarieren .....	44
Klassensystem, plattformunabh. ..	525
Klassenvervollständigung .....	228
Kommandozeile verarbeiten .....	372

Kommandozeilen-Optionen, Groß-/Kleinschreibung .....	371
Kommandozeilen-Optionen, indexierter Zugriff auf .....	371
Kommandozeilenschalter, Zeichen f. ....	371
Kommentar .....	120
Kommunikationsprotok., Listener f.	664
Kompilieren, Tastenkombination ..	157
Kompilierte Unit .....	252
Kompilierung, bedingte .....	17, 51, 294
Komponente .....	59
— in IDE installieren .....	65, 281
— lesen/speichern .....	526
— ohne Parent .....	527
—, abgeleitete anlegen .....	247
—, eigenes Handle .....	527
—, Eigenschaft Name .....	526
—, einfache .....	527
—, installierte suchen .....	133
—, neue erzeugen .....	67
—, plattformunabh./-abhäng. ....	528
—, ..	526
Komponentenauswahl, mehrere ..	455
Komponentenbeispiele .....	457
Komponenten-Editoren .....	286
—, Grundklasse .....	286
Komponenteneigensch. über Fenster- sessions hinweg speichern ..	373
Komponentenname .....	352
Komponentenpal. Icon hinzufüg. ..	543
Komponentenpal., Kontextmenü ..	456
Komprimierungs-Stream .....	567
Konfigurationsdatei .....	339
—, global/lokal .....	339
—, Name ermitteln .....	339
—, Speicherorte .....	340
Konfigurationsdaten schreiben ..	545
Konsole .....	561
— automatisch öffnen .....	255
Konsolenanwendung .....	247
Konsolenfenster-Titel .....	633
Konsolen-Pipelining .....	254
Konsolenprogramm .....	253, 371
Konstanten-Array .....	49
Konstanten-Deklarationsblock ..	39
Konstruktornamen .....	150
Kontextmenü .....	460
Kreis/Ellipse zeichnen .....	613
Kreisbogen zeichnen .....	597

**L**

Label erlauben .....	150
Label, &-Zeichen .....	467
Label, Eigenschaften .....	468
Laufwerksbuchstabe .....	61
Laufzeitbibliothek .....	347
Layout, Werte .....	426
Layout-Objekte .....	419
Lazarus Daten-Desktop .....	689
Lazarus erstellen, StandardEinst. ..	197

Lazarus kompilieren .....	106, 108, 243
Lazarus-Anwendung in Emulator ..	308
Lazarus-Anwendung, Ebenen .....	535
Lazarus-Formular .....	218, 350
Lazarus-Package, Definition .....	54
Lazarus-Projekt .....	359
Lazarus-Verzeichnis, aktuelles ..	54
lazbuild .....	19, 76
lazdaemon .....	289
lazDBExport .....	744
LazReport .....	750
LCL .....	15, 248, 347
— einrichten .....	363
— neu kompilieren .....	243
—, Basisklasse .....	526
—, Basisklassen .....	357
—, Klassensysteme .....	525
—, portieren auf .....	535
LCL-Interface .....	59, 535
— zu Widgetset .....	525
—, abstrakte Basiskl. ....	528
LCL-Klassen, public/published Eigenschaften .....	528
LCL-Komponenten .....	455
LCL-Modus .....	313ff.
LCL-Unterstützung f. graf. Oberfl. ..	293
LCL-Widgetset .....	148
Lebensdauer v. Objekt verwalten ..	678
LED-API .....	306
Lesezeichen .....	699
Linie zeichnen .....	597, 610
Linie, Hintergrund .....	601
Linien verbinden, -endpunkte ..	602
Linken .....	31, 152
—, dynamisch .....	263
—, ext. Objektdatei/Biblioth. ....	259
Linkercode .....	66
Links auflösen .....	319
Linux, installieren auf .....	101
Linux, Verzeichnishierarchie .....	318
Linux-Versionen .....	101
Listbox .....	716
— umsortieren .....	435
—, Einträge einfügen .....	444
—, Einträge sortieren .....	475
—, Höhe .....	475
—, Mehrfachauswahl .....	475
Liste, Auswahl aus .....	474
Listenfeld .....	474, 500
Listen-Klasse .....	350
Listenstrukturen anzeigen .....	418
Listview, Einträge .....	502f.
Listview, Spaltendefinitionen ..	502
Listview, Verzeichnisscan .....	504
Login-Namen ermitteln .....	296
LongInt, Größe .....	269
Losgelöstes Fenster schließen ..	445
Low-Level-Routinen, Zugriff auf ..	296

**M**

MacOS X, Dateinamenvergleich ..	327
---------------------------------	-----



MacOS X, Dateisystem ..... 327  
 MacOS X, Framework linken ..... 325  
 MacOS X, installieren auf ..... 113  
 MacOS X, Unicode ..... 327  
 MacOS X, Verzeichnisse ..... 328  
 MacOS X-API ..... 324  
 MacOS, Framework in-/deinstall. 265  
 make.exe ..... 90  
 Makedatei, Makefile ..... 18, 90, 156  
 Makefile.fpc ..... 156  
 make-Werkzeug, Pfad ..... 184  
 Makros ..... 51  
 — der IDE ..... 57  
 — in Suchpfaden ..... 71  
 —, eigene definieren ..... 56  
 Makro-Skriptsprache, Fähigkeiten 58  
 Master-Detail-Beziehungen in  
   SQLDB ..... 735  
 Mathematische Routinen, Unit ..... 40  
 Mausbewegung, reagieren auf ..... 458  
 Mausclick, Ereignis ..... 413  
 Maus-Rechts-Doppelclick im  
   Formular-Designer ..... 286  
 Maustaste gedrückt/gelöst ..... 459  
 Maustaste, Ereignis ..... 413  
 Maustaste, linke, auf El. klicken ..... 459  
 Mauszeiger ist über Element ..... 459  
 Mauszeiger passiert Elementrand 459  
 Mauszeiger-Aussehen ..... 458  
 MDI-Fenster ..... 390  
 Mehrere Dateien verbinden ..... 50  
 Mehrfachvererbung ..... 45, 323  
 Meldungsdialoge anlegen ..... 231  
 Memofeld, Zeile/Absatz ..... 470  
 Menü ..... 373, 459, 464  
 — anpassen ..... 287  
 —, dynamisches ..... 464  
 —, Systemschrift ..... 403  
 —, Tastenkürzel ..... 461  
 Menübeschriftungen umbenennen ..... 463  
 Menüeditor ..... 462  
 Menüeintrag ..... 460  
 — als Radiobutton ..... 461  
 — eingeschaltet? ..... 430  
 — mit Betätigt-Zeichen ..... 461  
 —, Bild vor ..... 462  
 Menüfunktion schnell anwählen ..... 481  
 Menüfunktion, Zugriff auf ..... 507  
 Menüs, Widgetset ..... 526  
 Menüstruktur mit Menüeinträgen 416  
 Message Composer ..... 231  
 Message Loop ..... 359  
 MessageDlg ..... 397  
 Message-Methoden, Portierung ..539  
 Meta-Datenbank ..... 737  
 Methode ..... 43  
 — autom. extrahieren ..... 28  
 — exportieren ..... 261  
 — static deklarieren ..... 150  
 — überschreiben ..... 43  
 —, leere entfernen ..... 29

Methodenaufrufe ..... 259  
 MIME-Datenbank vervollständigen.. 113  
 Minimale IDE bauen ..... 244  
 Mobile Plattformen, Unterschiede ... 305  
 Modale Fenster ..... 376  
 —, Ergebniskonstanten ..... 379  
 Modales Ergebnis ..... 398  
 Modified ..... 692  
 Moduswechsel ..... 56  
 Monitore, Anzahl ..... 403  
 Multitasking, präemptives ..... 631  
 Multithreaded Anwendungen ..... 639  
 Mutex-Objekte ..... 632  
 MySQL ..... 684, 729

## N

Nachlade-Optimierung ..... 152  
 Nachricht anzeigen ..... 396  
 Nachricht über Anwender-Aktion 372  
 Nachrichten an LCL-Interf. senden 59  
 Nachrichten, Analyse d. .... 366  
 Nachrichtenbehandlungs-Methode . 536  
 Nachrichtenfenster ..... 115, 131  
 Nachrichtenfenster, Fokus auf .... 186  
 Nachrichtenschleife ausführen.... 365  
 Nachrichtenverarbeitung ..... 358  
 Nachschau-Tabelle ..... 686  
 Nachschlagelisten ..... 719  
 Namensvergabe ..... 41  
 Namespaces ..... 32  
 Nativer Zugriff auf DB ..... 717  
 Navigieren in Datensätzen,  
   Methoden für ..... 690  
 Navigieren in DB ..... 692  
 Navigieren in DB, Methoden für... 693  
 ncursor ..... 253  
 Neues Fenster i. Projekt hinzufüg. 248  
 Next ..... 694  
 NexusDB ..... 718  
 Nicht größenveränderb. Fenster .. 390  
 Nicht-ASCII-Zeichen ..... 81  
 Nichtmodales Fenster anzeigen... 376  
 Nicht-visuelle Klassen, Aufteilung ... 348  
 Nicht-visuelle Komponenten ..... 59  
 Nonvisuelle Komponenten ..... 347  
 Normales Fenster ..... 390  
 Nullsoft ..... 346  
 Nullterminierter String ..... 302

## O

Oberflächenbibliothek, gekapselt. 348  
 Objective-C-API, MacOS X ..... 325  
 Objective-C-Quelle in Pascal  
   konvertieren ..... 333  
 Object-Pascal-Typ ..... 33  
 Object-Pascal-Übersetzung e.  
   Bibliotheks-Interfaces ..... 33  
 Objekt Pascal, Syntaxausnahmen 149  
 Objekt über Steuerelement ziehen,  
   Ereignis ..... 413  
 Objekte ..... 42

— auf Vorrat anlegen ..... 678  
 — ziehen u. ablegen ..... 435  
 —, Cut&Paste ..... 408  
 Objectfactory für Implementation  
   von Webservice ..... 678  
 Objektspektor ..... 59, 115, 190, 218  
 —, Einstellungen ..... 191  
 Objektpooling ..... 678  
 ODBC ..... 688, 729  
 ODBC-Treiber ..... 684, 717  
 Öffnen-/Schließen-Dialog,  
   Eigenschaften ..... 516  
 OK-Button ..... 465  
 — anzeigen ..... 396  
 —, Verarbeitung ..... 378  
 OldFpcCall ..... 260  
 OLE-Objekte ..... 529  
 OnActivate ..... 374  
 OnChange ..... 712  
 OnClick ..... 413, 459  
 OnClose ..... 374, 380  
 OnCloseQuery ..... 370, 374  
 OnContextPopup ..... 413  
 OnCreate ..... 370, 374  
 OnDbClick ..... 413, 459  
 OnDeactivate ..... 374  
 OnDestroy ..... 370, 374  
 OnDragDrop ..... 413  
 OnDragOver ..... 413  
 OnDropFiles ..... 372, 374  
 OnEditingDone ..... 413  
 OnEndDock ..... 413  
 OnEndDrag ..... 413  
 OnEndSession ..... 372  
 OnException ..... 372  
 OnGetText ..... 712  
 OnHelp ..... 372  
 OnHide ..... 374  
 OnHint ..... 372  
 OnIdle ..... 372  
 OnIdleEnd ..... 372  
 OnMinimize ..... 372  
 OnMouse ..... 413, 459  
 OnMouseWheel ..... 413  
 OnNewRecord ..... 708  
 OnPaint-Event ..... 370  
 OnPostError ..... 708  
 OnQuadClick ..... 413  
 OnQueryEndSession ..... 372  
 OnRestore ..... 372  
 OnSetText ..... 712  
 OnShortCut ..... 374  
 OnShow ..... 374, 376  
 OnShowHint ..... 372  
 OnStartDock/OnStartDrag ..... 413  
 OnTripleClick ..... 413  
 OnUserInput ..... 372  
 OnWindowStateChange ..... 374  
 OpenGL ..... 325  
 Optionenschnittfläche ..... 417  
 Oracle ..... 684, 729

OS/2 .....	293
Output .....	561
Override .....	43
Owner-owned Hierarchy .....	408

**P**

Package .....	19, 60, 140, 161, 276
— der IDE hinzufügen .....	277
— dynamisch laden .....	541
— Editor .....	65
— generieren .....	247
— Graph .....	61
— in Projekt aufnehmen .....	278
— installieren .....	64, 162
— kompilieren .....	68, 76, 163, 243
— suchen .....	72
— übersetzen .....	76
—, Abhängigkeit .....	75, 167
—, aufnehmen in .....	162
—, Bestandteile .....	278
—, Compilerinstellungen vererben .....	74
—, Dateien hinzufügen .....	60
—, dynamisches .....	276
—, externe i. Komponentenpal. ....	459
—, in Projekttinspektor als	
Abhängigkeit hinzufügen .....	278
—, Lazarus- .....	102
—, Liste d. geöffneten .....	280
—, Sprachen .....	173
—, Suchpfad .....	74, 279
—, welche in IDE? .....	74
Package-Editor .....	65ff, 161ff., 277ff.
Package-Einstellungen/-Suchpfade .....	54
Package-Konzept .....	17
Package-Name .....	72
packagename.compiled .....	76
Package-Registrierung .....	243
Package-Status .....	174
Package-Typ .....	172
Package-Unitpfade d. zum Quelltext-	
Pfad e. Projekts hinzufügen .....	138
Package-Version .....	171
Paket, Dateien eines .....	66
Paket bauen, neues .....	247
Paketmanager v. Linux .....	102
Palettenseite .....	166
Panel .....	479
—, andere Controls ablegen .....	479
—, Rahmen .....	479
Panel-Editor .....	506
Paradox-Dateien, Zugriff auf .....	717
Parallele Kommunikation,	
Linux/Unix .....	582
Parallele Protokolle .....	579
Parallelinstallation, FPC .....	319
Parallel-Port .....	580
Parameterhinweise .....	21
Parameterliste erzeugen .....	124
Pascal .....	260
—, Bezeichner .....	407
PasCocoa .....	329

Passive Steuerelemente .....	416
Paßwort-Abfrage .....	402
PChar .....	34
PDA .....	315
Peephole-Optimierungen .....	151
Performanceanalyse .....	153
Persistente Objekte .....	571
Pfade, Makros für .....	148
Pfadtrenner .....	297
Pfeil zeichnen .....	519
Pinbelegung d. pllen. Steckers .....	582
Pinsel .....	604
—, am Canvas ausgerichteter,	
gebitmappter .....	606
—, benutzerdefinierter .....	605
Pinselmuster .....	604, 606
Pixel setzen/lesen .....	597, 610
Pixelbildformate, Basisklasse .....	620
Plattformabhängige Quelltextteile	
auslagern .....	294
Plattformunabhängigkeit .....	133
Platzhalter umwandeln .....	126
Plugin .....	276
PocketPC .....	305
POEdit .....	77
Pointer .....	34
Pointer-Arrays .....	48
Polygon, Koordinaten übergeben .....	611
Polbare Objekte .....	678
Popup-Fenster mit Hint .....	458
Popup-Menü anzeigen, Ereignis .....	413
Popup-Menü öffnen .....	464
Popup-Menüeinträge editieren .....	287
Port, Zugriffsrecht auf .....	582
POSIX-Thread-Bibliothek .....	640
Post-Anweisung/-Methode .....	701
PostgreSQL .....	684, 729
Primärbildschirm .....	384
Printer4lazarus .....	117, 225, 590, 751
Printers4lazide .....	590
Print-Event beginnen/beenden .....	591
Private-Abschnitt .....	42
Profiling .....	33
Program .....	259
Programm, Optionen übergeben .....	633
Programme ohne Anwender-	
Interaktion .....	289
Programm-Formulare,	
Einstellungen .....	188
Programmstart beeinflussen .....	633
Programm-Umgebung .....	402
Projekt .....	19, 52, 138
— aus IDE starten .....	187
— beginnen .....	247
— veröffentlichen .....	141
—, Datei hinzufügen .....	247
—, neues Form hinzufügen .....	146
Projektarten .....	246
— über externes Package .....	247
Projekteinstellungen .....	147
Projekt-Hauptdatei/-unit .....	140, 146

Projekt-Kopie .....	141
Projekt-Metainformationen .....	19
Projektspezifische Einstellungen .....	138
Property-Editor .....	282
Protected-Abschnitt .....	42
Prozedur, definiert wo .....	223
Prozeduraufrufvervollständigung .....	231
Prozeduren, geschachtelte .....	209
Prozedurrumpf .....	221
Prozeß anhalten .....	634f.
Prozeß läuft? .....	635
Prozeß starten .....	632
Prozeß, synchrone Aufrufe .....	638
Prozeß-Exitcode .....	298
Prozeß-Objekte .....	632
Prozeßwechsel .....	631
Prüfsumme .....	18
Public/Published Eigenschaften .....	528
Public-Abschnitt .....	42
Published Eigenschaften in	
Datenstrom speichern .....	350
Published-Abschnitt .....	42
Punkt .....	595

**Q**

Qt .....	59, 241, 248, 293, 320, 323, 348
Qt exportieren .....	321
Quelldatei in UTF-8 umwandeln .....	81
Quellen, L. installieren aus .....	101
Quelltext formatieren .....	119
Quelltext markieren .....	119
Quelltext sortieren .....	121
Quelltext vervollständigen .....	228
Quelltext, navigieren in .....	132
Quelltext, springen in .....	129
Quelltexte von Lazarus .....	87
Quelltexteditor .....	19, 55, 115, 223
—, Einstellungen .....	192
Quelltextschablonen, Liste der .....	26
Quelltextteile in Proz. auslagern .....	123
Quelltext-Unit .....	139
Quelltextvervollständigung .....	23
Quellverzeichnis aussuchen .....	574
Queries .....	686
—, Parameter .....	688
Query-Editor .....	746

**R**

Race Condition .....	642, 644
RAD-Fähigkeiten .....	15
Radiogroup .....	478, 716
Rahmeneigenschaften .....	386, 414
rc-Ressourcendatei, Aufbau .....	571
RDBMS, Verwaltungsroutinen .....	687
ReadLn .....	560
ReadOnly .....	695, 711
Recht-/Vieleck zeichnen .....	597
Rechteck, Zeichenroutinen .....	612
Recno .....	692
Record .....	35, 686
Recordcount .....	692



Record-Definition.....	36
Refactoring-Tools.....	28
Refresh.....	694
Register.....	260
RegisterComponents.....	281
Registersicherung.....	260
Registrierdatenbank, lesen aus.....	345
Reguläre Ausdrücke.....	168
— für Suche.....	125
—, Symbole.....	126
Reiter-/Tab-Control.....	510
RemObjects Software.....	718
Remote Debugging.....	234
Rename.....	560
Report.....	750
Report-Designer.....	755
—, Berichtgesteuert.....	750
Required.....	695, 711
Ressourcenstring.....	34
Ressourcen lesen.....	570
Ressourcen, interne/externe.....	571
Ressourcencompiler.....	532, 570
Ressourcendateien.....	139, 250, 342
—, MacOS X.....	343
Ressourcen-Stream.....	571
Ressourcenstrings.....	54, 178
— definieren.....	76
— übersetzen.....	78
—, Übersetzung.....	146
Ressourcen-Verzeichnis ermitteln.....	344
RGB-Wert zuweisen.....	610
Rollbalken.....	477
—, Einstellungen.....	373
RPC.....	654
RTL.....	347
RTL, Basisklassen.....	349
Rückgabewerte v. Schaltflächen.....	398
Run.....	365
Run Time Library.....	31
rundll.exe.....	267
Run-Time Type Information/RTTI.....	351

**S**

SafeCall.....	260
Sammlungen ident. strukt. Einträge.....	350
Schalter erhält Fokus.....	467
Schalter mit Beschriftung.....	416, 465
Schalter, fensterfähiger.....	467
Schalter, Standardaktion zuweis.....	466
Schaltfläche.....	465
— anzeigen.....	398
— für Menü für Fensteroperat.....	388
— für Fenster maximieren.....	388
— für Fenster schließen.....	388
— für Hilfe aufrufen.....	388
— mit Pfeilen.....	510
—, flach.....	508
—, Rahmenlinien anz.....	508
Schaltflächen OK, Abbruch, Schließen, Hilfe.....	521
Schaltflächen-Werte.....	398
Scheduler.....	631
Schiebereglern.....	491
Schnappschuß v. Lazarus.....	87
Schnittstelle, parallele.....	580
Schrift auf Kind-Elementen.....	373
Schrift eines Textelements.....	458
Schriftauswahl.....	418
Schriften installierte, Namen.....	403
Screenshot des Monitors.....	621
Scrollbar.....	477
Scrollbox.....	483, 616
SELECT.....	732
Self-Parameter.....	259, 321
Self-Variable.....	150
Semaphoren-Objekte.....	632
Serialisierungsformate.....	668
Serielle Schnittstelle.....	585
Serieller Adapter, Belegung.....	586
Serielles Protokoll.....	579
Serverprozess.....	647
Serviceanbieter implementieren.....	657
Serviceerweiterungen, Registrier- datenbank f.....	682
Service-Interface.....	658
Session.....	52, 56
Shared Object.....	247, 263
ShellExecute.....	639
Shellobjekte initialisieren.....	552
Shell-Verzeichnisdialog.....	547
ShortBool.....	34
ShortString.....	34, 150
Single.....	33
Sitzungsinformation speich.....	53, 339
Smartlinks.....	31, 151
Smartphone.....	315
Snapshot.....	86
SOAP-Header.....	655, 666
SOAP-Serialisierungsform. regist.....	664
Spaltentitel anzeigen.....	510
Speedbutton.....	481
Speicher allokieren.....	46
Speicher als lokales Array.....	302
Speicher freigeben.....	46
Speicher, Daten im.....	717, 719
Speicherbereich kopieren.....	47
Speicherbereich vergrößern.....	46
Speicherlecks.....	153
Speicherort e. Konfigurationsdatei angeben.....	373
Speicherplatz allokieren/freigeb.....	302
Spezialfenster.....	395
Splitter.....	427
Sprachen, mehrere.....	146
SQL.....	687
—, Abfragen schreiben.....	745
—, Daten aktualisieren.....	733
—, Delete-/Update-Anweisung.....	734
SQL-Datenbank.....	728
—, Verbindung mit.....	728, 730
SQL-DB.....	728
—, DB verbinden mit.....	730
—, Parameter.....	735
—, Datensatz aktualisieren.....	734
—, Datensatz einfügen.....	734
—, Master-Detail-Beziehungen.....	735
—, Schlüsselfelder.....	734
—, SQL-Anfragen ausführen.....	728
—, Transaktionen steuern.....	728
—, Zugriff a. Datensätze.....	733
SQL-DB-Verbindung, Controls für.....	729
SQLite3.....	684, 729
Stabile Version.....	86
Stackbereinigung.....	259
Standardausgabe.....	254
Standarddatenbankkomponente.....	684
Standard-Diagnostics-Output- Dateihandle, Wert.....	298
Standard-Ein-/Ausgaberroutinen.....	253
—, Unit.....	40
Standardeingabe.....	254
Standard-Hilfedatei d. Anwend.....	71
Standard-In-/Output-Gerät.....	561
Standard-Inp.-Dateihandle, Wert.....	298
Standard-Output-Dateih., Wert.....	298
Standard-Units von Free Pascal.....	40
State.....	692, 693
Statusaktualisierung v. Control prüfen.....	429
Statusanzeigen.....	416
Statusbar.....	505
—, Position verschieben.....	505
StdCall.....	62, 260
StdCall-Modifizierer, auf versch. Systemen.....	62
StdErr.....	561
StdIn-/StdOut-/StdErr-Bezeichner bei Programmstart schließen.....	289
StdInputHandle/StdOutputHandle.....	298
Steuerelement.....	405
— auf Fenster, Zugriff auf.....	407
— abreißen.....	445
— a. Rand d. Elternobj. kleben.....	422
— ein-/ausgeschaltet?.....	430
— erhält Fokus.....	373, 404, 414
— hinzufügen.....	405
— im Quelltext.....	250
— in Zeilen/Spalten organisier.....	426
— löschen.....	411
— mit Bild.....	416
— optisch gruppieren.....	482
— von Windows XP, Stil.....	250
— verschieben.....	408
— wiederanfügen.....	446
— woanders ablegen.....	445
—, Abstamm. d. visuell.....	457, 458
—, aktives ändern.....	376
—, Ausrichtung.....	419, 422
—, Drag&Drop/Dock.....	412
—, Eigenschaften, mehrere gleichzeitig setzen.....	410
—, Eigentümer.....	408
—, Größe relativ ändern.....	419, 427

—, Größenänderung, Aktionen...419ff.  
 —, mehrere Instanzen erzeug. ...410  
 —, Minimal-/Maximalgr. ... 412, 419  
 —, Schriftart .....412  
 —, springen auf mit Tab-Taste ..414  
 —, Statusaktualisierung prüf.....429  
 Steuerelemente, Ausrichtung ver-  
 schiedener.....422  
 —, gitterartig anordnen.....426  
 —, veröffentlichte Eigenschaft...419  
 —, vis., gem. Eigenschaften .....458f  
 Steuerelement-Farbe .....412  
 Steuerelement-Höhe/-breite .....412  
 Steuerelement-Name.....406  
 Steuerelement-Position.....419  
 Steuerelement-Randabstand.....419  
 Stored Procedures .....687  
 Stream .....563  
 — in Speicher .....566  
 — in Zeichenkette umwandeln ..567  
 —, Daten lesen aus.....565  
 —, Daten schreiben in .....565  
 —, Eigenschaften.....564  
 —, Position setzen .....565  
 Streaming-System, Unit .....40  
 String holen.....283  
 String konvertieren.....83  
 Stringgrid .....485  
 Stringkodierung ermitteln .....83  
 Stringkonstanten-Definitionen .....39  
 Stringliste aufbauen .....575  
 Stringliste freigeben.....394  
 Strings nach UTF-8 wandeln .....82  
 Strings übersetzen .....78  
 Stringtypen.....34  
 Strom über Schnittstelle .....581  
 Subclassing .....333  
 Subversion.....88  
 Suchen-&Ersetzen-Schablonen ...128  
 Suchfunktion .....577  
 Suchpfad f. Quelltextdateien .....148  
 Suchpfad für Bibliotheken.....148  
 Suchpfade, kontextbezogene .....55  
 Suchtext .....127  
 SVN-Client für Windows.....87  
 SVN-Kommandozeilenclient.....88  
 SVN-Repository.....86  
 SVN-Verzeichnisse auschecken .....90  
 SVN-Verzeichnisse v. Free Pascal. .92  
 Symbol in Titelleiste.....373  
 Symbole/Bilder/Trenner.....527  
 Symbolische Links.....318  
 Symbolleiste, Icon.....291  
 Synapse .....647, 656  
 Synapse-basierter TCP-Server .....655  
 Synaser .....586  
 Synchronisationsobjekte.....632  
 SynEdit.....19, 243, 523  
 Syntaxhervorhebung .....27, 197  
 Syntaxmodus .....149  
 Systemdatei .....578

Systemdialoge, Ergebniskonst..... 379  
 Systemfarben .....599  
 Systemkodierung ermitteln.....83  
 Systemschrift f. Hinweistexte.....403  
 Systemsteuerung, Fenst. a. Kind .268  
 Systemsteuerungs-Applet .....266  
 Systemverzeichnis, Windows.....266  
 System-Zeichensatz .....258

## T

TabletPC.....305  
 Tabulator in Leerzeichen umwand.122  
 Tabulatorreihenfolge im Fenster ..482  
 TActivableServicImplementation...658  
 Tags.....91  
 TApplication .....357, 359  
 —, Eigenschaften.....371  
 —, Ereignisse.....372  
 —, Methoden .....360, 372  
 —, öffentliche Methoden.....360  
 TApplicationProperties.....372  
 TargetOS .....295  
 Taskleiste d. Desktops, Icon auf .482  
 Tastaturkürzel betätigen, Ereign. .374  
 Tastaturshortcut/Hotkey ausführ. 287  
 Taste drücken/loslassen, Event ...414  
 Tastendrücke an Fenster senden.373  
 Tastenereignis, reagieren auf.....458  
 Tastenkombination zuweisen.....120  
 Tastenkürzel .....430  
 — auf Windows-Desktop .....301  
 TBarchart .....416  
 TBaseComponentEditor.....286  
 TBaseServicImplementation.....658  
 TBCDField .....698  
 TBevel .....416  
 TBitBtn .....465  
 TBitButton .....416  
 TBLOBField .....698  
 TBlockSerial .....587  
 TBrush .....604  
 TButton.....416, 465  
 TButtonEdit .....418  
 TCalendar .....416  
 TCanvas .....591  
 —, Zeichenmethoden.....598  
 TCheckbox .....417  
 —, Eigenschaften.....473  
 TCheckboxgroup.....417  
 TCheckBoxListBoxEditor .....288  
 TCollection .....350  
 TColorDialog .....418  
 TCombobox.....417  
 —, Eigenschaften.....477  
 TComponent .....349  
 —, Funktionen .....526  
 —, Nachfahren .....405  
 —, Streamingfähigkeit .....526  
 TComponentEditor.....286  
 TControl.....348, 357, 411, 457, 458  
 —, Events .....413

—, Haupteigenschaften .....412  
 TCP/IP.....647  
 TCP-Versand .....656  
 TCustomApplication .....247, 256  
 TCustomEditButton, abgeleitete  
 Steuerelemente.....518  
 TDatamodule .....691  
 TDatamodule-Instanzen, Anzahl...403  
 TDataSet .....688  
 —, Ableitungen v.....717  
 —, Eigenschaften v. ....692  
 —, Ereignisse f. Fehlerprüfung...707  
 TDataSetNotifyEvent .....708  
 TDataSource .....691, 715  
 TDateField .....698  
 TDateFormat .....579  
 TDateTime-Feld anzeigen .....716  
 TDBCcalendar .....716  
 TDBCcheckbox .....716  
 TDBCcomboBox .....716  
 TDBedit.....716  
 TDBF .....717, 725  
 —, Eigenschaften von .....724  
 TDBgrid .....483, 716  
 TDBimage.....716  
 TDBlabel .....716  
 TDBlistbox .....716  
 TDBmemo .....716  
 TDBnavigator .....716  
 TDBradiogroup.....716  
 TDefaultEditor.....287  
 TDragObject .....438  
 TDrawGrid .....483  
 TEdit.....418, 469  
 —, Eigenschaften .....468  
 —, Ereignisse .....469  
 TEditButton, Eigenschaften .....517  
 Telefon-API.....306  
 Terminal, Kommunikation mit.....253  
 Testdaten erzeugen .....742  
 Text auf Steuerelement .....430  
 Text ohne Hervorhebungen .....416  
 Text zeichnen .....597  
 Text, statischer .....481  
 Text, Systemschrift.....403  
 Textausgabe.....608, 615  
 Textdatei, Zugriff auf.....556  
 Textdateien, Standardvariablen ...561  
 Textdateien, Zeilenumbrüche .....562  
 Texteingabefeld, ein-/mehrzeilig ..469  
 Textfile.....555  
 Textlabel.....467  
 Text-Methoden .....615  
 Textmodus-Anwenderschnittst...253  
 TField .....690  
 —, Eigenschaften v. ....695  
 —, Standard-Ableitungen .....698  
 — Instanzen.....693  
 TFields .....695  
 TFileStream.....349, 563  
 —, Datei öffnen.....564

- TFixedDataset ..... 717  
TFloatField ..... 698  
TFont, Eigenschaften ..... 284, 610  
TForm ..... 357  
—, Abkömmling ..... 374  
—, Eigenschaften ..... 373, 383  
—, Ereignisse ..... 374  
—, Instanz, Kinder ..... 352, 374  
—, Methoden ..... 375  
TForm-Instanzen, Anzahl ..... 403  
TFrDesign ..... 755  
TGraphic, Eigenschaften ..... 620  
Thread ..... 631, 639  
— anhalten ..... 641  
Thread-Manager ..... 640  
Thread-Objekte ..... 632  
Thread-Priorität ..... 642  
Thread-Synchronisation ..... 643  
TIBConnection ..... 729  
TImage ..... 416, 481  
—, Bestandteile ..... 619  
—, Methoden/Eigenschaften ..... 616  
TImage/TImageList ..... 615  
Timesharing ..... 639  
TIndexDef ..... 688  
TInfile ..... 545  
TIntegerField ..... 698  
Titelleistensymbole anzeigen ..... 373  
Titelzeile e. Fensters, Symbole ..... 388  
TLabel ..... 416  
TLabelEdit ..... 482  
TLargIntField ..... 698  
TLCLComponent ..... 357  
TList ..... 350  
TListBox ..... 417, 474  
—, Eigenschaften ..... 476  
TListItem, Eigenschaften ..... 501  
TListView ..... 418  
—, Eigenschaften ..... 503  
TMainMenu ..... 460  
TMemDataset ..... 717, 719  
TMemo ..... 418, 469  
TMemoField ..... 698  
TMemoryStream ..... 349, 566  
TMenu ..... 416  
TMySQL4xConnection ..... 729  
TNotifyEvent ..... 367  
TODBCConnection ..... 729  
Toolbar ..... 507  
—, Bilder anzeigen ..... 508  
—, dynamische Buttons auf ..... 509  
—, Eigenschaften ..... 508  
—, Eigenschaft d. Schaltfläch. .... 508  
Toolchain ..... 311  
Toolkit-Versionen ..... 320  
Tools API ..... 246  
Tooltip teilen ..... 506  
Toolwindow ..... 389  
TOpenDialog ..... 546  
TOpenFileDialog ..... 418  
TOpenPictureDialog ..... 546  
TOracleConnection ..... 729  
TortoiseSVN ..... 87  
—, auschecken ..... 95  
TPagecontrol ..... 418  
TPaintbox ..... 416  
TPairSplitter ..... 427  
TParadox ..... 717  
TParameter ..... 688  
TPen ..... 600  
TPersistent ..... 349  
TPopupMenu ..... 416, 460  
TPopupNotifier ..... 416  
TPQConnection ..... 729  
TPrintDialog ..... 593  
—, Eigenschaften ..... 593  
TPrinter ..... 590  
—, Methoden/Eigenschaften ..... 592  
TPrinterDialog ..... 418  
TProcess, Eigenschaften ..... 633  
TProcess, Methoden ..... 636  
TProgressBar ..... 416  
TPropertyAttributes, Felder ..... 284  
TPropertyEditor ..... 282  
—, Methoden ..... 283  
TRadioButton ..... 417, 474  
TRadiogroup ..... 417  
—, Eigenschaften ..... 478  
Transparenzdefinition ..... 621  
Transportprotokoll ..... 655  
TreeView ..... 500  
Trennzeichen eines Pfads ..... 298  
Trennzeichen f. Laufwerk/Pfad ..... 298  
Trennzeichen f. Listeneinträge ..... 298  
Trennzeichen f. Suchpfade ..... 298  
Truncate ..... 560  
Trunk ..... 91  
TSaveDialog ..... 546  
TSaveFileDialog ..... 418  
TSavePictureDialog ..... 546  
TScreen ..... 402  
—, Eigenschaften ..... 403  
—, Ereignisse ..... 404  
TSDFDataset ..... 717, 721  
—, Eigenschaften ..... 721  
TSelectDirectoryDialog ..... 418, 546  
TShape ..... 416, 481  
TSmallIntField ..... 698  
TSpeedbutton ..... 416  
TSpinEdit ..... 418, 518  
TSplitter ..... 427  
TSQLConnection ..... 728  
—, Ableitung von ..... 729  
—, Ableitung, Eigenschaft v. .... 730  
TSQLite3Connection ..... 729  
TSQLQuery ..... 717, 728, 732  
—, WHERE-Klausel ..... 734  
TSQLTransaction ..... 728  
TStream ..... 349  
TStream-Ableitungen ..... 563  
TStringField ..... 698  
TStringGrid ..... 418, 483  
TStringList ..... 350, 393  
TStrings ..... 350  
TStringStream ..... 567  
TSynEdit ..... 223  
TThread, Method./Eigenschaft ..... 641  
TTimeField ..... 698  
TToggleBox ..... 472  
TToolbar ..... 416  
TToolbutton ..... 416  
TTrayIcon ..... 482  
TTreeView ..... 418, 493  
—, Eigenschaftem ..... 492  
TWinControl ..... 357, 411, 413  
—, Eigenschaften ..... 414  
—, Events ..... 414  
TWordField ..... 698  
TXMLConfig ..... 73  
Typ der Felddaten ..... 695  
typ\_lib\_edtr.exe ..... 657  
Typ-Deklarationsblock ..... 35, 38  
Type Library Editor ..... 657  
Typenbibliotheks-Editor ..... 657  
Typinformationen zur Laufzeit ..... 351  
Typisierte Datei ..... 555
- 
- U**
- Übersetzungsdatei ..... 77  
Überwachte Ausdrücke ..... 237  
Uhrzeitabweichung ..... 18  
UIB ..... 684, 718  
Umgebungsvariable, Wert einer ..... 371  
Umlaute ..... 81, 181  
Umschaltbox ..... 472  
Unicode ..... 81, 122, 610, 725  
Unit ..... 247  
Unit Classes ..... 40, 349  
Unit DateUtils ..... 40  
Unit des Hauptforms, Quelltext ..... 250  
Unit FileUtil ..... 82  
Unit in mehrere Projekte  
— einbinden ..... 68  
Unit LCLProc ..... 82  
Unit Lconvencoding ..... 83  
Unit Math ..... 40  
Unit m. neuer Komponentenkl. .... 165  
Unit nicht gefunden ..... 75  
Unit StrUtils ..... 40  
Unit System ..... 40  
Unit SysUtils ..... 40  
Unit, kompilierte ..... 66, 252  
Unit, Name ändern ..... 66  
Unit1 ..... 115  
Unit-Abhängigkeiten ..... 68, 135  
Unit-Aufbau ..... 39  
Unit-Bezeichner ..... 534  
Unit-Konzept ..... 32, 39  
Unit-Pfade ..... 136  
Units f. versch. Plattformen ..... 70  
Units mit selbem Namen ..... 137  
Units, Portabilität ..... 536  
Units, Suche nach ..... 40

Units, unbenutzte .....	29
Units, vorkompilierte .....	54
Unit-Testing .....	275
Unix, Datei öffnen .....	318
Unix, Dateiadung .....	317
Unix, Fensterverhalten .....	390
Unix, grafische Oberflächen .....	320
Unix, Pfadtrenner .....	317
Unix, Schreibzugriffe .....	318
Unix, Verzeichnisgröße/- datum .....	318
Unix, Verzeichnishierarchie .....	318
Unterrfenster .....	17
Untypisierte Datei .....	556
UnusedHandle .....	298
Update-Anw., Primärschlüssel .....	734
Ursprungskoordinaten, Grafik .....	595
USB-Gerät .....	585
User-Abfragen .....	415
Uses-Anweisung, Komponente hinzufügen .....	249
Uses-Klausel .....	40
—, Portierung .....	538
UTF-16 .....	302
UTF-8 .....	81f, 725
UTF8ToAnsi .....	82
UTF8ToSys .....	547

**V**

valgrind .....	33, 153
varargs .....	48
Variablen, lokale erzeugen .....	124
Variablenanzeige .....	237
Variablen-Deklarationsblock .....	38
Variablen Deklarationsvervoll- ständigung .....	229
Variablenvervollständigung .....	24
Variante Record .....	35
Veränderungen in DB melden .....	691
Verfügbarkeitssteuerung eines Controls .....	458
Verlinkte Liste .....	36
Veröffentlichte Eigenschaften in Datenstrom speichern .....	350
Verschlüsselungs-Stream .....	567
Versionskontrollsystem, Schlüsselworte .....	123
Versionsverwaltung .....	91
—, Download aus .....	87
Versteckte Datei .....	578
Verstecktes Fenster anzeigen .....	381
VertScrollbar .....	373
Verzeichnis aufräumen .....	117
Verzeichnis aufbau anzeigen .....	493
Verzeichnis auswahl .....	418
Verzeichniseinträge ausgeben .....	574
Verzeichnisinhalt anzeigen .....	519
Verzeichnisse durchsuchen .....	573
Verzeichnissuche, rekursive .....	573
Verzeichnistrenner .....	61, 297
Verzeichniswechsel .....	576

Virtual Methode .....	43
Virtual PC .....	307
Virtuelle Units .....	70
Visible .....	711
Visuelle Klassen .....	348
Visuelle Komponenten .....	347
Voreinstellungen von Laz .....	190
Vorwärtsdeklaration .....	21, 37
Vorzeichenbehafteter 32-Bit/ 64-Bit-Integer .....	338
Vorzeichenlose 32-Bit/ 64-Bit-Integer .....	338

**W**

Web Service Toolkit .....	655
Webservice als Bibliothek .....	665
Webservice registrieren .....	663
Webservice, Nachrichten- Logging .....	677
Webservice, Stadien d. Bearbeitg. von Clientanfragen .....	681
Webservice, URL .....	654
Webservice-Erweiterungen .....	681
Webservice-Implementations- klasse .....	663
Webservice-Module .....	656
Weiß, Kodierung .....	599
Werkzeuge .....	176
Werteliste füllen .....	283
WHERE-Klausel, TSQLQuery .....	734
WideChar .....	34
WideString .....	34, 302
Widgets .....	405
WidgetSet .....	15
— wechseln .....	76, 242
—, Eigenschaften von .....	133
—, Kommunikation mit .....	527
—, LCL-Interface zum .....	525
—, mögliche .....	241
Wiederholen-Button, Ergebnis .....	379
Win32/64 .....	241
Win32/Win64-API .....	348
Windows CE .....	98, 241, 305
—, API .....	306
—, Bluetooth .....	306
—, Build-Umgebung .....	309
—, Dialoge .....	306
—, Fenstersteuerung .....	314
—, Kurznachrichten .....	306
—, LCL .....	313
—, PIM-Daten .....	306
—, Schließen-Schalter .....	316
—, Shellroutinen .....	306
—, Standard-Steurelemente .....	306
—, Synchronisationsmodule .....	306
—, Systemsteuerung .....	306
—, virtuelle Objekte .....	306
Windows CE/Mobile .....	305, 348
Windows, Datei öffnen .....	299
Windows, installieren auf .....	98

Windows, Kern-API-Routinen .....	300
Windows, Pfadcodierung .....	303
Windows, Unicode .....	300, 302
Windows, Verzeichnisstruktur .....	303
Windows-API .....	299
Windows-API-Funktionen, Wrapper um .....	529
Windows-API-Nachrichten .....	59
Windows-Ressourcendatei .....	250
WindowState .....	373
— ändern, Ereignis .....	374
—, Konstanten .....	391
Windows-Units markieren .....	60
Windows-Verzeichnisname ermitteln .....	304
Wortumbruch .....	470
Write .....	560
WSDL .....	655, 657, 673f.
wsMaximized .....	391
wsMinimized .....	391
wsNormal .....	391
WST .....	655
—, Serialisierungs- und Transportprotokoll .....	669
—, Servicebeschreibung .....	657
WST-Laufzeitumgebung .....	664

**X**

XML-Datei lesen .....	73
XML-Schema-Definition parsen .....	673

**Z**

Zeichen maskieren .....	126
Zeichenfläche .....	482, 595
Zeichenkettenbearbeitung, Unit .....	40
Zeichenkettenlisten .....	350
Zeichenobjekt hat Fokus .....	612
Zeichenpinsel, Eigenschaften .....	598
Zeichensatz .....	610
Zeichensatzumwandlungen, API- Funktionen .....	553
Zeichenstift, Eigenschaften .....	600
Zeichentabelle .....	122
Zeichnen in eine Bitmap .....	597
Zeichnen v. Rahmen .....	416
Zeiger, typisierter .....	45
Zeigergröße, 64-Bit-Prozessor .....	338
Zeilenende-Zeichen .....	181, 297, 560
Zeilenposition d. Cursors im Dataset speichern .....	692
Zeilenumbrüche in Dateien .....	562
Zeos .....	684, 718
Zieh-Operation, Ereignis .....	413
Zielbetriebssysteme, mögliche .....	293
Ziel-CPUs, mögliche .....	293
Zieldateiformate, speichern in .....	513
Zielplattform .....	151
—, Einstellungen .....	151
Zielverzeichnis .....	141
Zwischenablage .....	118