

14.2 Die Indy-Komponenten

von Dr. Karlheinz Mörth

Delphi 6 bietet eine große Anzahl von Komponenten, die zur Internet-Programmierung dienen. Neben der *Internet*- und *FastNet*-Registerkarte enthält die Komponentenpalette von Delphi 6 nun auch drei weitere Registerkarten, auf denen sich die *Internet-Direct*-Komponenten befinden. Diese Sammlung, für die sich auch die Bezeichnung Indy-Komponenten eingebürgert hat, ist aus den Winshoes-Komponenten hervorgegangen und war schon unter Delphi 5 von vielen Delphi-Programmierern verwendet worden. Als Open Source konnte sie jeder zum eigenen Gebrauch verwenden, ohne dafür bezahlen zu müssen. Als bewährtes Instrumentarium der Netzprogrammierung wird sie nun standardmäßig mit der neuen Delphi-Version (Enterprise und Professional) ausgeliefert. Updates der Bibliothek sind über die Site der Firma *Nevrona Designs* (»www.nevrona.com/Indy«) zu beziehen. Die letzten Versuche für diesen Abschnitt wurden mit Indy (Winshoes 8.1) v8.10 gemacht.

Indy unterstützt die wichtigsten Internet-Protokolle, sowohl Low-Level-Protokolle wie IP, ICMP, TCP, UDP, als auch Anwendungsprotokolle wie FTP, Telnet, SMTP, POP3, HTTP und so weiter. Zu den ganz großen Pluspunkten dieser Bibliothek gehört zudem ihr plattformübergreifender Charakter. Dadurch können ein- und dieselben Komponenten sowohl in Delphi- als auch in Kylix-Projekten verwendet werden. Lassen sich viele Programme noch nicht ohne weiteres von Windows nach Kylix portieren, so ist die parallele Entwicklung mit Indy ein durchaus praktikables Unterfangen.

14.2.1 Die drei Indy-Registerkarten: Clients, Server und Vermischtes

Netz-Applikationen basieren meistens auf dem bekannten Paradigma, das als Client-Server-Modell bezeichnet wird. Dieses geht von der Vorstellung aus, daß eine Seite der Verbindung als Dienstleistungsnehmer (Client) fungiert, die andere als der Dienstleistungsanbieter (Server), der exakt definierte Dienstleistungen für die Klienten zu offerieren hat. Der Server spielt dabei in gewissem Sinne eine passive Rolle, indem er auf hereinkommende Anfragen wartet und in der Folge reagiert. Die Initiative zur Kommunikation geht vom Client aus. Im Grunde läßt sich alles auf zwei Basisaktionen reduzieren: ein Client lanciert eine Anfrage (*request*) und der Server reagiert darauf mit einer entsprechenden Antwort (*response*). In diesem Zusammenhang sollte man sich die Tatsache bewußt machen, daß die Grenze zwischen Hard- und Software hier in der Terminologie oft verschwimmt. Sehr oft wird Server heute im Sinne von Rechner gebraucht, der anderen Rechnern Dienste zur Verfügung stellt. Die Server, von denen hier im folgenden die Rede sein wird, gehören natürlich in den Bereich der Software.



Es sind Programme, die kontinuierlich in Betrieb sind, um jederzeit auf ankommende Anfragen reagieren zu können. In der Unix/Linux-Welt werden Prozesse, die Serverdienste leisten, oft als Daemons bezeichnet. Für die meisten im Internet zur Verfügung gestellten Dienste existieren solche »Dämonen«, die oft durch ein nachgestelltes »D« als dieser Kategorie von Programmen zugeordnet kenntlich gemacht werden. So startet man einen Webserver auf vielen Systemen durch den Aufruf von HTTPD, was für »Hypertext Transfer Protocol Daemon« steht. Im Übrigen ist der Ursprung dieses Terminus nicht gespenstischer Natur, sondern ein Akronym für »Disk And Execution MONitor«. Weitere Beispiele für solche im Hintergrund laufenden Prozesse kennen Sie vielleicht vom Starten eines Linux-Rechners. Wenn Sie da aufmerksam mitlesen, stoßen Sie auf *inetd*, der die Internet-Ports überwacht, und LPD (Line Printer Daemon), der Druckaufträge entgegennimmt und weiterverarbeitet.

Ein klassisches Beispiel für Server, die von allen, die im Internet unterwegs sind, verwendet werden, von deren Existenz die meisten Anwender aber gar keine Kenntnis haben, sind die sogenannten Domain Name Server, die als Service alphanumerische Domainnamen wie zum Beispiel »www.bbc.com« in numerische IP-Nummern 212.58.224.35 übersetzen. Diese Zahlen, auf denen alle Kommunikation im Internet beruht, entnehmen DNS- (Domain Name System)-Server riesigen Datenbanken, die die Daten für die entsprechenden Zuordnungen enthalten. Alle möglichen Programme, die im Internet eingesetzt werden, greifen auf DNS-Server zu, die jeweils nur für eine bestimmte geographische Region zuständig sind. Kein DNS-Server enthält alle Daten der Welt. Findet er zu einer Anfrage keine Antwort, wird der Server selbst zum Client und fragt bei anderen Servern an.

Zu einem gängigen Begriff ist mittlerweile der »Webserver« geworden, der für die Veröffentlichung von Webseiten im Internet eingesetzt wird. Ein Webserver wartet auf Anfragen von Clients, bei denen es sich zumeist um Webbrowser wie Netscape, Mosaic, Internet Explorer oder Lynx handelt. Die Client-Anfrage an einen Webserver besteht normalerweise in einer Internet-Adresse, die auf eine bestimmte Ressource, ein Text-Dokument, eine Grafik oder eine Sounddatei verweist. Auf eine solche Anfrage sendet der Webserver die gewünschte Seite an den Client. Diese Kommunikation wird, wie wir später noch genauer sehen werden, über das Hypertext Transfer Protocol abgewickelt.

Man sollte sich in diesem Zusammenhang vor einer konzeptuellen Vermengung von Daten und Dienstleistung in Acht nehmen. Die Metaphorik von »Geben und Nehmen« läßt sich auf das Verhältnis »Server – Client« nur im Hinblick auf Dienste umlegen. Daß beide Teile sowohl Daten empfangen als auch erhalten können, läßt sich schön am Beispiel eines Servers zeigen, mit dem wohl die meisten Leser schon zu tun hatten, einem »Mailserver«. Auch dieser Server hat eine ständige Verbindung zum Internet. Seine Dienstleistungen sind vielfältig und ziemlich komplex. Er nimmt Mails auf Anfrage für den Client entgegen und verwahrt, also speichert diese. Hierzu bedarf es einer ziemlich ausgeklügelten Verwaltung

der abgelegten Daten. Die Clients, die sogenannten E-Mail-Programme (Eudora, Outlook, Netscape Messenger usw.), kontaktieren ihn in zwei Fällen: (a) wenn sie Mails abholen und (b) wenn sie Mails versenden wollen. Bei diesen Transaktionen wandern die Daten in beide Richtungen, in vielen Fällen wiederholt, hin und her.

Die verfügbaren Indy-Komponenten sind in der Delphi-IDE auf drei Registerkarten verteilt, die auch das zuvor beschriebene Client-Server-Model reflektieren. Auf der ersten Karte finden Sie die Clients, auf der zweiten eine Reihe von Servern, die in vielen Fällen komplementär zu jenen der ersten Palette sind. Auf der dritten findet sich alles, was der Kategorie nach nicht auf eins oder zwei unterzubringen war.

Diese neunzehn Indy-Clients sind auf der ersten Registerkarte angeordnet:



TIdTCPClient	Diese Komponente kapselt einen vollständigen TCP- (Transmission Control Protocol)-Client. Sie stellt Klassen, Typen, Prozeduren und Funktionen zur Verfügung, die es dem Anwender ein Leichtes machen, eine Punkt-zu-Punkt-Verbindung zwischen zwei Rechnern einzurichten. <i>TIdTCPClient</i> ist der Vorfahr einer Reihe von Indy-Client-Komponenten, die spezifische Protokolle implementieren. <i>TIdDayTime</i> , <i>TIdEcho</i> , <i>TIdFinger</i> , <i>TIdFTP</i> , <i>TIdGopher</i> , <i>TIdHTTP</i> , <i>TIdNNTP</i> , <i>TIdPOP3</i> , <i>TIdQUOTD</i> , <i>TIdSMTP</i> , <i>TIdTelnet</i> und <i>TIdWhois</i> sind alle Nachfahren von <i>TIdTCPClient</i> .
TIdUDPClient	Das User Datagram Protocol (UDP) ist im RFC 768 von Jon Postel (August 1980) erstmals beschrieben und stellt neben TCP das zweite wichtige Protokoll dar, das bei der Implementierung von Internetprogrammen eine Rolle spielt. Im Gegensatz zum verbindungsorientierten TCP-Protokoll funktioniert UDP verbindungslos.
TIdDayTime	Ein <i>DayTime</i> -Server sendet den jeweiligen Tag und die Uhrzeit über den Port 13 und dient neben der Synchronisation von Programmen und Rechnern vorrangig zum Debuggen (RFC 867).
TIdDNSResolver	Das Akronym DNS bedeutet Domain Name System. <i>TIdDNSResolver</i> basiert nicht wie die meisten dieser Komponenten auf TCP, sondern auf UDP. <i>TIdDNSResolver</i> erlaubt den Zugriff auf DNS-Server (RFC 1034, 1035, 1591, 1183 und 2181).
TIdEcho	<i>TIdEcho</i> ist ein auf TCP beruhender Client. Er basiert auf dem sehr frühen Protokoll, das 1983 von Jon Postel im



	RFC 862 festgeschrieben wurde. Das Echo-Protokoll dient vor allem der Suche von Fehlern und dem Messen der Performanz von Datentransfervorgängen.
TIdFinger	Die auch auf TCP aufbauende Komponente erlaubt es, mit Hilfe des Finger-Protokolls, auch User Information Protocol genannt, Informationen über einzelne User auf einem bekannten Host zu ermitteln (RFC 1288).
TIdFTP	FTP steht für File Transfer Protocol (RFC 959, J. Postel und J. Reynolds, 1985). Hierbei handelt es sich um ein System, das den Zugriff auf die Dateisysteme anderer Rechner ermöglicht. Vielfältige Archive werden über dieses Protokoll zur Verfügung gestellt.
TIdGopher	Mit Hilfe des Gopher-Protokolls entstand schon in der Frühzeit des Internets ein weltumspannendes, über zahllose Rechner verteiltes Informationssystem, das ursprünglich an der Universität von Minnesota entwickelt worden war. Lange Zeit war es durch seine auf Intuition abzielende menügesteuerte Navigation sehr populär, allerdings spielt es heute nur noch beim Zugriff auf ältere Datenbestände eine Rolle (RFC 1436).
TIdHTTP	Das Hypertext Transfer Protocol (HTTP) ist in zwei Internet-Standard-Dokumenten definiert, die Version HTTP 1.0 in RFC 1945, die Version HTTP 1.1 in RFC 2616. Es verdankt seinen hohen Bekanntheitsgrad der Tatsache, daß es vornehmlich in Web-Browsern, Web-Servern und Web-Robots Anwendung findet.
TIdICMPClient	Das Internet Control Message Protocol ist integraler Bestandteil von IP. Zu den Aufgaben dieses Protokolls gehören die Meldung von Netzwerkfehlern (beispielsweise ein Host ist unerreichbar, die Meldung von Netzwerküberlastungen, ein Router beginnt zu viele Pakete zu puffern), Hilfe beim Auffinden von Fehlern und das Melden von Timeouts. Besondere Bedeutung kommt ICMP dadurch zu, daß es die Grundlage für viele <i>Ping</i> - und <i>TraceRoute</i> -Anwendungen ist (RFC 792).
TIdPOP3	POP bedeutet Post Office Protocol, die gegenwärtig aktuelle Version hat die Nummer 3. Mit POP3 können E-Mails von Mail-Servern empfangen werden.
TIdNNTP	Das Network News Transport Protocol (NNTP) dient zum Austausch von News. Mit dieser Komponente läßt sich auf einfache Art und Weise ein News-Reader einrichten (RFC 977, 2980).

TIdQUOTD	<i>Quotd</i> steht für Quote of the day (Zitat des Tages) und holt einen String von einem entsprechenden Quotd-Server (RFC 865).
TIdSMTP	Mit dem Simple Mail Transfer Protocol (SMTP) werden im Internet 7-Bit-ASCII-Nachrichten versandt (RFC 821).
TIdSNTP	Das Simple Network Time Protocol (SNTP) dient zur Synchronisation von Rechneruhren im Internet. Dieses Protokoll erlaubt ähnlich wie das ältere Time-Protocol die Abfrage der Uhrzeit von einem Time-Server. SNTP gilt als stabiles Protokoll, mit dem Genauigkeiten zwischen 1 und 50 Millisekunden auch über große Distanzen zu erreichen sein sollen (RFC 2030).
TIdTelnet	Das Telnet-Protokoll gehört zu den Sauriern unter den Internet-Standards. Mit seiner Hilfe kann man eine Kommunikation mit einem Server einrichten, die es möglich macht, nachdem man sich auf dem Server eingeloggt hat, alle Eingaben auf der Konsole auf den Server umzulenken, und diesen durch jene Eingaben auch zu steuern (RFC 854-861).
TIdTime	<i>TIdTime</i> implementiert einen Client auf der Grundlage des Time Protocols, das primär zur Synchronisation in lokalen Netzwerken dient. Die Zeit wird in diesem Protokoll in Sekunden seit dem 1. Januar 1900, 0 Uhr GMT gezählt. Dies hat eine Beschränkung zur Folge, die eine Verwendung über das Jahr 2035 hinaus nicht zulässt. Die Kommunikation läuft über den Port 37 (RFC 868).
TIdTrivialFTP	Trivial FTP (TFTP) ist ein einfaches, auf UDP basierendes Protokoll. <i>TFTP</i> kann keine Verzeichnisse auflisten und verfügt in der gegenwärtigen Form über keine Möglichkeiten zur Identifizierung von Usern (RFC 1350, 1782 und 1783).
TIdWhois^	Mit Hilfe des <i>Whois</i> -Dienstes erhält man Zugriff auf verteilte Datenbanken, die Informationen über die Besitzer von Domainnamen oder E-Mail-Adressen enthalten. Die Kommunikation läuft über den Port 43 (RFC 954).

Dreiundzwanzig weitere Komponenten finden sich auf der zweiten Registerkarte. Beim Großteil handelt es sich um Server:



TIdTCPServer	Dies ist die wichtigste Server-Komponente der Indy-Suite. Sie ist der Vorfahr einer Reihe von Klassen, die spezielle Server implementieren: <i>TIdChargenServer</i> , <i>TIdDayTimeServer</i> , <i>TIdDICTServer</i> , <i>TIdEchoServer</i> , <i>TIdFingerServer</i> , <i>TIdGopherServer</i> , <i>TIdHostNameServer</i> , <i>TIdHTTPServer</i> , <i>TIdIRCServer</i> , <i>TIdNNTPServer</i> , <i>TIdQUOTDServer</i> , <i>TIdTelnetServer</i> und <i>TIdWhoisServer</i> . <i>TIdTCPServer</i> unterstützt mehrfache Threads, was die Voraussetzung für effektiven Serverbau darstellt. Wie <i>TIdTCPClient</i> ist es ein Nachfahr von <i>TIdTCPConnection</i> .
TIdUDPServer	<i>TIdUDPServer</i> wird als Vorfahr von Komponenten gebraucht, die auf dem User Datagram Protocol (UDP) aufsetzen. <i>TIdTrivialFTPServer</i> ist beispielsweise eine solche Komponente.
TIdChargenServer	Ein <i>Chargen</i> -Server schickt an den anfragenden Client einen Strom von Zeichen zurück, der zum Feststellen der Übertragungsgeschwindigkeit verwendet werden kann (RFC 864).
TIdDayTimeServer	Das <i>DayTime</i> -Protokoll sendet den jeweiligen Tag und die aktuelle Uhrzeit in Form eines lesbaren Strings über das Internet. Es wird auch zum Debuggen verwendet.
TIdDICTServer	Das Dictionary Server Protocol ist ein TCP-basiertes Frage-Antwort-Protokoll, mit dessen Hilfe ein Client auf standardisierte Einträge von Wörterbuchdatenbanken zugreifen kann (RFC 2229).
TIdDISCARDServer	Discard-Server werden hauptsächlich bei der Fehlersuche in Netzwerken eingesetzt. Wie der Name schon sagt (to discard heißt »verwerfen«), nimmt ein solcher Server Daten entgegen, die er dann ohne irgendeine Art der Auswertung sofort verwirft.
TIdEchoServer	Durch das Anrufen eines Echo-Servers kann man auf der IP-Ebene feststellen, ob der Server erreichbar ist. Ein Echo-Server wartet auf Port 7 auf eingehende Daten.
TIdFingerServer	Finger-Server dienen dazu, Informationen über User zur Verfügung zu stellen.



TIdGopherServer	Das auf dem Gopher-Protokoll aufbauende verteilte Informationssystem verliert im World Wide Web heute immer mehr an Boden. Gopher-Dienste zeichnen sich durch ihre Einfachheit und Übersichtlichkeit aus.
TIdHostNameServer	Dieser Server liefert maschinenlesbare Namen und Adreßinformationen über Hosts, Gateways, Netzwerke und auch Domains (RFC 953).
TIdHTTPServer	Mit dieser Komponente läßt sich auf einfachste Art und Weise ein voll funktionstüchtiger Web-Server programmieren. (RFC 1945, 2616)
TIdIMAP4Server	Das Interactive Mail Access Protocol (IMAP) stellt eine anspruchsvollere Form des bereits erwähnten POP-Protokolls dar (RFC 2060).
TIdIRCServer	Auf dem Internet Relay Chat (IRC) Protocol basieren die sogenannten Chat-Rooms. Es handelt sich um ein textbasiertes Konferenzprotokoll, das die Netzkomunikation über TCP/IP realisiert (RFC 1459).
TIdMappedPortTCP	Diese Komponente implementiert einen Proxy-Server für Verbindungen auf einen entfernten Rechner. Sie ist ein Nachfahr von <i>TIdTCP</i> Server.
TIdNNTPServer	Mit dieser Komponente läßt sich ein Server, der auf dem Network News Transport Protocol aufbaut, programmieren.
TIdQUOTDServer	Der Quote-of-The-Day-Server liefert auf Anfrage über den Port 17 einen beliebigen, vom Administrator festzulegenden String.
TIdSimpleServer	<i>TIdSimpleServer</i> ist von <i>TIdTCPConnection</i> abgeleitet. Im Gegensatz zu <i>TIdTCP</i> Server verfügt dieser Server über einen einzigen Thread.
TIdTelnetServer	Mit <i>TIdTelnetServer</i> bekommt man das Gerüst für einen Telnet-Server geboten.
TIdTimeServer	Diese Komponente ist die Server-Entsprechung von <i>TIdTime</i> . Auch hier gilt die Jahr-2035-Beschränkung.
TIdTrivialFTPServer	<i>TIdTrivialFTP</i> Server ist das Gegenstück zur <i>TIdTrivialFTP</i> -Komponente. Es gelten die zuvor bereits beschriebenen Beschränkungen.



TidTunnelMaster	<i>TidTunnelMaster</i> ist ein Nachfahr von <i>TidTCPServer</i> . Als Controller eines IP-Tunnels fungiert er als Proxy für Anwenderverbindungen und ermöglicht durch IP-Tunnel den Aufbau verschlüsselter Kommunikationsverbindungen im Internet, sogenannter Virtual Private Networks. Die Bezeichnung rührt daher, daß den Anwendern die sichere Verbindung, die über den »Tunnel« im Internet läuft, wie eine Verbindung innerhalb eines privaten Netzes erscheint.
TidTunnelSlave	Obiger <i>TidTunnelMaster</i> funktioniert in Kombination mit einem <i>TidTunnelSlave</i> , der als Server am Endpunkt des IP-Tunnels fungiert.
TidWhoisServer	Mit dem <i>TidWhoIsServer</i> ermöglicht Indy dem Entwickler die einfache Implementierung eines Nickname-Servers. Mit Whois-Abfragen werden in der Regel Datenbanken nach Domain-Registrierungen durchsucht.

Die restlichen Komponenten auf der dritten Registerkarte mit der Aufschrift *Indy Misc* umfassen verschiedene Komponenten, die beim Programmieren mit Indy hilfreich sein können. Eine Reihe davon ist für den Einsatz im E-Mail-Verkehr vorgesehen und dient der Ver- und Entschlüsselung elektronischer Nachrichten.



TidAntiFreeze	Diese Komponente verhindert das »Einfrieren« der grafischen Oberfläche, während Indy-Prozeduren und -Funktionen ausgeführt werden.
TidDateTimeStamp	<i>TidDateTimeStamp</i> verarbeitet Datums- und Zeitangaben in verschiedenen Formaten, die von Internet-Protokollen verwendet werden. Die Methoden dieser Komponente arbeiten mit Zeiteinheiten von Millisekunden, Sekunden, Tagen und Jahren.
TidIPWatch	<i>TidIPWatch</i> dient dazu, den Online-Status und die aktuelle IP-Adresse festzustellen.
TidLogDebug	<i>TidLogDebug</i> stellt eine Möglichkeit dar, Log-Daten in eine Datei schreiben zu lassen. Darüber hinaus kann <i>TidLogDebug</i> auch <i>TidLogItemEvent</i> -Ereignisse auslösen.



TIdMessage	Diese Komponente kapselt eine vollständige Internet-Nachricht. Sie wird mit den Protokollen POP3, SMTP und NNTP verwendet und unterstützt auch Multipurpose Internet Mail Extensions (MIME).
TIdNetworkCalculator	<i>TIdNetworkCalculator</i> ist keine Rechenmaschine im üblichen Sinne, vielmehr handelt es sich um ein Validierungsinstrument für IP-Adressen.
TIdThreadMgrDefault	Der Thread-Manager eines Indy-Programms.
TIdThreadMgrPool	Indy unterstützt das Konzept des sogenannten Thread-Pooling. Darunter versteht man ein System zentraler Thread-Verwaltung, bei dem Threads bei Bedarf von einem Thread-Manager erzeugt werden und von diesem auf Anfrage Instanzen anderer Komponenten zur Verfügung gestellt werden. <i>TIdThreadMgrPool</i> verwaltet auch eine Liste aller aktiven Threads.
TIdVCard	Diese Komponente stellt die Implementierung der <i>VCard</i> -Spezifikation dar, die die Funktion einer elektronischen Visitenkarte übernehmen soll (RFC 2425 und 2426).
TIdIMFDecoder	Das Internet Message Format (IMF) legt einen Standard für Text-basierte elektronische Nachrichten (E-Mails) fest (RFC 822 und 1036).
TIdQuotedPrintableEncoder TIdQuotedPrintableDecoder	E-Mails waren ursprünglich auf Zeichen beschränkt, die sich mit sieben Bits darstellen lassen. Dies reichte für 128 Zeichen, die alle im Englischen gebräuchlichen Zeichen abdeckten. Quoted Printable (QP) ist eine Möglichkeit, Sonderzeichen auch in E-Mails zu integrieren, die auf der Grundlage von sieben Bits verschlüsselt sind. Diese Komponente erlaubt es, Texte in dieses Format zu konvertieren (RFC 2045).



TIdBase64Encoder TIdBase64Decoder	Auch mit dieser Komponente werden binäre Daten in 7-Bit-ASCII umgesetzt. <i>Base64</i> spielt besonders beim Versenden von binären Daten wie Grafiken, ZIP- oder TAR-Dateien eine Rolle. Attachments sind sehr oft mit Base64 kodiert (RFC 2045).
TIdUUEncoder TIdUUDecoder	<i>UUEncode</i> war vor <i>Base64</i> der Standard zur Umwandlung von Binär- in 7-Bit-Daten.
TIdXXEncoder TIdXXDecoder	<i>XXEncode</i> stellt eine Variante von <i>UUEncode</i> dar.
TIdCoderMD2 TIdCoderMD4 TIdCoderMD5	Die drei <i>CoderMD</i> -Komponenten dienen der kryptographischen Verschlüsselung von Daten zur Verhinderung des unerwünschten Zugriffs von Unbefugten. Sie bedienen sich hierzu des sogenannten RSA-Verfahrens. RSA steht für die drei Erfinder dieses Verschlüsselungsalgorithmus: Ron Rivest, Adi Shamir und Len Adleman.

Die Bibliothek bietet somit alles, was das Herz des Internet-Programmierers begehrt. Dem Spezialisten erlaubt sie die Verkürzung von Produktionszyklen, durch die standardisierten Verwendungsmechanismen wird es jedoch auch dem Einsteiger leicht, sich experimentierend zu brauchbaren Resultaten durchzukämpfen. Der Zahl der zur Verfügung stehenden Komponenten verbietet natürlich deren vollständige Erläuterung anhand von Beispielen. Im folgenden wird ihr Gebrauch beispielhaft an einigen einfachen, aber nichtsdestoweniger nützlichen Applikationen erläutert werden, wobei wir auch versuchen wollen, unser theoretisches Verständnis der Zusammenhänge zu vertiefen.

14.2.2 Erste Schritte mit Indy

Sie haben bereits gelernt, daß Rechner im Internet über sogenannte IP-Adressen identifiziert werden. Diese bestehen bekanntlich aus vier Byte und sind demnach 32 Bit lang. Viele Rechner arbeiten auch mit dynamisch vergebenen Adressen, die bei jeder Einwahl bei einem Provider neu vergeben werden. Sie sollten dabei sich dabei ins Bewußtsein rufen, daß es streng genommen nicht ein einzelner Rechner, sondern vielmehr eine Netzwerkkarte ist, die mit der IP-Adresse assoziiert wird, da ein Rechner ja auch mehrere solcher Schnittstellenkarten mit unterschiedlichen IP-Adressen haben kann.