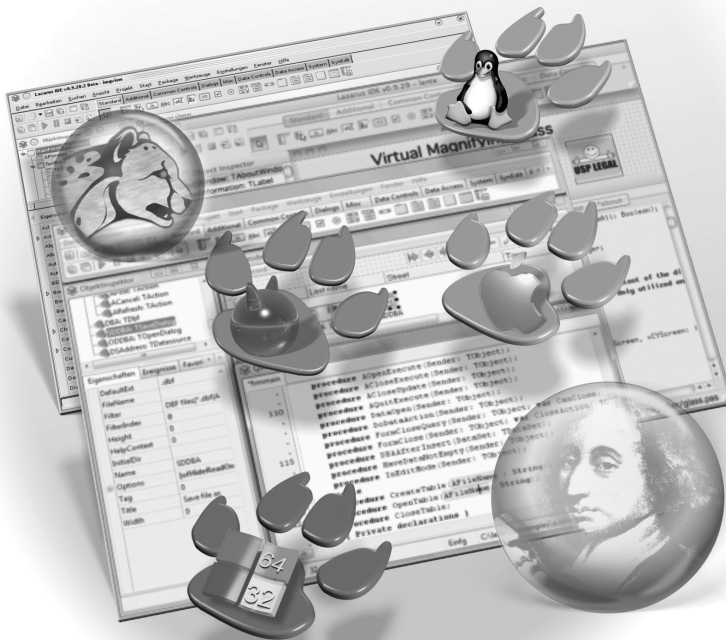




Computer & Literatur Verlag GmbH

LAZARUS

von Michaël Van Canneyt, Mattias Gärtner,
Swen Heinig, Felipe Monteiro de Carvalho,
Inoussa Ouedraogo 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.

1. Auflage 2010

© 2010 by C&L Computer und Literaturverlag
Zavelsteiner Straße 20, 71034 Böblingen
E-Mail: 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 978-3-936546-38-5

INHALT

| | |
|----------------------|-----------|
| Vorwort | 13 |
|----------------------|-----------|

| | |
|--|-----------|
| Kapitel 1 | |
| Die Architektur von Lazarus | 15 |

von Matthias Gärtner

| | |
|--|-----------|
| 1.1 Der Quelltexteditor | 19 |
| 1.1.1 CodeTools..... | 20 |
| 1.1.2 Schnelle Navigation im Code | 20 |
| 1.1.3 Automatische Vervollständigung | 23 |
| 1.2 Basiswissen zur Sprache | 30 |
| 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 |
| 1.3 Projekte | 52 |
| 1.4 Komponenten..... | 55 |
| 1.5 Packages | 56 |
| 1.5.1 C-Bibliotheken in Free Pascal..... | 58 |
| 1.5.2 Free-Pascal-Bibliotheken in C | 60 |
| 1.5.3 Neue Komponente in der IDE installieren..... | 61 |
| 1.5.4 Units in mehreren Projekten..... | 64 |
| 1.5.5 Virtuelle Units | 65 |
| 1.5.6 Units für verschiedene Plattformen | 65 |
| 1.5.7 Packages suchen | 68 |
| 1.5.8 Verzeichnisse und Suchpfade..... | 69 |

| | |
|--|-----------|
| 1.5.9 Abhängigkeiten und Vererbung | 71 |
| 1.5.10 Kompilierung | 71 |
| 1.6 Quelltextdokumentation | 75 |
| 1.7 Unicode | 77 |

Kapitel 2

Lazarus installieren 81

von Jörg Braun, Swen Heinig und Felipe Monteiro de Carvalho

| | |
|---|------------|
| 2.1 Download aus der Versionsverwaltung..... | 83 |
| 2.1.1 Die Installation von TortoiseSVN | 83 |
| 2.1.2 Das Programmpaket Subversion..... | 84 |
| 2.1.3 SVN-Verzeichnisse von Lazarus und FPC..... | 87 |
| 2.1.4 Basisarbeiten auf der Kommandozeile..... | 89 |
| 2.1.5 Auschecken mit TortoiseSVN..... | 91 |
| 2.2 Installieren auf Windows | 94 |
| 2.3 Installieren auf Linux | 98 |
| 2.4 Installieren auf FreeBSD | 101 |
| 2.5 Installieren auf MacOS X | 117 |

Kapitel 3

Die IDE 119

von Swen Heinig

| | |
|-----------------------------------|------------|
| 3.1 Das Lazarus-Menü | 120 |
| 3.1.1 Datei..... | 120 |
| 3.1.2 Bearbeiten..... | 122 |
| 3.1.3 Suchen..... | 129 |
| 3.1.4 Ansicht | 134 |
| 3.1.5 Projekt..... | 141 |
| 3.1.6 Compilereinstellungen..... | 149 |
| 3.1.7 Start | 160 |



| | |
|--|------------|
| 3.1.8 Package | 162 |
| 3.1.9 Werkzeuge | 176 |
| 3.1.10 Einstellungen | 183 |
| 3.1.11 Fenster | 216 |
| 3.1.12 Hilfe | 216 |
| 3.2 Tools | 217 |
| 3.2.1 Objektinspektor | 217 |
| 3.2.2 Quelltexteditor | 222 |
| 3.2.3 Quelltext vervollständigen | 226 |
| 3.2.4 Message Composer | 229 |
| 3.2.5 Debugger | 231 |
| 3.3 IDE rekompilieren | 237 |

Kapitel 4

Projekte..... 243

von *Felipe Monteiro de Carvalho*

| | |
|---|-----|
| 4.1 GUI-Anwendungen | 246 |
| 4.2 Konsolenanwendungen | 252 |
| 4.3 DLLs und Shared Objects..... | 257 |
| 4.3.1 Dynamische Bibliotheken | 258 |
| 4.3.2 Bibliotheken in MacOS X | 263 |
| 4.3.3 Systemsteuerungs-Applets für Windows..... | 265 |
| 4.4 CGI-Anwendungen | 273 |
| 4.4.1 CGI-Programme in Pascal | 275 |
| 4.4.2 CGI mit Powtils | 276 |
| 4.5 Unit-Testing..... | 279 |
| 4.6 Packages | 280 |
| 4.6.1 Komponenten einbinden | 285 |
| 4.6.2 Komponenten registrieren | 285 |
| 4.6.3 Eigenschafts-Editoren (Property-Editoren) | 286 |
| 4.6.4 Komponenten-Editoren..... | 290 |
| 4.7 Dienste und Daemonen..... | 293 |

Kapitel 5

Zielplattformen 297

von *Felipe Monteiro de Carvalho*

| | |
|--|-----|
| 5.1 Plattformspezifisches | 301 |
| 5.1.1 Das API von Windows32/64 | 303 |
| 5.1.2 Spezialfall Windows CE | 309 |
| 5.1.3 Linux, FreeBSD und andere Unix-Systeme | 321 |
| 5.1.4 Die APIs von MacOS X..... | 328 |
| 5.1.5 32 Bit und 64 Bit | 342 |
| 5.2 Konfigurationsdateien | 343 |
| 5.3 Ressourcendateien | 346 |

Kapitel 6

Die Klassenbibliotheken..... 351

von *Michaël Van Canneyt*

| | |
|---|-----|
| 6.1 Die RTL (FPC-Laufzeitbibliothek) | 353 |
| 6.1.1 Laden und Speichern mit Datenströmen..... | 354 |
| 6.1.2 Nomenklatur der Klasseninstanzen | 356 |
| 6.2 Die FCL (Free Component Library) | 358 |
| 6.3 Die LCL (Lazarus Component Library) | 360 |
| 6.3.1 Das Anwendungsobjekt | 363 |
| 6.3.2 Bildschirmfenster | 378 |
| 6.3.3 Arbeiten mit TForm..... | 379 |
| 6.3.4 Die Eigenschaften von TForm..... | 387 |
| 6.3.5 Spezialfenster | 399 |
| 6.3.6 Die Fensterumgebung | 406 |
| 6.3.7 Die Steuerelemente im Fenster | 409 |
| 6.3.8 Layout und Programmwurf..... | 422 |
| 6.3.9 Actions | 432 |
| 6.3.10 Drag&Drop | 438 |
| 6.4 Die Elemente der Komponentenpalette | 449 |
| 6.4.1 Der Reiter Standard | 453 |
| 6.4.2 Der Reiter Additional | 474 |
| 6.4.3 Der Reiter Common Controls..... | 486 |



| | |
|-------------------------------------|-----|
| 6.4.4 Der Reiter Dialogs | 505 |
| 6.4.5 Der Reiter Misc | 511 |
| 6.4.6 Der Reiter Data Controls..... | 516 |
| 6.4.7 Der Reiter Data Access | 516 |
| 6.4.8 Der Reiter System | 516 |
| 6.4.9 Der Reiter SynEdit | 518 |

Kapitel 7

Delphi-Komponenten portieren 519

von Michaël Van Canneyt und Mattias Gärtner

| | |
|--|-----|
| 7.1 Die Architektur von Lazarus-Komponenten | 519 |
| 7.1.1 Plattformunabhängige Schicht | 520 |
| 7.1.2 Plattformabhängige Schicht | 522 |
| 7.2 Komponentenmodelle | 523 |
| 7.3 Praxis der Portierung | 531 |
| 7.3.1 Von der Komponente zum Lazarus-Package | 535 |
| 7.3.2 Die Komponentenpalette | 537 |

Kapitel 8

Dateien und Geräte 539

von Felipe Monteiro de Carvalho und Jörg Braun

| | |
|-------------------------------------|-----|
| 8.1 Dateidialoge in Lazarus..... | 539 |
| 8.2 Arbeiten mit Dateien | 548 |
| 8.3 Verzeichnisse durchsuchen | 567 |
| 8.4 Kommunikation mit Geräten | 573 |
| 8.4.1 Der Parallelport | 574 |
| 8.4.2 Serielle Kommunikation..... | 579 |
| 8.4.3 Der Drucker | 584 |

Kapitel 9

Grafikprogrammierung 589

von *Felipe Monteiro de Carvalho*

| | |
|---|-----|
| 9.1 Die Zeichenfläche..... | 589 |
| 9.1.1 Farben..... | 592 |
| 9.1.2 TPen..... | 594 |
| 9.1.3 TBrush..... | 598 |
| 9.1.4 Fonts..... | 601 |
| 9.1.5 Die wichtigsten Grafikroutinen..... | 604 |
| 9.2 Grafikkomponenten..... | 609 |
| 9.3 Die Grafikformate..... | 614 |
| 9.3.1 TGraphic..... | 614 |
| 9.3.2 TRasterImage..... | 615 |
| 9.3.3 Bitmaps..... | 618 |
| 9.3.4 TJpegImage..... | 619 |
| 9.3.5 Icons..... | 620 |

Kapitel 10

Prozesse und Threads..... 625

von *Felipe Monteiro de Carvalho*

| | |
|--------------------|-----|
| 10.1 Prozesse..... | 626 |
| 10.2 Threads..... | 637 |

Kapitel 11

Netzwerk-Programmierung 645

von *Inoussa Ouedraogo*

| | |
|--------------------------------|-----|
| 11.1 TCP/IP programmieren..... | 645 |
| 11.1.1 Das Clientprogramm..... | 647 |
| 11.1.2 Das Serverprogramm..... | 650 |



| | |
|---|------------|
| 11.2 Webservice | 652 |
| 11.2.1 Programmierung des Servers..... | 654 |
| 11.2.2 Programmierung des Clients | 670 |
| 11.2.3 Nachrichten-Logging..... | 675 |
| 11.2.4 Objektpooling..... | 676 |
| 11.2.5 Serviceerweiterungen | 679 |

Kapitel 12

Der Zugriff auf Datenbanken 681

von Michaël Van Canneyt

| | |
|---|------------|
| 12.1 Die Architektur | 681 |
| 12.1.1 Datenbankzugriff..... | 681 |
| 12.1.2 Datenbank auswählen..... | 683 |
| 12.1.3 Hilfsanwendung: Der Daten-Desktop..... | 687 |
| 12.2 Die Datenbank-Zugriffsklassen | 688 |
| 12.2.1 Das Dataset | 688 |
| 12.2.2 Das Datenmodul | 711 |
| 12.2.3 Visuelle datenbezogene Steuerelemente | 713 |
| 12.3 Ableitungen von TDataSet | 715 |
| 12.4 Der Daten-Desktop | 733 |
| 12.4.1 Das Data Dictionary (Daten-Verzeichnis)..... | 735 |
| 12.4.2 Daten exportieren..... | 740 |
| 12.4.3 Codegenerierung | 742 |
| 12.5 Crashkurs SQL | 744 |
| 12.6 Berichte (Reports) | 748 |
| 12.6.1 Bericht anlegen | 750 |
| 12.6.2 Der Report-Designer..... | 753 |

Stichwortverzeichnis 755



VORWORT

Liebe Leser,

es gibt sicherlich gute Gründe, warum Sie dieses Buch 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-bittiger 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 entsprechenden 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 Datenbankanbindung 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 mit 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, hat die Aufgabe auf sich genommen, die IDE 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 Kamerun 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 den größten Teil von Lazarus 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 Matthias 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 for 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 die 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, BSD 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 auf die Plattformunabhängigkeit großer Wert 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-Klon, 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 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.

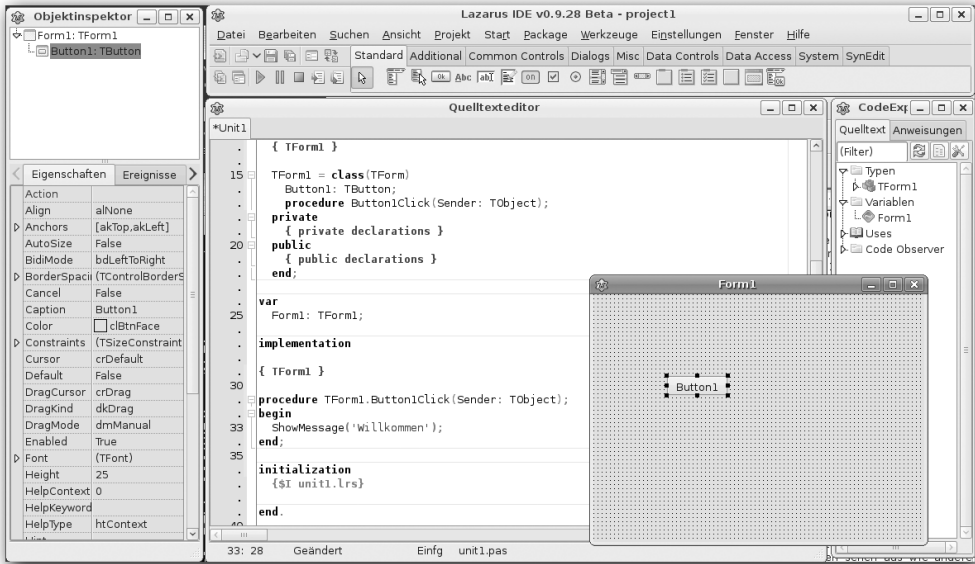


Bild 1.1: Delphi-Anwender werden vieles bereits kennen: Formulareditor, Objektinspektor, CodeExplorer

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, 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 Windowsprogramme unter Linux oder WinCE-Anwendungen unter Windows zu erzeugen. 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 denen spezieller Code je nach Betriebssystem, CPU oder 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. Dies sorgt gerade bei mehr als zwei Plattformen für eine saubere und übersichtliche Struktur. 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. Es sollen automatisch die veränderten Dateien in der richtigen Reihenfolge kompiliert werden und es müssen 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 Netzwerkverzeichnissen 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 fertigkompilierte `.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. Dies bedeutet, daß der Compiler automatisch die Dateien sucht und automatisch alles Veränderte in der richtigen Reihenfolge kompiliert.

Die Compilerparameter sind im Gegensatz zu den 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 Run Time 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 Bauen 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 erklärt, wie der Lazarus-Quelltexteditor mit diesem Wissen beim Programmieren selbst 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 mit beliebiger Kombination der Modifizierertasten. Unter Windows sind die Modifizierertasten [Strg], [Shift] und [Alt].

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

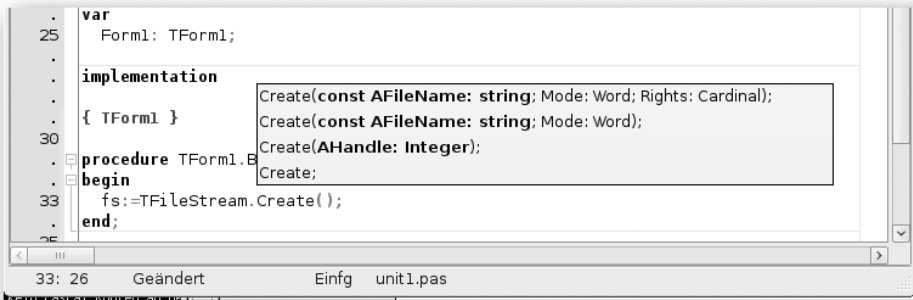


Bild 1.2: Mit [Strg]+[Shift]+Leertaste lassen sich die Parameterhinweise einblenden

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. Damit wird es erleichtert, konditionalen Code zu durchsuchen oder include-Direktiven zu finden.

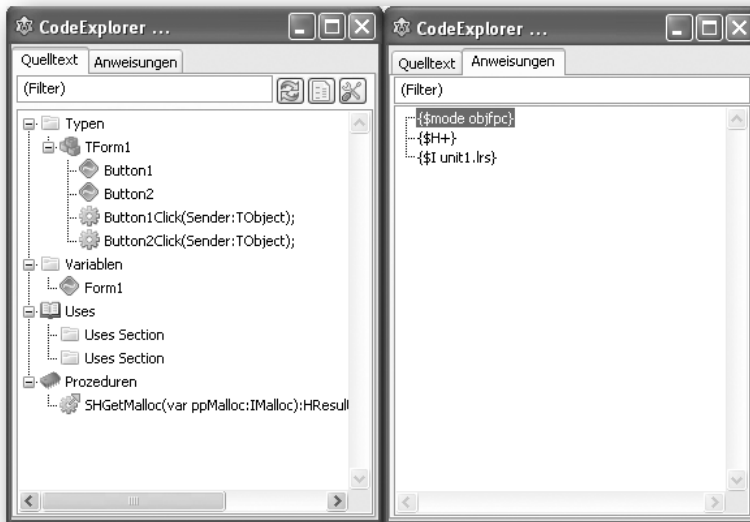


Bild 1.3: Die beiden Seiten des Code-Explorers

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 erzeugen. Dabei erkennt sie Property-Getter/-Setters und rät 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
  Active: 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 dasselbe 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;

```

der Cursor auf der Variable *i* in der *for*-Anweisung, wird nach dem Betätigen der Codevervollständigungstaste die Variable *i* als Integer definiert:

```

procedure TForm1.Button1Click(Sender: TObject);
var

```


STICHWORTVERZEICHNIS

\$

| | |
|---|---------------|
| \$(LazarusDir) | 53 |
| \$(TargetOS) | 299 |
| \$IFDEF ... \$ENDIF | 298 |
| \$include | 298 |
| .compiled-Datei | 250 |
| .cpl-Datei | 265 |
| .exe | 143 |
| .ico-Datei | 248 |
| .lfm-Datei. 17, 57, 142, 217, 248, 354 | |
| .lpi-Datei..... 19, 52, 143, 248, 249 | |
| .lpk-Datei..... 54, 68, 282 | |
| .lpr-Datei 19, 52, 142, 146, 248f., 363 | |
| .lrs-Datei | 142, 248, 250 |
| .manifest-Datei | 248 |
| .o-Datei | 250 |
| .or-Datei | 250 |
| .pas-Datei | 142, 248 |
| .po-Datei | 54, 72, 149 |
| .ppu-Datei | 53, 142, 250 |
| .rc-Datei | 248, 250 |
| .res-Datei | 250 |
| { \$apptype console | 252 |
| { \$apptype gui | 252 |
| { \$IFDEF MSWindows } | 56 |
| 32 Bit 64 Bit, Prozessor | 342 |
| 32-Bit-Programm | 303 |
| 64-Bit-Betriebssys., RAM | 303 |
| 64-Bit-Lazarus | 97 |
| 8-Bit-ISO-Zeichen | 78 |

A

| | |
|------------------------------------|------------|
| Abbrechen-Schaltfläche | 383, 459 |
| Abfragedialog, benutzerdef. | 404 |
| Abfrageelemente | 420 |
| Abfragen, tabellenübergreifende .. | 684 |
| abstract Methode | 43 |
| abstrakte Deklaration | 44 |
| Actionklassen | 435 |
| Actionliste | 423f., 474 |
| Actions, Eigenschaften | 434 |
| Actions, neue definieren | 435 |
| Active | 690 |
| ActiveControl | 377 |
| ActiveSync | 311f. |
| Addon-Installer | 313 |
| ADS/Advantage Datenbank. 682, 716 | |
| Akkumulator-Register | 258 |
| aktuelles Fenster schließen | 389 |

| | |
|--|-------------------|
| Align | 425, 709 |
| Alle-Button, Ergebnis | 383 |
| Alles ablehnen/bestätigen- Button, Ergebnis | 383 |
| AllFilesMask | 302 |
| AllowDropFiles | 377 |
| Allzweckklassen | 351 |
| Anker | 427 |
| Ankreuzfeld | 421, 467 |
| ANSI | 126 |
| Ansicht v. Fenstern | 134 |
| Ansi-String | 34, 152, 257, 306 |
| AnsiToUTF8 | 77 |
| Anwendereingabe, Reaktion auf. 374 | |
| Anwendung anhalt. b. Exception. 375 | |
| Anwendung geht in Idle-Zustand. 376 | |
| Anwendung wiederherstellen | 376 |
| Anwendung wird minimiert | 376 |
| Anwendungsfenster duplizieren... 389 | |
| Anwendungs-Icon | 524 |
| Anwendungsobjekt | 375 |
| Anwendungstitel | 146, 375 |
| Anwnd.-Actions zentr. sammeln. 433 | |
| AnyDAC | 716 |
| API-Fkt. i. Pascal suchen | 305 |
| Append | 553, 699, 700 |
| Application Bundle | 329 |
| Application.CreateForm | 146 |
| Application.Initialize | 367 |
| Application.Run | 383 |
| Application.Title | 146 |
| ApplicationType | 317 |
| Applikationsserver, Webservice einbetten in | 656 |
| ApplyUpdates | 731 |
| Arbeitsergeb. zwischenspeichern 569 | |
| ARM-Prozessor | 313 |
| Array, dynamisches | 48 |
| Array, offenes | 49 |
| Arrays m. fester Größe definieren .47 | |
| Arrays | 45 |
| ASCII-Dateien, Zugriff auf | 715 |
| Assembler | 31 |
| Assembler-Stil | 152 |
| assert-Anweisungen | 151 |
| AsXYZ-Eigenschaften | 694 |
| AT&T-Syntax, Assembler | 152 |
| atDefault | 319 |
| atKeyPadDevice | 319 |
| atPDA | 319 |

| | |
|--|----------|
| atSmartphone | 319 |
| Aufrufkonvention register | 60 |
| Aufrufkonvention wechseln | 51 |
| Aufrufkonvention | 51, 325 |
| Aufrufmechanismen | 258 |
| Aufrufstack | 236 |
| Aufzählungstypen | 35 |
| Aufzählungstypen-Editor | 287 |
| Ausdrücke auswerten | 235 |
| Ausführbare Datei | 143 |
| Ausnahme anzeigen | 376 |
| Ausnahmebehandlung bei DB | 702 |
| Ausnahmebehandlung, Unit | 40 |
| Ausnahmen in Hauptschleife | 375 |
| Auswahlfeld | 421, 468 |
| Auszuführendes Programm, Name | 627 |
| AutoScroll | 377 |

B

| | |
|--|-----|
| Barcodes auf Bericht | 748 |
| Basisklasse aller Fenster | 361 |
| Basisklasse aller Klassen | 352 |
| Basisklasse aller Komp. d. LCL | 361 |
| Basisklasse d. visuellen Komp. | 361 |
| Basispakete der IDE | 284 |
| Baumansicht durchlaufen | 491 |
| Baumansicht | 486 |
| Baumansicht, Eigenschaften | 487 |
| Baumansicht, Eintrag auswählen. 487 | |
| Baumansicht, Knoten editieren | 494 |
| Baumansicht, Knoten hinzufügen. 489 | |
| Baumansicht, Knotenbeschrift. 488 | |
| Baumknoten, automat.expand | 487 |
| Bedingte Definitionen | 300 |
| BeforeCancel | 705 |
| BeforeClose | 705 |
| BeforeDelete | 705 |
| BeforeEdit | 705 |
| BeforeInsert | 705 |
| BeforeOpen | 705 |
| BeforePost | 705 |
| BeforeRefresh | 705 |
| BeforeScroll | 705 |
| Benachrichtigungs-Popup | 505 |
| Bereichstypen | 35 |
| Bericht anlegen | 750 |
| Bericht, Bandtypen | 749 |
| Bericht, Datenband | 749 |
| Bericht, druckb. Elemente a. Band750 | |

Bericht, Exportformate..... 748
 Bericht, Gruppenband 749
 Bericht, Kopf-/Fußbänder..... 749
 Berichts-Komponente 748
 Berichtsvariablen 752
 Beschränkungen..... 416
 Beschriftungen, Text v..... 77
 Bestandteile von L..... 241
 Bestätigung-Button, Ergebnis..... 383
 Betriebssystem-Definitionen 300
 Betriebssystem-Exceptions..... 234
 Betriebssystemunabh. Konstanten302
 Bevel, Caption 476
 Bezeichner autom. umbenenn.28, 133
 Bezeichner automatisch anlegen.. 178
 Bezeichner d. akt. Unit anzeigen.... 22
 Bezeichner, deklariert wo 222
 Bezeichnerdeklaration suchen..... 21
 Bezeichnerdeklaration..... 36
 Bezeichner-Mehrdeutigkeit..... 32
 Bezeichnernamen..... 41
 Bezeichnerreferenzen suchen..... 133
 Biblioth., n. thread-sich. aufrufen..637
 Bibliothek aus Speicher entfernen 260
 Bibliothek statisch binden..... 58
 Bibliothek..... 19
 Bibliothek, dynamisch/statisch 257
 Bibliothek, Suchpfad 262
 Bibliotheken in MacOS X 263
 Bibliotheks-Versionsnummer 262
 Bild anzeigen 476
 Bild auf Canvas zeichnen 591
 Bild skalieren 591, 615
 Bild, Puffer anlegen f..... 591
 Bildbetrachter..... 609
 Bildelement..... 476
 Bilder anzeigen 714
 Bilder laden/speichern 540
 Bildindex in Bilderliste..... 434
 Bildklassen, Basisklasse 615
 Bildschirmauflösung..... 377, 407
 Bildschirmf. a. oberstes setzen ... 386
 Bildschirmfenst. in Textmod. öffn.412
 Bildschirmfenster autom. v.
 IDE anlegen lassen 396
 Bildschirmfenster manuell anlegen396
 Bildschirmfenster schließen 383
 Bildschirmfenster teilen 422
 Bildschirmfenster 378
 Bildschirmfenster, b. Start
 gleichzeitig zwei öffnen..... 381
 Bildspeicher 609
 Binärdatei der Anwendung,
 Verzeichnis..... 375
 Binärdatei, Zugriff auf..... 550
 Binderdatei, Webservice 656
 Binding 654
 Bitbutton..... 474
 Bitmap-Format 618
 Bitmap-Muster 600
 BLOB-Feld..... 688

Block einrücken..... 123
 Blockread/Blockwrite 557
 Blockvervollständigung 202
 Bluetooth..... 310
 BOF 690
 Bookmark..... 698
 Boolean 34
 Bordercons..... 377, 392
 BorderStyle, Eigenschaften 393
 Borland Database Engine 723
 Browser für bed. Eigenschaften... 136
 bsDialog 393
 bsNone..... 393
 bsSingle..... 393
 bsSizeable 393
 bsSizeToolWin 393
 bsToolWindow..... 393
 Build 146

C

C-/C++-Compiler, Aufrufkonv... 263
 C++-artige Inline-Anweisungen 152
 C++-Schnittstellendatei zu Qt ... 325
 caFree 385
 Cancel 699
 Canvas 589
 Canvas, Drucken auf..... 585
 Caption..... 377, 472
 Carbon..... 239, 329, 352
 C-artige Operatoren..... 151
 C-Bibliothek..... 32
 C-Bibliotheken in Free Pascal..... 58
 C-Bibliotheken in Object-Pascal.Pr.33
 CDecl 48, 58, 258
 CGI-Anwendung 273
 CGI-Anwendung, Header-Einträge275
 CGI-Anwendung, Standardoutput 275
 Char 34
 C-Headerdateien nach Pascal konv.60
 Checkbox 467, 714
 Checkbox, grayed..... 467
 Checkbox, undefinierter Status... 467
 Checkbox, Zustände..... 468
 Checkgroup..... 473
 class 44
 Clientprozeß 645
 Close..... 383, 692
 Cocoa 151, 239, 352
 Cocoa, Datentypen 339
 Cocoa, Sub-Frameworks..... 329
 Cocoa-Framework, Anbindung an 333
 Cocoa-Klassen 334
 Codeblöcke vervollständigen..... 27
 Code-Explorer 22, 136
 Codefaltung 203
 Codegröße 154
 Codepage 1250 78
 Codepage..... 77, 604
 Code-Schablonen..... 25, 211
 CodeTools 20, 203, 241
 CodeTools-Werte anzeigen..... 141

Combobox 471, 714
 Combobox, AutoComplete 470
 Combobox, Einträge in..... 470
 Combobox, Pfeiltasten z. Wechs. 470
 Comp 33
 Compilerausgaben 155
 Compilerdefinitionen 299
 Compilerdirektiven 50
 CompilerEinstellungen in/von Datei
 speichern/laden..... 159
 Compiler-Einstellungen vererben ... 54
 Compilereinstellungen 149
 Compilereinstellungen, Speichern. 52
 Compilereinstellungen, von
 Packages geerbt 156
 Compiler-Konfigurationsdatei139, 156
 Compiler-Metadaten..... 355
 Compiler-Suchpfade 139
 COM-Port 579
 conf-Datei 348
 configtest..... 344
 ConsoleApplication 375
 ConstraintError 709
 Container für Steuerelem..... 473
 Controls, datenbezogene 681, 713
 ConvertEncoding 78
 Cookie..... 273, 276
 CPIApplet..... 265
 CPU-Definitionen 299
 CPU-Zeit verteilen 625
 CreateForm..... 367
 CreateProcess..... 626
 Critical Section 642
 Crosscompiler-Version..... 315
 Crosscompiling 17, 153, 318
 CSV, Feldtrenner 718
 CSV-Datei anlegen 721
 CSV-Dateien 682
 cthreads..... 638
 C-Typen 34
 Cursor, momentaner 407
 Cursorposition im Dataset 690
 C-Werkzeuge..... 33

D

Daemon registrieren..... 295
 Daemonen 293
 Daemon-Klassen 295
 Data Dictionary in IDE einbinden .. 739
 Data Dictionary verwalten 733
 Data Dictionary z. Laufzeit
 einbinden 738
 Data Dictionary..... 733, 735
 Data Dictionary, neues anlegen... 737
 Data Dictionary, Speicherort..... 734
 Data Dictionary, SQL-
 Anweisungen generieren 737
 Database 711
 Data-Cursor setzen..... 690
 DataField..... 713
 Dataset mit DB verbinden..... 711



- Dataset prüfen auf Edit-Modus..... 698
 Dataset, Browse-Modus..... 699
 Dataset, Daten im aktualisieren.... 692
 Dataset, Eigenschaften 711
 Dataset, erstes Ereignis 705
 Dataset, leeres 691
 Dataset, Post-Methode..... 701
 Dataset-Komponenten,
 Container f. 689
 DataSource 713
 DataType 693
 Datei abschneiden..... 554
 Datei auswählen, mehr als eine... 509
 Datei erzeugen, Fehler beim 551
 Datei kopieren..... 552, 557
 Datei öffnen mit TFileStream 558
 Datei öffnen 548, 553, 633
 Datei schließen..... 553
 Datei sichern..... 507, 548
 Datei z. Lesen/Schreiben öffnen 550ff.
 Datei, blockweise Daten
 auslesen/schreiben 555
 Datei, lesen aus 556
 Datei, Moduszuweisung..... 550
 Datei, neue erzeugen/überschreib. 551
 Datei, sukzessive lesen aus 551
 Datei, ungeöffnete löschen 554
 Datei, vorhd. öffnen 551
 Datei, Werte lesen aus..... 554
 Datei-/Verzeichnisdialo-
 gkomponenten, Eigenschaften ... 540
 Dateianfang 551
 Dateianzeige, Eigenschaften 514
 Dateiattribute ermitteln/setzen.... 572
 Dateiauswahlfeld 514
 Dateidatum konvertieren 573
 Dateidatum lesen/schreiben 572
 Dateidialog, Eventhandler 540
 Dateidialoge d. Betriebssystems.. 541
 Dateidialoge 539
 Dateien a. Fenster ziehen, Ereignis 378
 Dateien auf Fenster fallenlassen... 376
 Dateien in Verzeichnis, Details zu . 495
 Dateien laden/speichern 539
 Dateien ziehen und ablegen 438
 Dateien, versteckte anzeigen..... 509
 Dateien/Verzeichnisse anzeigen ... 539
 Dateiende erreicht? 553
 Dateiende..... 551
 Dateiendung, suchen nach..... 508
 Dateierweiterungen v. Lazarus 96
 Dateifilter 539
 Dateigröße in Byte/Records 551
 Dateihandle Name zuweisen 550
 Dateihandle zuweisen/freigeben .. 556
 Dateihandle, ungenutzt./ungültig .. 302
 Datei-Maske 132, 144, 302
 Dateiname ändern 554
 Dateiname zuweisen 553
 Dateiname, Groß-/Kleinschreib... 302
 Dateinamen einstellen..... 190
 Dateinamen konvertieren 556
 Dateinamen, lange erlaubt?..... 302
 Dateinamenauswahl 422
 Dateinamenslänge, max. 302
 Datei-öffnen-Dialo schließen 540
 Datei-öffnen-Dialo, Autostart-
 Programmgruppe 545
 Datei-öffnen-Dialo, Callback..... 543
 Datei-öffnen-Dialo, Einträge im
 Startmenü zeigen..... 543
 Datei-öffnen-Dialo, Fenstergröße 539
 Datei-öffnen-Dialo, Netzwerkordner
 unterh. v. Domain anzeigen..... 542
 Datei-öffnen-Dialo, Papierkorb ... 545
 Datei-öffnen-Dialo, Programm-
 gruppe auswählen 544
 Datei-öffnen-Dialo, Schriften 545
 Datei-öffnen-Dialo,
 Startverzeichnis 539
 Datei-öffnen-Dialo, Statusbereich 542
 Datei-öffnen-Dialo,
 Systemsteuerung 545
 Datei-öffnen-Dialo, Titelzeile..... 539
 Datei-öffnen-Dialo,
 Unicode-Namen 541
 Datei-öffnen-Dialo,
 Verzeichnisse anzeigen..... 542
 Datei-öffnen-Dialo,
 voreingestellte Dateiendung ... 539
 Datei-öffnen-Menüeintrag 507
 Dateioperationen..... 553
 Dateiposition setzen 554
 Dateipositionszeiger auf
 Dateiende setzen 554
 Datei-Suchmaske..... 571
 Dateivergleich..... 180
 Dateivorschau..... 509
 Daten auf Oberfläche anzeigen ... 713
 Daten aus Datensatz anzeigen/
 verändern in GUI 713
 Daten e. Datensatzes auf
 Konsole ausgeben..... 694
 Daten einfügen, neue..... 700
 Daten im Speicher bearbeiten..... 699
 Daten im Speicher speichern 558
 Daten in Objekt erhalten 565
 Daten in String speichern 558
 Daten komprimieren 562
 Daten zw. Prozessen austausch. . 626
 Datenanzeige in Baumstruktur 422
 Datenbank, nativer Zugriff auf..... 715
 Datenbank-Benutzer, Paßwort 728
 Datenbank-Controls 516
 Datenbank-Controls, Eigensch. 713
 Datenbank-Engine..... 683
 Datenbank-Komponenten..... 360
 Datenbankname 728
 Datenbank-Navigationsmethoden 714
 Datenbankschnittstelle 682
 Datenbanktyp ändern 727
 Datenbanktypen..... 682
 Datenbank-Verbindungs-
 komponente, Proxy für..... 727
 Datenbank-Zugriff, Exception 236
 Datenbankzugriffsschichten v.
 Drittanbietern 682
 Datenbezogene Komponenten..... 353
 Datencursor 688
 Datendateien, Speicherort 347
 Daten-Desktop..... 733
 Daten-Desktop, Daten exportieren 740
 Daten-Desktop, DB-Abfragen 739
 Daten-Desktop, DB-Verbindung.... 739
 Daten-Desktop, Exportformate..... 740
 Daten-Desktop, Funktionen..... 734
 Datenmodul 244, 689
 Datensatz als letzten anfügen..... 699
 Datensatz ändern 698
 Datensatz anlegen, neuen..... 706
 Datensatz aus DB löschen ... 699, 701
 Datensatz in DB einfügen..... 701
 Datensatz öffnen 690, 692
 Datensatz schließen 690, 691f.
 Datensatz wird abgefragt.... 690, 691
 Datensatz wird editiert..... 690, 691
 Datensatz 684
 Datensatz, Änderung verwerfen... 699
 Datensatz, Bearbeit. abrechnen. 702
 Datensatz, best. suchen..... 697
 Datensatz, Cursor bewegen auf ... 705
 Datensatz, Editiermodus 699
 Datensatz, Einfügemodus 699
 Datensatz, erster/letzter,
 Cursor bewegen auf..... 692
 Datensatz, jeden zweiten löschen 701
 Datensatz, nächster bewegen auf 692
 Datensatz, navigieren auf nächst. 690
 Datensatz, neuen einfügen... 690, 691
 Datensatz, springen auf 692
 Datensatz, vorheriger beweg. a. . 692
 Datensatz-Änderungen abrechn... 704
 Datensätze filtern 691
 Datensätze im Dataset, Anzahl v. 690
 Datensätze im Speicher halten..... 691
 Datensätze sortieren 723
 Datensätze, im Dataset löschen... 701
 Datensatz-Status..... 690, 691
 Datentypen 33
 Datum und Zeit, Unit 40
 Datum-/Kalenderanzeige..... 420
 dBase, Indexdaten 721
 dBase, Zeichensatz 724
 dBase-Datei anlegen 725
 dBase-Datei öffnen..... 722f.
 dBase-Datei, Filter-Ausdruck..... 722
 dBase-Datei, Indizes 722
 dBase-Datei, UTF-8 723
 dBase-Dateien, Zugriff auf 715
 dBase-Tabelle-/Datei, Zugriff auf. 722
 DBF-Datei browsen 723
 DBF-Datei, Zugriff auf..... 723
 DBF-Dateien 682

DB-Komponenten, Basiskomp.....688
 DB-Zugriff, Klassen für686
 DCR-Dateien bearbeiten 524, 525
 Debuggen über ActiveSync.....312
 Debugger33, 154, 231
 Debugger, Unterschiede zwischen
 Lazarus und Delphi235
 Debuggerausgaben236
 Debugger-Einstellungen232
 Debuggersymbole entfernen155
 Debuggertyp232
 Debugging-Prozeß beenden236
 Delphi-Komponenten56, 523
 Delphi-Modus des Compilers527
 Delphi-Packages in Lazarus-
 Package konvertieren55
 Delphi-Packages524
 Dereferenzierungsoperator46
 Desktop, anderer161
 Desktop, Infos über406
 Desktophöhe/breite407
 Desktop-Instanz406
 Dialog anzeigen506
 Dialog m. benutzerdefinierter
 Titelseite402
 Dialog mit Warnung403f.
 Dialog, Farben in594
 Dialog, Ganzzahlen/Fließkomma-
 werte bearbeiten513
 Dialog, Position a. Bildschirm403
 Dialoge399
 Dialoge, Widgetset520
 Dialogfenster394, 401
 Dialog-Icon403
 Dialog-Kompon, Rückgabewerte..507
 Dialogtypen401
 Dienst293
 Dienst, Datenbankzugriff711
 Dienstanbieter/-benutzer653
 DirectorySeparator302
 DLL245
 DLLs in Speicher laden265
 DMA-Modus573
 Docking-Operation, Ereignis.....417
 Dokumentation von Pascal-Code ...54
 DOS-Zeichensatz724
 Double33
 dpr-Datei52
 Drag&Drop438
 Drag&Drop, Ausgangspunkt443
 Drag&Drop, Eigensch./Ereignisse.439
 Drag-Objekt, eigenes442
 Drag-Quelle in anderem Fenster...442
 DriveSeparator302
 Drop-down-Auswahlliste anbieten287
 Drucken223, 584
 Drucker auswählen422, 585
 Drucker, Datei senden an633
 Drucker-Eigenschaften586
 Druckjob senden585
 Druckparameter einstellen587

dwarf31
 Dynamic Link Libraries262

E

Edit699
 Edit-Elemente511
 Edit-Methode, Dialog anzeigen288
 Editor-Steuerelemente, Hierarchie462
 Eig. Zeichenop. ausführen374
 Eigenschaft auf ursprüngl. Wert
 zurücksetzen288
 Eigenschaft setzen43
 Eigenschaften einer Komponente
 anzeigen/ändern217
 Eigenschaften veröffentlichen.....355
 Eigenschaften, als published def. 520
 Eigenschaften, auf and. Komp.
 referenzieren288
 Eigenschaften, mehrauswahl-fäh.288
 Eigenschaften, untergeordnete...287
 Eigenschaften/Ereignisse zur
 Entwicklungszeit ändern119
 Eigenschafteneditor289
 Eigenschafteneditor, Grund-
 funktionalität290
 Eigenschaften-Mehrfachauswahl,
 Kriterien415
 Eigenschafts-Editor, Name286
 Eigenschafts-Editor, TFont288
 Eigentümerfenster, Position auf...389
 Ein-/Ausgabe, erweiterte, Unit.....40
 Ein-/Ausgabedatenströme353
 Ein-/Ausgabeop., Prioritätsgrad...639
 einfache Fenstertypen399
 Eingabeelemente421
 Eingabefeld für Zahlenwerte
 mit Schalter422
 Eingabefeld422
 Eingabefeld, schreibgeschütztes.463
 Eingabefeld, Text markieren463
 Eingabefeld, Textauswahl464
 Eingabemaske477
 Eingabewert, fragen nach405
 Einsprungsroutine257
 Einträge zw. Listboxen
 hin-/herbewegen439
 Eintrittspunkt, Name d. Biblioth....58
 Einzelfeld ändern714
 Einzelfeld im Datensatz688
 Einzeltabellen683
 Elemente e. graf. Oberfläche352
 Ellipse zeichnen591
 Erkennung einer Datei302
 Endpoint655
 EOF690
 Ereignisbehandlungsroutine .371, 374
 Ereignis-Objekte626
 Ereignisse55, 370
 Ereignisse, verknüpfte anzeigen...220
 Ereignissteuerung17
 Eventhandler (Definit.)371, 433

Eventhandler auswählen372
 Eventhandler erzeugen25
 Eventhandler schreiben373
 examples-Unterverzeichnis241
 Execute628
 ExecuteProcess636
 Explorer, Verzeichnis öffnen in.....633
 exports60, 258
 extdecl51
 Extended33
 external58

F

Farbauswahl422, 484
 Farben in Dialog594
 Farben592
 Farbkodierung593
 Farblistbox485
 Farbschalter513
 Farbuweisung593
 FBLib682, 716
 FCL55, 351, 358f.
 FCL, nicht-visuelle Komponenten.359
 Fehlerprüfung., After-Ereignisse...706
 Fehlerprüfung., Before-Ereignisse 706
 Fehlerprüfungen705
 Feld684
 Feld als benötigt kennzeichnen ...709
 Feld enthält falsche Datentypen...703
 Feld in akt. Datensatz690
 Feld in Klassendekl. e. Fensters ...357
 Feld löschen700
 Feld mit Index 0, Zugriff auf693
 Feld nur lesbar693
 Feld, alternat. Anzeige-/Edit-Text.710
 Feld, Alternativwert710
 Feld, Bearbeitungsmaske709
 Feld, Formatstring angeben709
 Feld, nur lesbar709
 Feld, persistentes, Eigensch. v710
 Feld, Standardwert einfügen709
 Feld, Wert muß stehen in693
 Feldanzeige, Breite709
 Feld-Anzeigeeigenschaften setzen707
 Feldattribute, erweiterte735
 Felddaten konvertieren694
 Feld-Definition691
 Feld-Eigenschaften mit
 Platzhalter belegen707
 Felder, berechnete693
 Felder, boolesche anzeigen714
 Felder, Ereignisse710
 Felder, erscheinen als Komp710
 Felder, persistente706
 Felder, prüfen auf leere696
 Felder, Zugriff auf689
 Feldinhalt, Anzeige steuern709
 Feldname357, 693, 709
 Feldtext anzeigen714
 Feldwert ändern, Ereignis710
 Feldwert auswählen714



- Feldwert löschen 699
 Feldwert setzen/suchen 688
 Feldwert, Anzeige eines steuern .. 710
 Feldwert, Bedingung prüfen 709
 Feldwert, Integer 696
 Feldwerte abfragen/setzen 695
 Feldwerte in Datensatz, Zugr. auf. 693
 Feldwerte, BCD 696
 Feldwerte, BLOBs 696
 Feldwerte, String 696
 Feldwerte, Typen 694
 Feldzugriff, Eigenschaften 694
 Fenster ableiten aus Projekt 245
 Fenster ableiten 357
 Fenster aktualisieren 571
 Fenster als Child zur Systemst. 267
 Fenster als Symbol in Taskleiste 394
 Fenster anzeigen 380
 Fenster auf Taskbar anzeigen 377
 Fenster aufzählen 408
 Fenster aus Speicher entfernen 384
 Fenster ausblenden 385
 Fenster b. Programmstart,
 welches erzeugen? 369
 Fenster drängt sich in Liste
 nach vorn 394
 Fenster dynamisch anlegen 395
 Fenster erhält Fokus 408
 Fenster für Programmstart 394
 Fenster hat sehr kleine
 Titelleiste (Toolwindow) 393
 Fenster holen 408
 Fenster ikonisieren 385
 Fenster in Desktopmitte anzeigen 389
 Fenster instantieren 393
 Fenster ist größenveränderbar 374
 Fenster kann kein Menü enthalten 393
 Fenster maximieren 394
 Fenster minimieren 394
 Fenster mit Memo u. Button 426
 Fenster nicht autom. v. IDE
 erzeugen lassen 396
 Fenster niemals regulär schließen 384
 Fenster öffnen, Position 389
 Fenster ohne Rahmen 393
 Fenster ohne Titelleiste 393
 Fenster schließen 374, 383
 Fenster sichtbar machen, Ereignis 378
 Fenster umschalten 386
 Fenster unsichtbar machen. 378, 385
 Fenster wird aktiviert, Ereignis 378
 Fenster wird deaktiviert,
 Ereignis 378
 Fenster wird geschlossen,
 Ereignis 378
 Fenster zerstören 374, 385
 Fenster 17
 Fenster, Dateien fallenlassen auf. 377
 Fenster, Erzeugung d. 364
 Fenster, Fokus setzen auf 408
 Fenster, Mehrfachinstanzen 396
 Fenster, Pos. u. Ursprungsgröße. 389
 Fenster, Rahmenbreite/-höhe 392
 Fenster, Ressourcendatei 379
 Fenster, Rollbalken anzeigen 377
 Fenster, Unit 378
 Fensterart 394
 Fenster-Basisklasse 521
 Fensterbeschriftung abfragen 405
 Fensterigentümer setzen 390
 Fenstergröße 387
 Fenstericons 392
 Fensterinstanz erzeugen, Ereignis 378
 Fensterinstanz wird zerstört 378
 Fensterinstanz, neue anlegen 390
 Fensterinstanzen freigeben 379
 Fensterliste des Taskbars 390
 Fensterliste führen 397
 Fensterliste 394, 408
 Fenster-Menü 377, 408
 Fenstermenü-Eintrag anlegen 408
 Fenstername 379
 Fensterposition auf Bildschirm 377
 Fensterpositionierung 387
 Fensterrahmen, Aussehen 390
 Fensterrahmen, Verhalten 390
 Fenstersystem fährt Anwendung
 herunter 376
 Fenstertitel ändern 395
 Fenstertitel 377, 390, 403
 Fenstertyp 377
 Fensterzugriff, Formvariable für .. 379
 Fensterzustand speichern 559
 Fensterzustand, anfänglicher 377
 FileMode 550
 FileName 719
 FileNameCaseSensitive 302
 Filter 691
 Filterauswahlbox 515
 FindFirst/FindNext/FindClose 567
 Firebird 682, 727
 Fläche füllen 591
 Flächen zeichnen 598
 Flat Files 683
 Flat-File-Format mit fester
 Datensatzlänge 721
 Font 377, 603
 Fonts, Eigenschaften 601
 Form 244
 Form-Instanz zerstören 398
 Form-Instanzen speichern 398
 FormShow 380
 FormStyle 377, 394
 Formular anlegen 120
 Formular, Layoutinformationen ... 142
 Formular-Designer, Mausclick 290
 Formulareditor 187
 Fortschrittsanzeige 420, 486
 FPC installieren 95
 FPC, mehrere parallel
 installieren 323
 fpc.cfg 139, 156
 FPCUnit 279
 fpdoc 19, 75, 136
 fpGUI 239, 247, 352
 Frame 17, 244, 473
 Frames schachteln 473
 Free Pascal Compiler 15
 Free Vision 252
 FreeLibrary 260
 Free-Pascal-Anwendung
 beginnen 245
 Free-Pascal-Bibliotheken in C 60
 fremdsprachige Zeichen 77
 FTP-Server, Pfadtrennzeichen 301
 Füllmuster, eigene 599
 Funktion exportieren 60
 Funktionen/Prozeduren
 exportieren 258
 Funktionsergebnis, Rückgabe 258
- ## G
- gdb 33, 231
 geometrische Einfassungen 420
 geometrische Figur anzeigen 476
 Gerätenamen in /dev 321
 Gitter anzeigen 422
 Gitter mit Datenbankanbindung ... 478
 Gitter-Elemente 477
 Gitterstruktur 714
 Gleitkommatypen 33
 Globales Anwendungsobjekt aller
 GUI-Applikationen 361
 Glyph 186, 456, 474
 Glyph, Bitmaps in 460
 Glyphs auf Schaltflächen 375
 Glyphs in Menüs 375
 GNU Assembler 33
 GNU Binutils 315
 GNU Debugger 31, 231
 GNU Linker 33
 GNU-Übersetzungsprojekt 72
 GOTO erlauben 151
 gprof 33
 Grafikformate 540, 614
 Grafikkomponenten 609
 Grid 714
 Grid, Eigenschaften 480
 Grid, fixierten Zellen 481
 Grid, Spalte automatisch anpassen 482
 Grid, Spalten 481
 Grid, Zeile in Eingabemodus 481
 Grid, Zeilenfarbe 481
 Groß-/Kleinschreibung 36, 207
 Groupbox 472
 Gruppe v. Steuerelem. anzeigen .. 420
 Gtk 55, 239, 247, 297, 324, 352
 Gtk 1, Theme 324
 196, 239
 GUI-Anwendung beginnen 245
 GUI-Bibliotheken 246
 GUI-Element, Funktionalitäten 415
 GUI-Programme, Aufbau 362

| | |
|---------------------------|-----|
| GUI-Toolkits | 324 |
| GUI-Widget-Sammlung | 351 |

H

| | |
|---------------------------------------|---------------|
| h2pas | 33, 60 |
| Haltepunkte | 235 |
| Handler in Idle-Schleife e. Anw. | 434 |
| Hardlinks | 323 |
| Hardwareschnittstelle | 573 |
| Hardware-Zugriffsprotokolle | 574 |
| Haupt-/Kontextmenü | 453 |
| Hauptfenster blockieren | 382 |
| Hauptfenster d. Anwendung | 248 |
| Hauptfenster erzeugen | 367 |
| Hauptfenster explizit anzeigen | 375 |
| Hauptfenster schließen | 379 |
| Hauptfenster, zentriert auf | 389 |
| Hauptmenü anlegen | 457 |
| Hauptmenü, neuer Eintrag | 408 |
| Hauptprogramm, Verwaltung d. | 364 |
| Hauptprogrammquelltext | 248 |
| Hauptprogramm Schleife | 249 |
| Hauptprojektdatei | 249 |
| Hauptprojekt-Symbol | 248 |
| Hauptquelldatei, Name | 19 |
| Hauptroutine | 257 |
| Hauptthread | 637 |
| Header mit Cookie | 276 |
| Headerdateien | 39 |
| HFS-Dateisystem | 331 |
| Hide | 385 |
| Hilfesystem aufrufen | 376 |
| Hilfetooltip aufpoppen | 375 |
| Hilfslinien | 189 |
| Hilfsprogramme | 96 |
| Hint anzeigen | 375 |
| Hint in Statusbar anzeigen | 500 |
| Hintergrundfarbe e. Controls | 452 |
| Hinweis in Popup-Fenster | 452 |
| Hinweisanzeigen | 376, 420, 472 |
| Histogramme anzeigen | 420 |
| Hosting v. Services | 653 |
| Hostname d. DB | 728 |
| HTML-Datei als Template | 277 |
| HTML-Seite anzeigen | 275 |
| HTTP-Applikationsserver | 661 |
| HTTP-Header in HTTP-Antworten | 274 |
| HTTP-Servermodul | 653 |
| HTTP-Versand | 654 |
| I/O-Überprüfung ein-/aus | 551 |

I

| | |
|--------------------------------------|---------|
| i18n | 72, 174 |
| Icon in Titelzeile e. Fensters | 392 |
| Icon laden | 621 |
| Icon | 377 |
| Icon, Bestandteile | 620 |
| Icon, Ressourcen-ID | 270 |
| Icon, Systemschrift | 407 |
| Icon-Definitionsdatei | 265 |
| Icon-Format | 620 |

| | |
|---|---------------|
| Icons f. Speedbuttons | 475 |
| IDE erweitern | 244 |
| IDE fährt nicht hoch | 242 |
| IDE findet Package nicht | 284 |
| IDE für and. Widgetset kompil. | 237 |
| IDE in and. Verzeichnis kompil. | 242 |
| IDE neu kompilieren | 237, 241, 281 |
| IDE, Build-Optionen | 239 |
| IDE, Hauptfenster | 119 |
| IDE, minimale bauen | 242 |
| IDE, Sicherungsdateien | 190 |
| IDE, Tastaturbelegung | 197 |
| IDE-Fenster, Größe | 187 |
| IDE-Neustart | 241 |
| IDE-Schnittstelle | 241 |
| IDE-Suchpfade | 139 |
| ifdef | 27, 298 |
| Image löschen | 623 |
| Implementationsdatei, Webservice | 656 |
| Implementierungs-Abschnitt | 39 |
| Include-Datei f. Linux | 299 |
| Include-Datei f. Win32 | 299 |
| Include-Dateien | 27, 50, 298 |
| Includesuchpfad | 299 |
| Index | 684 |
| Index, Feldname des | 722 |
| Index, Name des | 722 |
| Indexdefinitionen e. Tabelle | 686 |
| Indy | 654 |
| Indy-basierter HTTP-/TCP-Server | 653 |
| ini-Datei | 348 |
| In-memory-Daten | 717 |
| InnoSetup | 350 |
| In-Process-Transportprotokoll | 654 |
| Input-/Output-Dateideskriptoren | 629 |
| InputQuery | 405 |
| Insert | 699, 700 |
| Installationsprogramm | 350 |
| Integer, 64-Bit-System | 343 |
| Integer, Größe | 267 |
| Intel-Syntax, Assembler | 152 |
| Interbase | 682 |
| Interface-Abschnitt | 39 |
| Interfacedatei, Webservice | 656 |
| Interface-Definition | 45 |
| Interface-Units | 530 |
| Interrupt | 258 |
| Introspection | 355 |
| IP-Adresse | 645 |

J

| | |
|---------------|-----|
| JPEG | 619 |
| JsonRPC | 653 |

K

| | |
|----------------------------|----------|
| Kalender | 714 |
| KDE, Fenstericons | 324, 392 |
| Kernelfunktionen | 31 |
| Kernel-Gerätetreiber | 575 |
| Kernelobjekte | 626 |

| | |
|---|-------------|
| Kinderkomponenten | 521 |
| Klasse ableiten | 44 |
| Klasse in Objektspektor bearb. | 353 |
| Klassen | 35 |
| Klassenbibliotheken | 351 |
| Klassendeklaration, Abschnitte | 42 |
| Klasseninstanzen, Namen | 356 |
| Klassenmethode deklarieren | 44 |
| Klassenvervollständigung | 519 |
| Klassenvervollständigung | 226 |
| Kommandozeile verarbeiten | 376 |
| Kommandozeilen-Opt., Nummer | 375 |
| Kommandozeilen-Optionen, Groß-/Kleinschreibung | 375 |
| Kommandozeilen-Optionen, indexierter Zugriff auf | 375 |
| Kommandozeilenschalter, Zeichen für | 375 |
| Kommentarmarkierung | 124 |
| Kommunikationsprotok., Listener f. | 662 |
| Kompilieren, Tastenkombination .. | 157 |
| komplizierte Unit | 250 |
| Kompilieren, bedingte | 17, 50, 298 |
| Komponente in IDE installieren | 61 |
| Komponente, abgeleitete anlegen .. | 245 |
| Komponente, eigenes Handle | 521 |
| Komponente, installierte suchen .. | 137 |
| Komponenten installieren | 285 |
| Komponenten lesen/speichern | 520 |
| Komponenten ohne Parent | 521 |
| Komponenten | 55 |
| Komponenten, Eigenschaft Name | 520 |
| Komponenten, einfache | 521 |
| Komponenten, plattformabh. | 522 |
| Komponenten, plattformunabh. | 520 |
| Komponentenauswahl, mehrere | 450 |
| Komponentenbeispiele | 451 |
| Komponenteneditor, Grundklasse .. | 290 |
| Komponenteneigensch. ü. Fenster- sessions hinweg speichern | 377 |
| Komponentenname | 356 |
| Komponentenpal. Icon hinzufügen .. | 537 |
| Komponentenpal., Kontextmenü .. | 450 |
| Komponentenregistrierung | 285 |
| Komprimierungs-Stream | 561 |
| Konfigurationsdat., Name ermitt. | 343 |
| Konfigurationsdat., Speicherort .. | 344 |
| Konfigurationsdatei, global/lokal .. | 343 |
| Konfigurationsdateien | 343 |
| Konfigurationsdaten schreiben | 539 |
| Konsole automatisch öffnen | 253 |
| Konsole | 555 |
| Konsolenanwendung | 245 |
| Konsolenfenster-Titel | 627 |
| Konsolen-Pipelining | 253 |
| Konsolenprogramm | 252, 375 |
| Konstanten-Array | 49 |
| Konstanten-Deklarationsblock | 39 |
| Konstruktorname | 152 |



| | |
|------------------------------|-----|
| Kontextmenü | 454 |
| Kreis/Ellipse zeichnen | 607 |
| Kreisbogen zeichnen | 591 |

L

| | |
|--|--------------|
| Label, &-Zeichen | 461 |
| Label, Eigenschaften | 462 |
| Laufwerksbuchstaben | 57 |
| Laufzeitbibliothek | 351 |
| Layout, Werte | 430 |
| Layout-Objekte | 423 |
| Lazarus Daten-Desktop | 687 |
| Lazarus kompilieren | 196, 241 |
| Lazarus-Anwendung in Emulator | 312 |
| Lazarus-Anwendung, Ebenen | 529 |
| Lazarus-Formular | 217, 354 |
| Lazarus-Package, Definition | 54 |
| Lazarus-Projekt | 363 |
| Lazarus-Verzeichnis, aktuelles | 53 |
| lazbuild | 19, 71 |
| lazdaemon | 293 |
| lazDBExport | 742 |
| LazReport | 748 |
| LCL einrichten | 367 |
| LCL neu kompilieren | 241 |
| LCL | 15, 247, 351 |
| LCL, Basisklassen | 361, 520 |
| LCL-, grafische Oberflächen | 297 |
| LCL, Klassensysteme | 519 |
| LCL, portieren auf | 529 |
| LCL-Interface zu Widgetset | 519 |
| LCL-Interface | 55 |
| LCL-Interface, abstr. Basisklasse | 522 |
| LCL-Klassen, public/published Eigenschaften | 522 |
| LCL-Komponenten | 449 |
| LCL-Modus | 317 |
| LCL-Schnittstelle | 529 |
| LCL-Widgetset | 150 |
| Lebensdauer v. Objekten | 676 |
| LED-API | 310 |
| Lesezeichen | 697 |
| LFNSupport | 302 |
| library | 40, 258 |
| LineEnding | 302, 556 |
| Linie zeichnen | 591, 604 |
| Linie, Hintergrund | 595 |
| Linien verbinden | 596 |
| Linienendpunkte | 596 |
| Linken | 154 |
| Linken, dynamisch | 261 |
| Linken, ext. Objektdat./Biblioth. | 258 |
| Linker | 31 |
| Links auflösen | 323 |
| Linux, Verzeichnishierarchie | 322 |
| Linux-Versionen | 98 |
| Listbox umsortieren | 440, 469 |
| Listbox | 714 |
| Listbox, Einträge einfügen | 447 |
| Listbox, Höhe | 469 |
| Listbox, Mehrfachauswahl | 469 |

| | |
|--|----------|
| Liste, Auswahl aus | 468 |
| Listefeld | 469, 495 |
| Listen-Klasse | 354 |
| Listenstrukturen anzeigen | 422 |
| Listview, Einträge | 496 |
| Listview, neuen Eintrag erzeugen | 497 |
| Listview, Spaltendefinitionen | 496 |
| Listview, Verzeichnisscan | 498 |
| LoadFromFile | 548 |
| Login-Namen ermitteln | 300 |
| Longbool | 34 |
| LongInt, Größe | 267 |
| Low-Level-Routinen einer Plattform, Zugriff auf | 300 |

M

| | |
|---|---------|
| MacOS X, Dateinamenvergleich | 331 |
| MacOS X, Dateisystem | 331 |
| MacOS X, Framework linken | 329 |
| MacOS X, Unicode | 331 |
| MacOS X, Verzeichnisse | 332 |
| MacOS X-API | 328 |
| MacOS, Framework in-/deinstall. | 264 |
| Makedatei | 18, 157 |
| Makefile.fpc | 157 |
| make-Werkzeug, Pfad | 184 |
| Makros | 51, 66 |
| Master-Detail-Bez. in SQLDB | 733 |
| Mathematische Routinen, Unit | 40 |
| Mausbewegung, reagieren auf | 452 |
| Mausklick, Ereignis | 417 |
| Maus-Rechts-Doppelklick im Formular-Designer | 290 |
| Maustaste wird gedrückt | 453 |
| Maustaste wird gelöst | 453 |
| Maustaste, Ereignis | 417 |
| Maustaste, linke, auf El. klicken | 453 |
| Mauszeiger ist über Element | 453 |
| Mauszeiger passiert Elementrand | 453 |
| Mauszeiger-Aussehen | 452 |
| MDI-Eltern-/Kindfenster | 394 |
| Mehrere Dateien zu einer verbinden | 50 |
| Mehrfachvererbung | 45, 327 |
| Meldungsdialoge anlegen | 229 |
| Memofeld, Zeile/Absatz | 464 |
| Menü anpassen | 291 |
| Menü, dynamisches | 458 |
| Menü, Systemschrift | 407 |
| Menü, Tastenkürzel | 455 |
| Menübeschriftung umbenennen | 457 |
| Menüeditor | 456 |
| Menüeintrag eingeschaltet? | 434 |
| Menüeintrag mit Betätigt-Zeichen | 456 |
| Menüeintrag | 454 |
| Menüeintrag, Bild vor | 455 |
| Menüeinträge als Radiobuttons | 456 |
| Menüfunktion schnell anwählen | 474 |
| Menüfunktion, Zugriff auf | 501 |
| Menüs | 453 |
| Menüs, Widgetset | 520 |

| | |
|--|----------|
| Menüstruktur mit Menüeinträgen | 420 |
| Message Composer | 229 |
| Message Loop | 363 |
| MessageDlg | 401 |
| Message-Methoden, Portierung | 533 |
| Meta-Datenbank | 735 |
| Methode autom. extrahieren | 28 |
| Methode überschreiben | 43 |
| Methoden als static deklarieren | 152 |
| Methoden exportieren | 259 |
| Methoden | 43 |
| Methoden, leere entfernen | 28 |
| Methodenaufrufe | 258 |
| MIME-Type e. Website | 275 |
| minimale IDE bauen | 242 |
| mobile Plattformen, Unterschiede | 309 |
| modale Fenster | 380 |
| Monitore, Anzahl | 407 |
| Multitasking, präemptives | 625 |
| multithreaded Anwendungen | 637 |
| Mutex-Objekte | 626 |
| MySQL | 682, 727 |

N

| | |
|--|----------|
| Nachlade-Optimierung | 153 |
| Nachricht anzeigen | 400 |
| Nachricht über Anwender-Aktion | 376 |
| Nachrichten an LCL senden | 56 |
| Nachrichten, Analyse d. | 370 |
| Nachrichtenbehandlg.-Methoden | 530 |
| Nachrichtenfenster | 119, 134 |
| Nachrichtenfenster, Fokus auf | 186 |
| Nachrichtenschleife ausführen | 369 |
| Nachrichtenverarbeitung | 362 |
| Nachschau-Tabelle | 684 |
| Nachschlagelisten | 717 |
| Namensvergabe | 41 |
| Namespaces | 32 |
| Nativer Zugriff auf DB | 715 |
| Navigier. in Datensätzen, Meth. | 688 |
| Navigieren in DB | 690f. |
| ncurses | 252 |
| Neues Fenster zu Projekt hinzufügen | 247 |
| NexusDB | 716 |
| Nicht größenveränderbare Fenster | 394 |
| Nicht-ASCII-Zeichen | 77 |
| Nichtmodales Fenster anzeigen | 380 |
| Nicht-visuelle Klassen, Aufteilung | 352 |
| Nicht-visuelle Komponenten | 55 |
| NoGUI | 239 |
| Normales Fenster | 394 |
| Nullsoft | 350 |
| Nullterminierter String | 306 |

O

| | |
|--|-----|
| Oberflächenbiblioth., gekapselte | 352 |
| Objective-C-API, MacOS X | 329 |
| Objective-C-Quelle in Pascal | 337 |
| Object-Pascal-Typ | 33 |

Object-Pascal-Übersetzung e.
 Bibliotheks-Interfaces 33
 Objekt Pascal, Syntaxausnahmen 151
 Objekt über Steuerelement ziehen 417
 Objekte auf Vorrat anlegen 676
 Objekte ziehen u. ablegen 439
 Objekte, cut&paste 412
 Objectfactory für Implementation v.
 Webservice 676
 Objektinspektor 55, 119, 189, 217
 Objektinspektor, Einstellungen 189
 Objektpooling 676
 ODBC-System 686, 727
 ODBC-Treiber 682, 715
 Öffnen-/Schließen-Dialog,
 Eigenschaften 509
 OK-Button anzeigen 400, 459
 OK-Button, Verarbeitung 382
 OLE-Objekte 523
 Optionenschaltfläche 421
 Oracle 682, 727
 Ordner-Inhalt in Explorer darstellen 633
 OS/2 297
 Output 555
 override 43
 Owner-owned Hierarchy 412

P

Package beginnen 282
 Package Dateien hinzufügen 57
 Package der IDE hinzufügen 281
 Package dynamisch laden 535
 Package generieren 245
 Package Graph 57
 Package in Projekt aufnehmen 282
 Package installieren 60, 162
 Package kompilieren 71, 164
 Package neu kompilieren 72, 241
 Package ohne Lazarus kompilieren 71
 Package suchen 68
 Package 143, 162, 280
 Package, Abhängigkeit
 hinzufügen 169
 Package, Abhängigkeiten 71
 Package, aufnehmen in 163
 Package, Bestandteile 282
 Package, Compilereinstellungen
 vererben 70
 Package, dynamisches 280
 Package, externes 453
 Package, in Projektinspektor als
 Abhängigkeit hinzufügen 282
 Package, Liste d. geöffneten 284
 Package, Sprachen 174
 Package, Suchpfade 283
 Package, Suchpfade/Einstellungen
 vererben 70
 Package-Editor 162, 281
 Package-Einstell. u. -Suchpfade 54
 Package-Konzept 17
 Package-Name 68

Package-Registrierung 241
 Packages, welche in IDE? 69
 Package-Status 174
 Package-Typ 172
 Package-Unitpfade d. zum Quelltext
 Pfad e. Projekts hinzuf. 141
 Package-Version 172
 Pakete bauen, neue 245
 Paketmanager v. Linux 98
 Palettenseite 167
 Panel 473
 Panel, andere Steuerelemente
 ablegen auf 473
 Panel, Rahmen 473
 Panel-Editor 500
 Paradox-Dateien, Zugriff auf 715
 Parallele Protokolle 573
 Parallelinstallation, FPC 323
 Parallel-Port 574
 Parameterhinweise 21
 Parameterliste erzeugen 127
 Pascal-Bezeichner 411
 PasCocoa 333
 passive Steuerelemente 420
 Paßwort-Abfrage 406
 PathDelim 301
 PathSeparator 302
 PChar 34
 PDA 319
 Peephole-Optimierungen 153
 Performanceanalyse 155
 Persistenz 565
 Pfade, Makros für 150
 Pfadtrenner 301
 Pfeil zeichnen 513
 Pins d. parallelen Steckers 576
 Pinsel 598
 Pinsel, am Canvas ausgerichtet,
 gebitmapter 600
 Pinsel, benutzerdefinierter 599
 Pinselmuster 598, 600
 Pixel setzen/lesen 591, 604
 Pixelbildformate, Basisklasse 614
 PixelsPerInch 377
 plattformabhängige Quelltextteile
 auslagern 298
 Plattformunabhängigkeit 137
 Platzhalter umwandeln 130
 Plugin 280
 PocketPC 309
 po-Dateien automatisch laden 73
 POEdit 73
 Pointer 34
 Pointer-Arrays 48
 Polygon, Koordinaten übergeben. 605
 poolbare Objekte 676
 Popup-Fenster mit Hint 452
 Popup-Menü anzeigen, Ereignis... 417
 Popup-Menü öffnen 458
 Popup-Menüeinträge editieren ... 291
 Port, Zugriffsrecht auf 576

Position 377
 POSIX-Thread-Bibliothek 638
 PostgreSQL 682, 727
 Powtils 276
 Primärbildschirm 388
 Printer4Lazarus 584, 749
 printers4lazide 121, 223, 584
 Print-Event beginnen/beenden 585
 Prior 692
 private-Abschnitt 42
 Profiling 33
 program 257
 Programm, anderes nachladen ... 632
 Programm, Optionen übergeben.. 627
 Programme ohne Anwender-
 Interaktion 293
 Programm-Formulare, Einstell. 187
 Programmstart beeinflussen 627
 Programm-Startverzeichnis 627
 Programmumgebung 406
 Projekt aus IDE starten 186
 Projekt beginnen 245
 Projekt Datei hinzufügen 244
 Projekt neues Form hinzufügen ... 146
 Projekt veröffentlichten 144
 Projekt 19, 52, 141
 Projektarten 243
 Projekteinstellungen 149
 Projekt-Hauptdatei 143
 Projekt-Hauptunit 146
 Projekt-Kopie 144
 Projekt-Metainformationen 19
 Projektspezifische Einstellungen ... 141
 Property-Editoren 286
 protected-Abschnitt 42
 Prozedur, definiert wo 222
 Prozeduraufrufvervollständigung. 229
 Prozeduren, geschachtelte 209
 Prozedurrumpf 220
 Prozeß beenden 628
 Prozeß starten 626
 Prozeß, synchrone Aufrufe 632
 Prozeß-Exitcode 302
 Prozeß-Objekte 626
 Prozeßwechsel 625
 Prüfsumme 18
 public/published Eigenschaften... 522
 public-Abschnitt 42
 published Eigenschaften in
 Datenstrom speichern 354
 published-Abschnitt 42
 Punkt 589

Q

Qt exportieren 325
 Qt 55, 239, 247, 297, 324, 352
 Qt-4-Zugriff 327
 Quelldatei in UTF-8 umwandeln 77
 Quellen, L. installieren aus 98
 Quelltext formatieren 122
 Quelltext markieren 123



Quelltext sortieren.....124
 Quelltext vervollständigen226
 Quelltext, navigieren in136
 Quelltext, springen in132
 Quelltexteditor19, 55, 119, 222
 Quelltexteditor, Einstellungen192
 Quelltextschablonen, Liste der26
 Quelltextteile in Prozedur auslag.126
 Quelltext-Unit142
 Quelltextvervollständigung23
 Quellverzeichnis aussuchen568
 Queries684
 Queries, Parameter686
 Query-Editor743

R

Race Condition639, 641
 RAD-Fähigkeiten15
 Radiogroup472, 714
 Rahmeneigenschaften390, 418
 rc-Ressourcendatei, Aufbau565
 RDBMS, Verwaltungsroutinen685
 ReadLn554
 ReadOnly693, 709
 Recht-/Vieleck zeichnen591, 606
 Record35, 684
 Recordcount690
 Record-Definition36
 Refactoring-Tools28
 Refresh692
 Register258
 RegisterComponents285
 Registersicherung258
 Registrierdatenbank, lesen aus349
 Reguläre Ausdrücke129, 170
 RemObjects Software716
 Remote-Debugging232
 Report-Designer748, 753
 Ressourcenstring34
 Ressourcen lesen564
 Ressourcen, interne/externe565
 Ressourcencompiler526, 564
 Ressourcendatei142, 248, 346
 Ressourcendateien, MacOS X347
 Ressourcen-Stream565
 Ressourcenstrings definieren72
 Ressourcenstrings übersetzen74, 149
 Ressourcenstrings54, 178
 Ressourcen-Verzeichnis ermitteln348
 Rewrite554
 RGB-Wert zuweisen604
 Rollbalken, Einstellungen377, 472
 RPC652
 rst-Datei54, 72
 RTL, Basisklassen353
 Rückgabewerte v. Schaltflächen402
 Run Time Library31
 Run369
 rundll.exe265
 Run-Time Type Information/RTTI355

S

SafeCall258
 Sammlungen ident. strukt. Einträge354
 Schalter erhält Fokus461
 Schalter mit Beschriftung u. Bild.420
 Schalter mit Text459
 Schalter, fensterfähiger461
 Schalter, Standardaktionen zuw.460
 Schaltfläche anzeigen402
 Schaltfläche f. Fenster
 maximieren392
 Schaltfläche f. Hilfefunktion
 aufrufen392
 Schaltfläche f. Menü für
 Fensteroperationen392
 Schaltfläche für Fenster schließen392
 Schaltfläche mit Pfeilen504
 Schaltfläche OK, Abbruch,
 Schließen, Hilfe515
 Schaltfläche, flach503
 Schaltfläche, Rahmenlinien503
 Schaltflächen-Namen402
 Schaltflächen-Werte402
 Scheduler625
 Schieberegler486
 Schnittstelle, parallele574
 Schrift auf Kind-Elementen377
 Schrift e. Textelements452
 Schriftauswahl422
 Schriften installierte, Namen407
 Screenshot des Monitors615
 Scrollbar472
 Scrollbox477, 610
 SELECT730
 Self-Parameter152, 258, 325
 Semaphoren-Objekte626
 Serialisierungsformate666
 Serielle Protokolle573
 serielle Schnittstelle579
 serieller Adapter, Belegung580
 Serviceanbieter implementieren655
 Serviceerweiterungen,
 Registrierdatenbank f.680
 Service-Interface definieren656
 Service-Interface, Webservice656
 Shared Object245, 262
 ShellExecute, Fenster anzeigen635
 Shellobjekte initialisieren546
 Shell-Verzeichnisdialog541
 Shortbool34
 Shortstring34, 152
 Show381
 Sitzungsinform. speichern53, 343
 Smartlinken31, 153
 Smartphone319
 Snapshot82
 SOAP653
 SOAP-Serialisierungsformat662
 Spaltentitel anzeigen505
 Speedbutton474
 Speicher allokiieren46
 Speicher als lokales Array306

Speicher freigeben46
 Speicher, Daten im715ff.
 Speicherbereich kopieren47
 Speicherbereich vergrößern46
 Speicherlecks155
 Speicherort e. Konfigurationsdatei
 angeben377
 Speicherplatz allokiieren/
 freigeben306
 Spezialfenster399
 Splitter431
 Sprachen, mehrere149
 SQ-DB, Master-Detail-
 Beziehungen733
 SQL685
 SQL, Daten aktualisieren731
 SQL, Delete-/Update-
 Anweisungen732
 SQL-Abfragen schreiben744
 SQL-Datenbank, verbinden m.728
 SQL-Datenbank, Verbindung m.726
 SQL-Datenbanken726
 SQLDB726
 SQLDB, Parameter733
 SQL-DB, Schlüsselfelder732
 SQL-DB, SQL-Anfragen ausführen726
 SQL-DB, Transaktionen steuern726
 SQL-DB, Zugriff a. Datensätze731
 SQLDB-Verbindung, Controls für727
 Sqlite3682, 727
 Stackbereinigung258
 Standardausgabe253
 Standarddatenbankkomponente682
 Standard-Ein-/Ausgaberroutinen40, 252
 Standardeingabe253
 Standard-Hilfedatei d. Anwendung375
 Standard-In-/Output-Gerät555
 Standard-Input-Dateihandle302
 Standard-Output-Dateihandle302
 Standard-Units von Free Pascal40
 start Lazarus241
 Statusaktualisierung v. Control433
 Statusanzeigen420
 Statusbar. Position verschieben499
 StdCall58, 258
 Stdcall-Modifizierer, auf versch.
 Systemen58
 StdErr302, 555
 stdin-/stdout-/stderr-Bezeichner b.
 Prgstart schließen293
 StdIn/OutputHandle302
 Steuerelement an Rand d. Elternobj.
 ankleben426
 Steuerelement auf Fenster, Zugriff411
 Steuerelement bekommt/verliert
 Fokus, Event418
 Steuerelement ein-/ausgeschaltet434
 Steuerelement erhält Fokus408
 Steuerelement hat Fokus377
 Steuerelement hinzufügen409
 Steuerelement löschen415

Steuerelement mit Bild 420
 Steuerelement verschieben 412
 Steuerelement, aktives ändern 380
 Steuerelement, Drag&Drop/
 Docking 416
 Steuerelement, Eigenschaften mehre-
 rer gleichzeitig setzen 414
 Steuerelement, Eigentümer 412
 Steuerelement, Farbe 416
 Steuerelement, Größenänd. mit
 Ankern 427
 Steuerelement, manuelle
 Größenänderung 425
 Steuerelement, mehrere Instanzen 414
 Steuerelement, Mindest- und
 Maximalgröße 416, 423
 Steuerelement, Schriftart 416
 Steuerelement, springen auf
 mit Tab-Taste 418
 Steuerelement, Statusaktualisierung
 prüfen 433
 Steuerelement, veröffentlichte
 Eigenschaften 423
 Steuerelement-Ausrichtung 423
 Steuerelement-
 Ausrichtungsmöglichkeiten 426
 Steuerelemente in Zeilen/
 Spalten organisieren 430
 Steuerelemente opt. gruppieren 476
 Steuerelemente v. Windows XP .. 248
 Steuerelemente, Abstammung
 der visuellen 452
 Steuerelemente, Ausrichtung
 verschiedener 426
 Steuerelemente, gitterartig
 anordnen 430
 Steuerelemente, vis., gemein.
 Eigenschaften 452
 Steuerelemente, vis., gemeinsame
 Ereignisse 453
 Steuerelementg, Größe rel. ändern 431
 Steuerelement-Größe 423
 Steuerelement-Größenanpassung 423
 Steuerelementhöhe/-breite 416
 Steuerelement-Name 410
 Steuerelement-Position 423
 Steuerelement-Randabstand 423
 Stored Procedures 685
 Stream in Speicher 560
 Stream in Zeichenkette umw. 561
 Stream, Daten lesen aus 559
 Stream, Daten schreiben in 559
 Stream, Eigenschaften 558
 Stream, Position setzen 559
 Streaming-System, Unit 40
 Streams 557
 String holen 287
 String konvertieren 78
 Stringgrid 482
 Stringkodierung ermitteln 78
 Stringkonstanten-Definitionen 39

Stringliste aufbauen 569
 Stringliste freigeben 398
 Strings nach UTF-8 wandeln 77
 Strings übersetzen 74
 Stringtypen 34
 Strom über Schnittstelle 575
 Subclassing 337
 Suchen-und-Ersetzen-Schablonen 131
 Suchfunktion 571
 Suchpfad f. Pascal-Quelltextdat. . 150
 Suchpfad für Bibliotheken 150
 Suchpfade, kontextbezogene 54
 Suchtext 130
 SVN-Repository 82
 Symbol in Titelleiste 377
 Symbole/Bilder/Trenner 521
 Symbolische Links 322
 Symbolleiste, Icon 295
 Synapse Sockets Library 645
 Synapse 654
 Synapse-basierter TCP-Server ... 653
 Synaser 580
 Synchronisationsobjekte 626
 SynEdit 241, 518
 SynEdit-Komponente 19
 Syntaxhervorhebung 27, 201
 Syntaxmodus 151
 Systemdatei 572
 Systemdialoge, Ergebniskonst. 383
 Systemfarben 593
 Systemkodierung ermitteln 78
 Systemschrift f. Hinweistexte 407
 Systemsteuerung, Fenster als Kind 267
 Systemsteuerungsapp. deinstall. 272
 Systemsteuerungsapplet 265
 Systemsteuerungsfkt. exportieren 270
 Systemsteuerungs-Start 267
 Systemverzeichnis, Windows 265
 System-Zeichensatz 257
 SysToUTF8 541

T

TabletPC 309
 Tabulator in Leerzeichen umw. 125
 Tabulatorreihenfolge e. Fensters. 476
 TApplication 361, 363
 TApplication, Eigenschaften 375
 TApplication, Ereignisse 376
 TApplication, Methoden 364, 376
 TargetOS 299
 Taskleiste d. Desktops, Icon auf. . 477
 Tastaturkürzel betätigen, Ereignis 378
 Tastaturshortcut/Hotkey ausführen 291
 Taste drücken/loslassen, Event .. 418
 Tastendrücke an Fenster senden. 377
 Tastenereignis, reagieren auf 452
 Tastenkombination zuweisen 124
 Tastenkürzel auf Win.-Desktop ... 305
 Tastenkürzel 434
 TBevel 420
 TBitBtn 420, 459

TBrush 598
 TButton 420, 459
 TCanvas, Zeichenmethoden 592
 TCheckbox, Eigenschaften 467
 TComboBox, Eigenschaften 471
 TComponent, Funktionen 520
 TComponent, Nachfahren 409
 TComponent, Streamingfähigkeit 520
 TComponentEditor 290
 TControl 352, 361, 415, 452
 TControl, Events 417
 TControl, Haupteigenschaften 416
 TCP/IP 645
 TCP-Bibliothek 645
 TCP-Versand 654
 TCustomEditButton, abgeleitet
 Steuerelemente 512
 TDatamodule 689
 TDatamodule-Instanzen, Anzahl .. 407
 TDataSet 686
 TDataSet, Ableitungen v. 715
 TDataSet, Eigenschaften v. 690
 TDataSet, Ereignisse f. Fehlerprüf. 706
 TDataSource 689, 713
 TDateTime-Feld anzeigen 714
 TDBF, Eigenschaften v. 722
 TEdit, Eigenschaften 463
 TEdit, Ereignisse 464
 TEditButton, Eigenschaften 511
 Telefon-API 310
 Terminal, Kommunikation mit 252
 Testdaten erzeugen 740
 Text auf Steuerelement 434
 Text ohne Hervorhebungen 420
 Text zeichnen 591
 Text, statischer 475
 Text, Systemschrift 407
 Textausgabe 602, 609
 Textdatei, Zugriff auf 550
 Textdateien, Standardvariablen ... 555
 Textdateien, Zeilenumbrüche 556
 Texteingabefeld, ein-/mehrzeilig .. 462
 Textfile 549
 Textlabel 462
 Text-Methoden 609
 Textmodus-Anwenderschnittst. . 252
 TField, Eigenschaften v. 693
 TField, Standard-Ableitungen 696
 TField-Instanzen 691
 TFileStream 353, 557
 TFileStream, Datei öffnen 558
 TFont, Eigenschaften 604
 TFont, Eigenschafts-Editor 288
 TFontDialog 422
 TForm 361
 TForm, Eigenschaften 377, 387
 TForm, Ereignisse 378
 TForm, Instanzen von 378
 TForm, Methoden 379
 TForm.Position, Konstanten 389
 TForm-Abkömmling 378



| | |
|---|----------|
| TForm-Instanz, Kinder | 356 |
| TForm-Instanzen, Anzahl..... | 407 |
| TGraphic, Eigenschaften | 614 |
| Thread anhalten..... | 639 |
| Thread..... | 625, 637 |
| Thread-Manager | 638 |
| Thread-Objekte..... | 626 |
| Thread-Priorität | 639 |
| Thread-Synchronisationsmech..... | 640 |
| TImage..... | 420, 476 |
| TImage, Bestandteile | 613 |
| TImage, Methoden/Eigenschaften | 610 |
| TImage/TImageList | 609 |
| Titelleistensymbole anzeigen..... | 377 |
| Titelzeile e. Fensters, Symbole | 392 |
| TLabel | 420 |
| TList | 354 |
| TListbox, Eigenschaften | 469 |
| TListItem, Eigenschaften | 496 |
| TListView | 422 |
| TListView, Eigenschaften..... | 496 |
| TMainMenu | 454 |
| TMemDataset | 715, 717 |
| TMemo | 422, 462 |
| TMemoryStream..... | 353, 560 |
| TMenu..... | 420 |
| TMySQL4xConnection..... | 727 |
| TNotifyEvent..... | 371 |
| TODBCConnection..... | 727 |
| Toolbar..... | 501 |
| Toolbar, Bilder anzeigen..... | 503 |
| Toolbar, dynamische Buttons auf | 503 |
| Toolbar, Eigenschaften d. Schaltflächen auf..... | 502 |
| Toolbar, Eigenschaften..... | 503 |
| Toolchain | 315 |
| Toolkit-Versionen..... | 324 |
| Tools API..... | 244 |
| Tooltip teilen | 501 |
| Toolwindow | 393 |
| TOpenDialog | 540 |
| TPagecontrol | 422 |
| TPaintbox..... | 420 |
| TPairSplitter..... | 431 |
| TParadox | 715 |
| TParameter..... | 686 |
| TPen..... | 594 |
| TPersistent | 353 |
| TPopupMenu..... | 420, 454 |
| TPopupNotifier | 420 |
| TPrintDialog, Eigenschaften..... | 587 |
| TPrinter, Methoden/Eigenschaften | 586 |
| TProcess, Eigenschaften..... | 627 |
| TProcess, Methoden..... | 630 |
| TPropertyAttributes, Felder | 288 |
| TPropertyEditor, Methoden | 287 |
| TRadioButton..... | 421, 468 |
| TRadiogroup | 421 |
| TRadioGroup, Eigenschaften | 473 |
| Transparenzdefinition..... | 615 |
| Transportprotokoll..... | 653 |

| | |
|--|---------------|
| Treeview | 495 |
| Trennzeichen f. Laufwerk u. Pfad. | 302 |
| Trennzeichen f. Listeneinträge | 302 |
| Trennzeichen f. Suchpfade | 302 |
| Truncate | 554 |
| TSaveDialog | 540 |
| TSaveFileDialog | 422 |
| TSavePictureDialog | 540 |
| TScreen | 406 |
| TScreen, Eigenschaften..... | 407 |
| TScreen, Ereignisse | 408 |
| TSDFDataset, Eigenschaften..... | 719 |
| TShape | 420, 476 |
| TSpeedbutton | 420 |
| TSpinEdit | 422, 513 |
| TSplitter | 431 |
| TSQLConnection, Ableitungen v. . | 727 |
| TSQLConnection-Ableitungen, Eigenschaften v. | 728 |
| TSQLite3Connection | 727 |
| TSQLQuery, WHERE-Klausel | 732 |
| TSQLTransaction | 726 |
| TStream | 353 |
| TStream-Ableitungen | 557 |
| TStringgrid | 422, 478 |
| TStringList..... | 354, 397 |
| TStrings | 354 |
| TStringStream..... | 561 |
| TSynEdit | 222 |
| TThread, Methoden/Eigenschaften | 638 |
| TToolbar | 420 |
| TToolbutton..... | 420 |
| TTrayIcon | 477 |
| TTreeView | 422, 486 |
| TTreeView, Eigenschaftem | 487 |
| TWinControl..... | 361, 415, 417 |
| TWinControl, Eigenschaften..... | 418 |
| TWinControl, Events..... | 418 |
| Typ-Deklarationsblock..... | 35, 38 |
| Type Library Editor | 655 |
| Typenbibliotheks-Editor..... | 655 |
| Typinformationen zur Laufzeit | 355 |
| Typisierte Datei | 549 |

U

| | |
|------------------------------------|-------------------|
| Übersetzungsdatei..... | 73 |
| Überwachte Ausdrücke..... | 235 |
| Uhrzeitabweichung | 18 |
| UIB..... | 682, 716 |
| Umgebungsvariable, Wert d..... | 375 |
| Umlaute falsch dargestellt | 182 |
| Umlaute | 77 |
| Umschaltbox | 467 |
| Unicode | 77, 126, 604, 723 |
| Unit Classes | 40, 353 |
| Unit DateUtils..... | 40 |
| Unit des Hauptformulars, Quelltext | 248 |
| Unit FileUtil..... | 77 |
| Unit LCLProc | 77 |
| Unit Lconvencoding | 78 |
| Unit Math | 40 |

| | |
|--|---------|
| Unit mit neuer Komponentenklasse erzeugen | 167 |
| Unit nicht gefunden..... | 70 |
| Unit StrUtils | 40 |
| Unit System | 40 |
| Unit SysUtils..... | 40 |
| Unit | 244 |
| Unit, kompilierte | 250 |
| Unit-Abhängigkeiten | 138 |
| Unit-Aufbau | 39 |
| Unit-Bezeichner | 528 |
| Unit-Konzept | 32, 39 |
| Unit-Pfade..... | 139 |
| Units f. versch. Plattformen..... | 65 |
| Units in mehr. Projekte einbinden .. | 64 |
| Units mit demselben Namen | 141 |
| Unit, Portabilität | 530 |
| Units, Suche nach | 40 |
| Units, unbenutzte | 29 |
| Units, vorkompilierte..... | 53 |
| Unit-Testing | 279 |
| Unix, Datei öffnen | 322 |
| Unix, Dateiendung..... | 321 |
| Unix, Fensterverhalten..... | 394 |
| Unix, grafische Oberflächen | 324 |
| Unix, Pfadtrenner | 321 |
| Unix, Schreibzugriffe..... | 322 |
| Unix, Verzeichnisgröße/-datum | 322 |
| Unix, Verzeichnishierarchie..... | 322 |
| Unterfenster | 17 |
| Untypisierte Datei | 550 |
| USB-Gerät | 579 |
| User-Abfragen..... | 419 |
| Uses-Anw., Komp. hinzufügen..... | 249 |
| Uses-Klausel..... | 40 |
| Uses-Klausel, Portierung..... | 532 |
| UTF16 | 306 |
| UTF-8..... | 77, 723 |
| UTF8ToAnsi..... | 77 |
| UTF8ToSys | 541 |

V

| | |
|---|---------|
| valgrind..... | 33, 155 |
| varargs..... | 48 |
| Variablen, lokale erzeugen | 127 |
| Variablenanzeige | 235 |
| Variablendekl.-vervollständigung | 227 |
| Variablen-Deklarationsblock | 38 |
| Variablenvervollständigung | 24 |
| Varianter Record | 35 |
| Veränderungen in DB melden | 689 |
| Verfügbarkeitssteug. e. Controls .. | 452 |
| Verlinkte Liste | 36 |
| Veröffentlichte Eigenschaften in Datenstrom speichern..... | 354 |
| Verschlüsselungs-Stream..... | 561 |
| Versionskontrollsystem | 126 |
| Versteckte Datei..... | 572 |
| Verstecktes Fenster anzeigen | 385 |
| VertScrollbar..... | 377 |
| Verzeichnis aufräumen..... | 121 |

| | |
|--|---------|
| Verzeichnisaufbau anzeigen..... | 486 |
| Verzeichnisauswahl..... | 422 |
| Verzeichniseinträge ausgeben..... | 568 |
| Verzeichnisinhalt anzeigen..... | 514 |
| Verzeichnisse durchsuchen..... | 567 |
| Verzeichnissuche, rekursive..... | 567 |
| Verzeichnistrenner..... | 57, 301 |
| Verzeichniswechsel..... | 570 |
| Virtual PC..... | 311 |
| Virtuelle Methode..... | 43 |
| Virtuelle Unit..... | 65 |
| Visible..... | 351 |
| Visuelle Komponenten..... | 709 |
| Voreinstellungen von L..... | 189 |
| Vorwärtsdeklaration..... | 21, 37 |
| Vorzeichenbehalteter 32-/64-Bit-Integer..... | 342 |
| Vorzeichenloser 32-/64-Bit-Integer..... | 342 |

W

| | |
|--|---------|
| Web Service Toolkit..... | 653 |
| Webservice als Bibliothek..... | 663 |
| Webservice registrieren..... | 661 |
| Webservice, Nachrichten-Logging..... | 675 |
| Webservice, Stadien d. Bearbeitg. v. Clientanfragen..... | 679 |
| Webservice, URL..... | 652 |
| Webservice-Erweiterungen..... | 679 |
| Webservice-Implementationskl. | 661 |
| Webservice-Module..... | 654 |
| Weiß, Kodierung..... | 593 |
| Werkzeuge..... | 176 |
| Werteliste füllen..... | 287 |
| WHERE-Klausel, TSQLQuery..... | 732 |
| Widechar..... | 34 |
| Widestring..... | 34, 306 |
| Widgets..... | 409 |
| Widgetset wechseln..... | 239 |
| Widgetset wechseln..... | 71 |

| | |
|---------------------------------------|--------------|
| Widgetset, Kommunikation mit... .. | 521 |
| Widgetset, LCL-Interface zum..... | 519 |
| Widgetsets..... | 15 |
| Widgetsets, Eigenschaften von... .. | 136 |
| Widgetsets, mögliche..... | 239 |
| Wiederholen-Button, Ergebnis..... | 383 |
| Win32/64..... | 239, 352 |
| WinCE..... | 239 |
| WinCE, Bluetooth..... | 310 |
| WinCE, Build-Umgebung..... | 313 |
| WinCE, Dialoge..... | 310 |
| WinCE, Fenstersteuerung..... | 318 |
| WinCE, Schließen-Schalter..... | 320 |
| WinCE, Shellroutinen..... | 310 |
| WinCE, Standard-Steuererelemente..... | 310 |
| WinCE, Synchronisationsmodule..... | 310 |
| WinCE, Systemsteuerung..... | 310 |
| WinCE-API..... | 310 |
| Windows CE/Mobile..... | 94, 309, 352 |
| Windows, Datei öffnen..... | 303 |
| Windows, Kern-API-Routinen..... | 304 |
| Windows, Pfadcodierung..... | 307 |
| Windows, Unicode..... | 304 |
| Windows, Unicode-Kodierung..... | 306 |
| Windows, Verzeichnisstruktur..... | 307 |
| Windows-API..... | 303 |
| Windows-API-Fukt., Wrapper um..... | 523 |
| Windows-API-Nachrichten..... | 55 |
| Windows-Ressourcendatei..... | 248 |
| WindowState ändern, Ereignis..... | 378 |
| WindowState, Konstanten..... | 395 |
| Windows-Units markieren..... | 56 |
| Windows-Units unter LCL..... | 56 |
| Windows-Verzeichnisnamen such..... | 308 |
| Wortumbruch..... | 464 |
| Write..... | 554 |
| ws_helper, Kommandozeilenopt..... | 673 |
| WSDL..... | 653, 655 |
| WSDL importieren..... | 671 |
| WSDL-Datei parsen..... | 671 |

| | |
|---------------------------------------|-----|
| WSDL-Datei übersetzen..... | 672 |
| WST, Serialisierungsprotokolle d..... | 668 |
| WST, Servicebeschreibung..... | 655 |
| WST, Transportprotokolle..... | 667 |
| WST-Laufzeitumgebung..... | 662 |

X

| | |
|-----------------------------------|-----|
| XML-Datei lesen..... | 68 |
| XML-Schema-Def.-Datei parsen..... | 671 |
| XSD-Datei..... | 671 |

Z

| | |
|---|---------------|
| Zeichen maskieren..... | 129 |
| Zeichenfeld..... | 476 |
| Zeichenfläche..... | 589 |
| Zeichenkettenbearbeitung, Unit..... | 40 |
| Zeichenkettenlisten..... | 354 |
| Zeichenobjekt hat Fokus..... | 606 |
| Zeichenpinsel, Eigenschaften..... | 592 |
| Zeichensatz..... | 604 |
| Zeichensatzumwandlungen, API-Funktionen..... | 547 |
| Zeichenstift, Eigenschaften..... | 594 |
| Zeichentabelle..... | 125 |
| Zeichnen in eine Bitmap..... | 591 |
| Zeichnen v. Rahmen..... | 420 |
| Zeiger, typisierter..... | 45 |
| Zeigergröße, 64-Bit-Prozessor..... | 342 |
| Zeilenende-Def. i. Textdateien..... | 301 |
| Zeilenendezeichen..... | 180, 302, 554 |
| Zeilenposition d. Cursors im Dataset speichern..... | 690 |
| Zeilenumbrüche in Dateien..... | 556 |
| ZeosLib..... | 682, 716 |
| Zieh-Operation, Ereignis..... | 417 |
| Zielbetriebssysteme, mögliche..... | 297 |
| Ziel-CPUs, mögliche..... | 297 |
| Zieldateiformate, speichern in..... | 507 |
| Zielplattform, Einstellungen..... | 153 |
| Zielverzeichnis..... | 144 |
| Zwischenablage..... | 122 |