

Indy Knowledge Base

© 2003-2005 Chad Z. Hower and the Indy Pit Crew

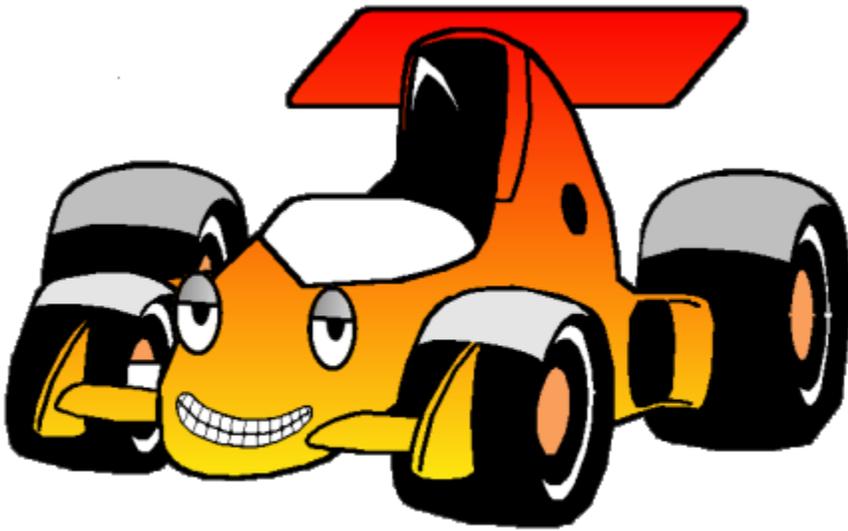


Table of Contents

Foreword	0
Part I Introduction	1
1 Welcome To the Indy Knowledge Base.....	1
2 Feedback	1
Part II General Indy Questions	1
1 Can I use Indy in my shareware/commercial applications?.....	1
2 What are these "Teams" you have and how can I help?.....	1
3 Where can I find a demo for ____?.....	1
Part III Winshoes	2
1 What is Winshoes?.....	2
2 Can I use Winshoes 7 and Indy at the same time?.....	2
3 How hard is it to migrate my application from Winshoes 7 to Indy?	2
Part IV Borland Delphi Installation	2
1 How do I install Indy into Borland Delphi?.....	2
2 How do I install Indy 8.0 in Delphi 6 Personal Edition?.....	3
3 In Indy 9.0, I am getting the error "Error on loading Winsock2 library (WS2_32.DLL)" and it only occurs on Win95 machines.....	3
4 When I try to compile my application that uses Indy it says it cannot find Indy.dcu or some other file?.....	3
5 When I attempt to setup Indy on a fresh copy of Delphi 4.0, it fails to compile?	3
6 When I try to compile Delphi says it cannot find SyncObjs?.....	3
7 After installing Indy, I am getting an error message such as: "The procedure entry point @ldglobal@Fetch\$qqrr175System@AnsiStringxoxo could not be located in the dynamic link library Indy60.bpl.....	3
8 After installing Indy, I get a "A device attached to the system is not functioning." when loading another package. What do I do about it?	4
9 Why do you change the Indy API between major Indy versions?.....	4
10 How hard is it to upgrade my code to the latest version of Indy?	4
Part V Borland C++ Builder Installation	4
1 How do I install Indy into Borland C++ Builder?.....	4

2	I tried to install Indy 8.0 in Borland C++ Builder and got missing package errors - the files seem to listed okay in the Indy source directory.	5
3	In Indy 9.0, there is no IndyC60.bpk. What do I do to get Indy to work with C++Builder 6?	6
Part VI Exceptions		6
1	Why does Indy raise exceptions as part of its normal operation?.....	6
2	Why do I keep getting EldConnClosedGracefully exceptions?.....	7
3	Why do I get 10038 exceptions raised when I shut down my servers?	7
4	When I connect with TIdLPR or TIdRSH, I get a EldSocketException saying "Socket Error # 10048 Address already in use." Why?.....	8
Part VII Socket Programming (Winsock)		8
1	What is the difference between blocking and non-blocking sockets?	8
2	Where can I find Socket error messages explanations, like Socket error 10057?.....	8
3	What is an "RFC" and where do I find them?.....	9
Part VIII Firewalls and Proxies		9
1	How do I use Indy with a Proxy?.....	9
2	How do I use a TIdFTP with a FTP Proxy?.....	10
3	How do I use a TIdFTP with a SOCKS proxy?.....	10
4	How do I use a TIdHTTP with a HTTP Proxy?.....	11
5	How do I use a TIdTCPClient with a HTTP Tunneling Proxy?.....	12
6	How do I use a TIdTCPClient with a SOCKS Proxy?.....	12
7	How do I use a TIdUDPClient with a SOCKS Proxy?.....	13
8	How do I use Indy with a NAT (Network Address Translator)?.....	13
9	How do I use FTP with SSL behind a NAT?.....	14
10	What should I do if I do not know what proxy a user will need to use and what the correct settings are?.....	15
Part IX Server		15
1	How can I send a record using Indy?.....	15
2	How can I send a file using Indy?.....	16
3	Is it safe to access a database connection in a TIdTCPServer.OnExecute event?.....	16
4	IHow do I get the Peer's IP Address in the TIdTCPServer?.....	16
Part X Ping, ICMP (Internet Control Message Protocol), Raw Sockets		17

- 1 I can not ping with Indy on Windows NT or Windows 2000. What do I do about this?..... 17
- 2 I can not ping with Indy using Linux. What do I do about this?..... 17

Part XI SMTP (Simple Mail Transport Protocol) and E-Mail 17

- 1 How can I send / receive HTML email?..... 17
- 2 How can I send or receive mail attachments?..... 17
- 3 Whenever I do a LoadFromFile or a LoadFromStream, I get an EldReadTimeout exception. What am I doing wrong?..... 18
- 4 How do I send mail directly to a recipients mail server? 18
- 5 How do I send emails through a SMTP server that require authentication?..... 19
- 6 I am using the TIdSMTP but when I try to send mail I get an error message such as: "521 potserver.organization.com access denied" or "'572 Relay Not Authorized."..... 20
- 7 I am using the TIdSMTP but when I try to send mail I get an error message such as: "530 Must issue a STARTTLS command first"..... 20
- 8 In C++Builder, a class methods such as TIdSMTP.QuickSend require an extra parameter. What parameter should I pass?..... 21
- 9 Where can I find examples for using the SMTP / POP3 server components?..... 21
- 10 Why doesn't the MAPS Demo work anymore?..... 21
- 11 A SMTP server is treating E-Mail from my program as if it was spam or the server rejects it with a 554 error. I don't have any problems with Outlook. How do I fix this? 21
- 12 But that still does not help with SpamAssassin. What now?..... 22

Part XII SSL (Secure Sockets Layer) or TLS (Transport Layer Security) 22

- 1 Can I legally use and/or export software that uses Indy's SSL capabilities?..... 22
- 2 Where can I download Indy's SSL support .DLL's and source-code?..... 22
- 3 I need SSL help, where can I find answers?..... 22
- 4 How do I use FTP with SSL?..... 23
- 5 What is the difference between implicit TLS and explicit TLS?..... 24

Part XIII FTP (File Transfer Protocol) 24

- 1 With some servers, when I try to abort a transfer, the client is automatically disconnected from the server, instead of aborting just the transfer. Is this an Indy bug? 24
- 2 How do I obtain the size of a file I am about to download with TIdFTP? 25

3 I want to download files only if they have been modified on the server by comparing the date and time. How do I do this with Indy?	25
4 In TIdFTP, is there a method to parse the contents of the List result, so I can use a ListView component to display the directory?	26
5 When using the TIdFTP.SetModTime function, why do I get a "no such file or directory" but the file exists.....	27
Part XIV HTTP (Hypertext Transfer Protocol)	28
1 I am getting a 403 Forbidden error with some sites while they work fine with Opera, Netscape Communicator, or Internet Explorer. How do I fix this?.....	28
Part XV IP Multicasting	28
1 TTIdIPMCastClient and TIdIPMCastServer do not work on Linux?	28
Part XVI Migrating to Indy 10	28
1 Where is the Intercept property?.....	28
2 TIdFTP.DirectoryListing is returning nothing for my FTP server. Why?	29
3 Where is the TIdPOP3.APOP property?.....	29
Part XVII Miscellaneous Topics	29
1 Why did the chicken cross the road?.....	29
Part XVIII For Further Reading	30
1 Indy Introductory Articles.....	30
2 Socket Programming	31
3 RFCs (Request for Comments) and Internet Drafts.....	31
4 Indy Exceptions.....	32
5 HTTP (Hypertext Transfer Protocol).....	32
6 SSL (Secure Sockets Layer) or TLS (Transport Layer Security).....	32
7 Web Services.....	33
8 Other Indy Websites.....	33
Index	34

1 Introduction

1.1 Welcome To the Indy Knowledge Base

Welcome to the Indy Knowledge Base. Please read this before you post your questions about Indy because it may be answered here.

Please check <http://www.indyproject.org/KB/> for the latest updates to the Indy Knowledge Base.

1.2 Feedback

Before [E-Mailing the maintainer](#), make sure you do not ask for any support. If you need support, please see <http://www.indyproject.org/Support.html>.

2 General Indy Questions

2.1 Can I use Indy in my shareware/commercial applications?

Yes, you may use Indy in your commercial or shareware application if you follow the Indy license. We are aware of several commercial programs and even a commercial component set that uses Indy.

Normally what we ask is you use a "made with Indy" or even the button on the website in an about box or such, and the license text in the docs somewhere. You should display the following copyright notice somewhere in your program such as an About Box, splash screen, or documentation:

```
Portions of this software are Copyright (c) 1993 - 2004, Chad Z. Hower (Kudzu) and the Indy Pit Crew - http://www.nevrona.com/Indy/
```

2.2 What are these "Teams" you have and how can I help?

The teams page can be viewed at <http://www.nevrona.com/Indy/Teams/> and contains information about each team, its members, and information on how to help out!

2.3 Where can I find a demo for ___?

Go to the downloads page on the Indy website (<http://www.indyproject.org/>), select downloads, and then select your version of Indy. There is a section near the bottom of the page where you can download the demo programs. You may also find demos for Indy 10 at <http://www.indyproject.org/Socket/Demos/index.iwp> and AToZed Software has a Demo Playground at <http://www.atozed.com/indy/Demos/>.

Please visit <http://www.indyproject.org/support.html> for information about where to make inquiries about Indy and the demos but do so only after downloading the demos at the location described earlier.

Please do not ask individual Indy developers for demo programs. Most developers working on Indy do not have time to help individual users and you might be addressing someone who is unable to help you.

3 Winshoes

3.1 What is Winshoes?

Winshoes was the predecessor to Indy and was the name until version 8.0 when it was officially changed to Indy. With version 8, Indy began to support Linux and thus the need to remove "Win" from Winshoes.

3.2 Can I use Winshoes 7 and Indy at the same time?

Yes, you can do so because the unit and class names are totally different. We recommend that you switch to Indy because most Winshoes 7 development has stopped in favor of Indy development and you should only keep using Winshoes 7 if there is a very compelling reason to do so (e.g. you still need support for older Borland products, or your application is only being maintained). We say this because Winshoes 7 may not have some newer features and there are some bugs that were fixed since Winshoes 7.

3.3 How hard is it to migrate my application from Winshoes 7 to Indy?

This greatly depends on your code. Indy is based closely on Winshoes 7 and most interface changes are slight. Most users find that they can port their applications quite easily to Indy.

4 Borland Delphi Installation

4.1 How do I install Indy into Borland Delphi?

- Close the Borland Delphi IDE if it is open.
- If you are using the version of Indy included in Delphi 6 or 7, use the MSI installer to remove Indy.
- Remove all Indy files including **dclIndy*.bpl**, **Indy*.bpl**, the **ld*.pas**, and the **ld*.dcu's**. Take care that you only remove the old Indy files and not something else. Be sure that you also remove any **Indy*.bpl** from your Windows\System directory.
- Place the new version of Indy in a directory of your choice. When unzipping, please keep the \source directory for the archive intact because that is used by some build batch files.
- In the source directory, there are several batch files. Run the appropriate one for your version of Indy:
 - FULLD4.BAT - Delphi 4
 - FULLD5.BAT - Delphi 5
 - FULLD6.BAT - Delphi 6
 - FULLD7.BAT - Delphi 7
- These batch files create subdirectories in the main Indy directory folder. They are (D4 for Delphi 4, D5 for Delphi 5, D6 for Delphi 6, and D7 for Delphi 7). These directories contain:
 - The Indy .DCU's
 - The Indy Design-Time .BPL
- In your Delphi IDE, add the Indy design-time package with *Component|Install Package...|Add...* Go to the subdirectory where the Indy .DCU's and Design-Time .BPL was placed by the batch file. Add the Design-Time .BPL that is listed. It usually is named dclIndy followed by the Borland Delphi version and an 0.
- Add the path where the .DCU's are located to your environment. Do this with *Tools|Environment Options...|Library...|Library Path...*

For Indy 10, there is an automated install included in the Indy Plus Package at <http://www.atozed.com/indy/> or if you prefer, you may build Indy 10 yourself.

4.2 How do I install Indy 8.0 in Delphi 6 Personal Edition?

Do the following:

1. Download the Indy 8.0 distribution at <http://www.indyproject.org/Sockets/Download/Files/Indy8.iwp> along with the updated *Delphi 6 Files*.
2. Run FULLD6.BAT from your Indy\Source directory
3. Place the DclIndy60.bpl in your Indy\D6 directory in the IDE (*Component\Install Packages\Add*).
4. Go to *Tools\Environment Options...|Library* and add your Indy\D6 directory to the Library Path.

4.3 In Indy 9.0, I am getting the error "Error on loading Winsock2 library (WS2_32.DLL)" and it only occurs on Win95 machines.

It's just that Indy 9.0 requires Winsock 2 and Winsock 2 is built into Windows 98, Windows Me, Windows NT 4.0, and Windows 2000. Winsock 2 was not included in Windows 95 but there is a free Winsock 2 upgrade. More information is available at http://www.microsoft.com/windows95/downloads/contents/WUAdminTools/S_WUNetworkingTools/W95Sockets2/Default.asp. This update is also included in some Windows 95 Dial-Up Networking, DUN updates. Your Windows 95 end-users should upgrade to Winsock 2 anyway because the original Winsock 1.1 version had some bugs and security problems that were addressed in the Winsock 2 update.

We decided to move towards Winsock 2 because a vast majority of computers are Windows 98, Me, NT 4.0, 2000, and XP (those operating system already include it). Furthermore, there is a freely available update for Windows 95 and Winsock 2 has an improved API.

4.4 When I try to compile my application that uses Indy it says it cannot find Indy.dcu or some other file?

You need to either compile using the Indy package, or add the Indy directory to your library path. This is not a Indy problem, this is just the way that Delphi / C++ Builder work.

4.5 When I attempt to setup Indy on a fresh copy of Delphi 4.0, it fails to compile?

Please update to the Delphi 4 Update Pack 2. This is available at <http://www.borland.com/devsupport/delphi/downloads/>. We recommend that most users upgrade to this version anyway as it fixes some problems in the initial Delphi 4 release.

4.6 When I try to compile Delphi says it cannot find SyncObjs?

Delphi 4 Standard does not include this file. For Indy 9.0, we are discontinuing support for Delphi 4 Standard Edition.

4.7 After installing Indy, I am getting an error message such as: "The procedure entry point @ldglobal@Fetch\$qqrr175System@AnsiStringxoxo could not be located in the dynamic link library Indy60.bpl"

You probably still have an older Indy .BPL on your system that's causing a conflict. Search through your computer, delete all copies of Indy60.bpl, and then rerun the FULLD6.BAT file as administrator.

4.8 After installing Indy, I get a "A device attached to the system is not functioning." when loading another package. What do I do about it?

This is a misleading Windows error message that means that a package which depended on Indy can not work with the new version of Indy.

If you have source-code for the Indy-dependent package, you will have to rebuild it. If you do not have the have the source-code, you have to get the vendor to rebuild it with the new version of Indy.

If this is happening with the Indy design-time package or with Indy 10, the protocols packages, you should search through your computer and remove all Indy binaries, rebuild Indy, and then reinstall it.

4.9 Why do you change the Indy API between major Indy versions?

We often change the API's, sometimes class names, or unit names between major version of Indy. We have read criticism of this and even some of us are not always keen on such changes. However, there are some reasons for changing the Indy API. Those are:

- Sometimes, inconsistencies in the API are resolved after we release a major Indy version. For example, in Indy 8.0, a username property was named UserID.
- Sometimes, we see some design issues when coding on a prior release of Indy and find that addressing the issue requires an API change. In the later stages of Indy development, we hesitate to make such changes in the API because it can cause unintended consequences. Thus, we slate those changes to the next release of Indy. This is very likely to happen because we have a time frame determined by forces outside of our control.
- Sometimes, we add new features to Indy and find limitations in the current Indy framework that make adding the feature difficult or impossible. As a result, we often have to change the API to better suit our needs for the new feature.

4.10 How hard is it to upgrade my code to the latest version of Indy?

Most of your Indy 8.0 based code can be ported to Indy 9.0. However, some classes were given some major restructuring. For example:

- The coders were completely redesigned and rewritten.
- The DNS resolver was cleaned up significantly and some class names were changed in the process.
- The SocksInfo class was made into it's own component. You use this with a TIdSocketHandler that you link to from the client component using the IOHandler property.
- The SSL Intercepts in Indy 8.0 were made into IOHandlers for Indy 9.0 so your code will have to change.

For porting your code to the current Indy 10 code, we have started keeping notes in a Compatibility.txt file in the main VCS.

5 Borland C++ Builder Installation

5.1 How do I install Indy into Borland C++ Builder?

- Close the Borland C++ Builder IDE if it is open.
- Temporarily copy **IDPAS32.obj** to a temporary directory. You do not want to delete that file because it is NOT part of the Indy package.
- Remove all Indy files including **dclindy*.bpl**, **dclindy.***, **dclindy*.bpl**, **indy*.lib**, **id*.pas**, **id*.hpp**, **id*.obj**, and **id*.dcu**. Take care that you only remove the old Indy files and not something else. Be

sure that you also remove any **indy*.bpl** from your Windows\System32 directory.

- Restore **IDPAS32.obj** from the temporary directory you created earlier to the directory where it was originally located.
- Remove all old Indy Help files, **indy.***, from the Help subdirectory and replace them with the new ones.
- Place the new version of Indy in a directory of your choice. When unzipping, please keep the \source directory for the archive intact because that is used by some build batch files.
- In the source directory, there are several batch files. Run the appropriate one for your version of Borland C++ Builder:
 - FULLC4.BAT - Borland C++ Builder 4
 - FULLC5.BAT - Borland C++ Builder 5
 - FULLC6.BAT - Borland C++ Builder 6
- These batch files create subdirectories in the main Indy directory folder. They are (C4 for C++Builder 4, C5 for C++Builder 5, and C6 for C++Builder 6). These directories contain:
 - The Indy .DCU files
 - The Indy .HPP files
 - The Indy .LIB file
 - The Indy .OBJ files
 - The Indy Design-Time .BPL
- Open the Borland C++ Builder IDE.
- In your C++Builder IDE, add the Indy design-time package with *Component\Install Package...|Add...* Go to the subdirectory where the Indy .DCU's and Design-Time .BPL was placed by the batch file. Add the Design-Time .BPL that is listed. It usually is named **dclIndy** followed by the Borland C++ Builder version and an 0.
- Add the path where the .DCU's are located to your environment. Do this with *Tools\Environment Options...|Library...|Library Path...*
- Add the path where the .HPP's are located to your project. Do this with *Project\Options\Directories\Conditionals\Include path*. Check the box **Default** and click **OK** to save this setting.

5.2 I tried to install Indy 8.0 in Borland C++ Builder and got missing package errors - the files seem to listed okay in the Indy source directory.

An old Indy 8.0 installer fails to compile the appropriate binaries for C++Builder. To compile Indy, we suggest using either:

- Running FullC4.bat for C++Builder 4 or FullC5.bat for C++Builder 5. These batch files are in your source directories.
- Compiling the IndyC40.BPK and DclIndyC40.BPK for C++Builder 4 or Compiling the IndyC50.BPK and DclIndyC50.BPK for C++Builder 5.

Starting with Indy 9.0, we discontinued our installer and we have [discontinued the Indy Borland C++ Builder BPK](#) files.

5.3 In Indy 9.0, there is no IndyC60.bpk. What do I do to get Indy to work with C++Builder 6?

Run FULLC6.BAT from your Indy\Source directory and that will build the C++Builder 6 binaries.

We decided to discontinue supporting the C++Builder .BPK's. We realize that using BPK's is more intuitive than using the batch files and ideally, we would have liked to include the Borland C++Builder .BPK's. Unfortunately, there are a several issues involved and those are discussed below:

- The Indy install needed to be relatively small. Building with the .BPK's would be rather inefficient when you consider that the run-time .BPL is the same for Delphi 4 and C++Builder 4, Delphi 5 and C++Builder 5, and Delphi 6 and C++Builder 6. Interestingly enough, C++Builder 6 has the exact same RTL library as Delphi 6 Service Pack 2.
- We had to do something unusual in the Indy packages by putting a run-time unit (IdAntiFreeze) into the design-time package instead of the run-time package so the run-time package would only depend on the RTL and not the visual Component Library. In Borland Delphi, this is not a problem if the IdAntiFreeze was included and because that unit has no design-time code in it at all. Unfortunately, this is a problem in C++Builder because you depend upon a .HPP and .OBJ file for that unit and in C++Builder, those are not generated for design-time packages at all. For C++Builder, we have to trick the Delphi compiler (that's included in C++Builder) into generating the .HPP and .OBJ files for IdAntiFreeze using a dummy unit named IdDummyUnit.
- We never did use the IndyC*.bpk packages at all when building the Indy binaries for installations. We were simply the Delphi .dpk's along with some special parameters that cause DCC32 to generate .OBJ and .HPP files and then making a .LIB file. Now just so there's no misunderstanding, you can get C++Builder binaries from Delphi .dpk's and we know of two vendors that do this successfully. The processed for building the .BPK's with MAKE is really just the same thing because MAKE is just a batch processor that calls the Delphi compiler with the exact parameters we are using and having TLIB add the objects to the .LIB files.

6 Exceptions

6.1 Why does Indy raise exceptions as part of its normal operation?

All exceptions are **not** errors. Exceptions are exactly that - exceptions. Delphi / C++ Builder use exceptions to handle errors in an elegant way. However exceptions have other uses besides errors as well.

EAbort is one example of an exception that is not necessarily an error. Exceptions such as these are used to modify standard program flow and communicate information to a higher calling level where they are trapped.

Indy uses exceptions in such a way as well. There are two primary places that Indy does so and are addressed in the following two questions:

- [Why do I get 10038 exceptions raised when I shut down my servers?](#)^[7]
- [Why do I keep getting EldConnClosedGracefully exceptions?](#)^[7]

The 10038 exception for clients is an error and for servers sometimes is too. However the EldConnClosedGracefully for servers is trapped and handled. While EldConnClosedGracefully can be an error it is often a "normal" condition. Because of this, it descends from EldSilentException. You can add EldSilentException to your debugger options to tell the debugger not to stop on these exceptions. The exception will still occur and your EXE will still function the same, however the debugger just wont stop on them for debugging.

6.2 Why do I keep getting EldConnClosedGracefully exceptions?

For Servers

The client has disconnected the socket normally and this exception is used to notify the server handling code. From the IDE just hit F9 to resume program execution and Indy will catch and handle the exception.

For Clients

The server side of this connection has disconnected normally but your client has attempted to read or write to the connection. You should trap this error using a try..except or in C++ Builder, a try..catch block.

There is more detail about the EldConnClosedGracefully exception in an article at the [Indy Portal](#) titled [Connection Closed Gracefully](#).

Please also see the help file for possible further information.

See also:

- [Why does Indy raise exceptions as part of its normal operation?](#)^[6]
- [Why do I get 10038 exceptions raised when I shut down my servers?](#)^[7]

6.3 Why do I get 10038 exceptions raised when I shut down my servers?

This exception is normal and will be handled by Indy. Resume program execution by hitting F9 and Indy will catch the exception. In addition, this will only happen in the Delphi / C++ Builder IDE (and only if you have stop on exception turned on) and not a stand alone executable.

NOTE: Code has been changed so that this appears less often than before, but it still can occur and is normal .

See also:

- [Why does Indy raise exceptions as part of its normal operation?](#)^[6]
- [Why do I keep getting EldConnClosedGracefully exceptions?](#)^[7]

6.4 When I connect with TIdLPR or TIdRSH, I get a `EIdSocketException` saying "Socket Error # 10048 Address already in use." Why?

The reason you get an address already in use is because after a local port has been used, it goes into a `FD_WAIT` state. This `FD_WAIT` state is intended to give the stack enough time to negotiate the final TCP/IP connection close sequence without interference from other connections as well as preventing the close sequence from interfering with other connections. While in `FD_WAIT`, you can not use that particular local port and address combination.

For `TIdLPR` and `TIdRSH`, we force the client to bind to a local port within a specific range before connecting to a server because these protocols require the client to use specific local port ranges when making a connection to a server. This is done with the `TIdTCPClient.BoundPortMin` and `TIdTCPClient.BoundPortMax` properties.

Usually, if you do this with a specific IP address while a local port is in `FD_WAIT` state, the bind fails and `Indy` will then try to bind to the next port. Unfortunately, when using the wildcard IP address (0.0.0.0), the bind will succeed while a port is in a `FD_WAIT` state but when you connect, you get an "Address already in use" error.

The only workarounds available are:

- Wait a minute for the local port to get out of `FD_WAIT` state.
- Set the `TIdTCPClient.BoundIP` property to the to the machine's current local IP address. This workaround can be problematic if a machine has more than one local IP address and you do not know which one to use.

For most clients, the best practice is to let the stack select any available local port because most servers do not care what local port the client is using and because of the issue we mentioned earlier. Do not use the `TIdTCPClient.BoundPort`, `TIdTCPClient.BoundPortMax`, and `TIdTCPClient.BoundPortMin` properties unless you have a very compelling reason to do so.

7 Socket Programming (Winsock)

7.1 What is the difference between blocking and non-blocking sockets?

Read "Introduction to Indy" for a good explanation - <http://www.swissdelphicenter.ch/en/showarticle.php?id=4>

7.2 Where can I find Socket error messages explanations, like Socket error 10057?

The error codes are listed in `Winsock.pas` as constants. A full error descriptions can be found at http://www.sockets.com/a_c.htm.

7.3 What is an "RFC" and where do I find them?

RFC is an acronym for Request for Comments and official documents from the Internet Engineering Task Force (IETF) with an unlimited distribution. RFC's are numbered in a series and are referred to by numbers.

RFC's contain information for the Internet Community including users and system administrators as well as developers. Some RFC's describe standardized protocols such as FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), IMAP4 (Internet Message Access Protocol 4), and DNS (Domain Name Service). Sometimes, a protocol is updated by a RFC.

FTP is based mostly on RFC 959 but is updated and supplemented by

RFC 1639	FTP Operation Over Big Address Records (FOOBAR)
RFC 2428	FTP Extensions for IPv6 and NATs
RFC 2389	Feature negotiation mechanism for the File Transfer Protocol
RFC 2228	FTP Security Extensions

You can locate many RFC's at <http://www.rfc-editor.org/> and the standardization process is described in RFC 2026.

We should also note that sometimes Indy also follows Internet Drafts. These are working documents are considered works in progress and some drafts do not become standards. We do this because some features described in the Draft are in common usage or provide a valuable feature. You can search for Internet Drafts at <http://www.ietf.org/>.

Occasionally, Indy developers will sometimes implement features and protocols which are not described in RFC's or drafts because we see that something is in common usage and demand and either is in a specification which is readily available or appears to be a *de facto* standard. Such things include Gopher+ support, YEnc encoding in Indy 10, RSH, Rexec, and UUEncoding.

8 Firewalls and Proxies

8.1 How do I use Indy with a Proxy?

The answer depends upon the type of proxy you need to use so you need to ask your administrator about this or consult your proxies documentation. Indy currently supports several different types of proxies. Indy 8, 9, and 10 support SOCKS (<http://www.socks.permeo.com/>) proxies, and HTTP proxies. Indy 9.0 added support for FTP proxies. Indy 10 also has HTTP Tunneling Proxy support as specified in *Tunneling TCP based protocols through Web proxy servers* (<http://www.web-cache.com/Writings/Internet-Drafts/draft-luotonen-web-proxy-tunneling-01.txt>).

See also:

- [How do I use a TIdFTP with a FTP Proxy?](#)^[10]
- [How do I use a TIdHTTP with a HTTP Proxy?](#)^[11]
- [How do I use a TIdFTP with a SOCKS proxy?](#)^[10]
- [How do I use a TIdTCPClient with a HTTP Tunneling Proxy?](#)^[12]
- [How do I use a TIdTCPClient with a SOCKS Proxy?](#)^[12]
- [How do I use a TIdUDPCClient with a SOCKS Proxy?](#)^[13]
- [How do I use Indy with a NAT \(Network Address Translator\)?](#)^[13]

8.2 How do I use a TIdFTP with a FTP Proxy?

For Indy 8.0

Indy 8.0's FTP client does not support FTP proxies so you have to upgrade to Indy 9.0 or Indy 10.

For Indy 9.0 and Indy 10

Ask your network administrator for information about using the FTP proxy. Indy 9 and 10 support the following types of proxies with the `TIdFTP.ProxySettings.ProxyType` property

- `fpcmNone` - No FTP Proxy
- `fpcmUserSite` - Send command `USER user@hostname`
- `fpcmSite` - Send command `SITE` (with logon)
- `fpcmOpen` - Send command `OPEN`
- `fpcmUserPass` - `USER user@firewalluser@hostname / PASS pass@firewallpass`
- `fpcmTransparent` - First use the `USER` and `PASS` command with the firewall username and password, and then with the target host username and password
- `fpcmUserHostFireWallID` - In Indy 10 only - `USER hostuserId@hostname firewallUsername`
- `fpcmNovellBorder` - In Indy 10 only - Novell BorderManager Proxy
- `fpcmCustomProxy` - In Indy 10 only - Use the `OnCustomFTPProxy` event to provide custom proxy support by writing your own FTP proxy and server login sequence. When using this feature, we recommend that use the `TIdFTP.GetLoginPassword` method instead of `TIdFTP.Password` so your proxy login will also support One-Time Password system (OTP). We also recommend that you provide some end-user scripting in this event for a mass-deployed FTP program so users can write their own scripted sequences for odd configurations.

Then set the `TIdFTP.ProxySettings.Host`, `TIdFTP.ProxySettings.Port`, `TIdFTP.ProxySettings.Username`, and `TIdFTP.ProxySettings.Password` properties using the information provided by your network administrator.

8.3 How do I use a TIdFTP with a SOCKS proxy?

For Indy 8.0

In Delphi, do the following:

```
IdFTP1.SocksInfo.Host := [the DNS name of the proxy server or its IP address]
IdFTP1.SocksInfo.Protocol := [svNoSocks for no SOCKS support, svSocks4 for SOCKS 4, svSocks4A
for SOCKS 4 with protocol extension, and svSocks5 for SOCKS 5 protocol]
IdFTP1.SocksInfo.Authentication := [saNoAuthentication for no SOCKS authentication or
saUsernamePassword for username and password authentication]
IdFTP1.Passive := True;
```

In C++Builder, do the following:

```
IdFTP1->SocksInfo->Host = [the DNS name of the proxy server or its IP address]
IdFTP1->SocksInfo->Protocol = [svNoSocks for no SOCKS support, svSocks4 for SOCKS 4, svSocks4A
for SOCKS 4 with protocol extension, and svSocks5 for SOCKS 5 protocol]
IdFTP1->SocksInfo->Authentication = [saNoAuthentication for no SOCKS authentication or
saUsernamePassword for username and password authentication]
IdFTP1->Passive = True
```

For Indy 9.0

1. Create an `IOHandler` such as `TIdIOHandlerSocket` or `TIdSSLIOHandlerSocket` and assign the `TIdFTP.IOHandler` property this.
2. Set the `TIdFTP.Passive` property to true.
3. Create a `TIdSocksInfo` object and assign it to the `IOHandler's SocksInfo` property.

4. Set the TIdSocksInfo properties as required.

For Indy 10

1. Create an IOHandler such as TIdIOHandlerStack or SSL IOHandler and assign the TIdFTP.IOHandler property this.
2. Create a TIdSocksInfo object and assign it to the IOHandler's TransparentProxy property.
3. Set the TIdSocksInfo properties as required.
4. If there is another proxy that the SOCKS Proxy itself needs to use, you have to use, set the Chained Proxy to a TIdCustomTransparentProxy such as TIdSocksInfo for another socks Proxy or a TIdConnectThroughHttpProxy for a HTTP Tunneling proxy.
5. If you are not using another type of chained proxy, you may be able to use standard PORT transfers if your SOCKS proxy supports the bind method described in *SOCKS: A protocol for TCP proxy across firewalls* (<http://archive.socks.permeo.com/protocol/socks4.protocol>) and *RFC 1928 - SOCKS Protocol Version 5*. If your SOCKS proxy does not support that capability, you have to use PASV (Passive := True) transfers.

8.4 How do I use a TIdHTTP with a HTTP Proxy?

You will often find the values you for setting HTTP proxy information for your network in *Internet Explorer |Tools|Internet Options..|Connections|Lan Settings*. If you can find them there, ask your network administrator for this information.

For Indy 8.0

Set the the properties as described below:

In Delphi, do the following:

```
HTTP1.Request.ProxyServer := [the DNS name of the proxy server or its IP address]
HTTP1.Request.ProxyPort := [port the HTTP the HTTP proxy listens on]
HTTP1.Request.ProxyUsername := [username if the proxy requires authentication]
HTTP1.Request.ProxyPassword := [password if the proxy requires authentication]
```

In C++Builder, do the following:

```
HTTP1->Request->ProxyServer = [the DNS name of the proxy server or its IP address]
HTTP1->Request->ProxyPort = [the DNS name of the proxy server or its IP address]
HTTP1->Request->ProxyUsername = [username if the proxy requires authentication]
HTTP1->Request->ProxyPassword = [password if the proxy requires authentication]
```

For Indy 9.0 and Indy 10

```
HTTP1.ProxyParams.ProxyServer := [the DNS name of the proxy server or its IP address]
HTTP1.ProxyParams.ProxyPort := [port the HTTP the HTTP proxy listens on]
HTTP1.ProxyParams.ProxyUsername := [username if the proxy requires authentication]
HTTP1.ProxyParams.ProxyPassword := [password if the proxy requires authentication]
```

In C++Builder, do the following:

```
HTTP1->Request->ProxyServer = [the DNS name of the proxy server or its IP address]
HTTP1->Request->ProxyPort = [the DNS name of the proxy server or its IP address]
HTTP1->Request->ProxyUsername = [username if the proxy requires authentication]
HTTP1->Request->ProxyPassword = [password if the proxy requires authentication]
```

8.5 How do I use a TIdTCPClient with a HTTP Tunneling Proxy?

For Indy 8.0 and Indy 9.0

Both Indy 8.0 and Indy 9.0 do not support HTTP Tunneling proxies so you have to use Indy 10.

For Indy 10

1. Create an IOHandler such as TIdIOHandlerStack or SSL IOHandler and assign the TCPClient.IOHandler property this.
2. Create a TIdCustomTransparentProxy object and assign it to the IOHandler's TransparentProxy property.
3. Set the TIdCustomTransparentProxy properties as required.
4. If there is another proxy that the HTTP Tunneling Proxy itself needs to use, you have to use, set the Chained Proxy to a TIdCustomTransparentProxy such as TIdSocksInfo for a SOCKS Proxy or a TIdConnectThroughHttpProxy for another HTTP Tunneling Proxy.

For TIdFTP in Indy 10

You have to set the Passive property to true (PASV) to work with a HTTP Tunneling Proxy because standard FTP PORT transfers will not work in this configuration. You should allow the user to be able to choose between using PORT transfers (TIdFTP.Passive := False) and using PASV transfers (TIdFTP.Passive := True).

8.6 How do I use a TIdTCPClient with a SOCKS Proxy?

For Indy 8.0

In Delphi, do the following:

```
IdTCPClient1.SocksInfo.Host := [the DNS name of the proxy server or its IP address]
IdTCPClient1.SocksInfo.Protocol := [svNoSocks for no SOCKS support, svSocks4 for SOCKS 4,
svSocks4A for SOCKS 4 with protocol extension, and svSocks5 for SOCKS 5 protocol]
IdTCPClient1.SocksInfo.Authentication := [saNoAuthentication for no SOCKS authentication or
saUsernamePassword for username and password authentication]
```

In C++Builder, do the following:

```
IdTCPClient1->SocksInfo->Host = [the DNS name of the proxy server or its IP address]
IdTCPClient1->SocksInfo->Protocol = [svNoSocks for no SOCKS support, svSocks4 for SOCKS 4,
svSocks4A for SOCKS 4 with protocol extension, and svSocks5 for SOCKS 5 protocol]
IdTCPClient1->SocksInfo->Authentication = [saNoAuthentication for no SOCKS authentication or
saUsernamePassword for username and password authentication]
```

For Indy 9.0

1. Create an IOHandler such as TIdIOHandlerSocket or TIdSSLIOHandlerSocket and assign the TCPClient.IOHandler property this.
2. Create a TIdSocksInfo object and assign it to the IOHandler's SocksInfo property.
3. Set the TIdSocksInfo properties as required.

For Indy 10

1. Create an IOHandler such as TIdIOHandlerStack or SSL IOHandler and assign the TCPClient.IOHandler property this.
2. Create a TIdSocksInfo object and assign it to the IOHandler's TransparentProxy property.
3. Set the TIdSocksInfo properties as required.
4. If there is another proxy that the SOCKS Proxy itself needs to use, you have to use, set the Chained Proxy to a TIdCustomTransparentProxy such as TIdSocksInfo for another socks Proxy

or a `TidConnectThroughHttpProxy` for a HTTP Tunneling proxy.

See also:

- [How do I use a TidFTP with a SOCKS proxy?](#)^[10]

8.7 How do I use a TidUDPClient with a SOCKS Proxy?

Starting with Indy 10 build 51, we introduced the ability to use a SOCKS with UDP. However, this will only work with SOCKS protocol version 5 and the proxy server must support the UDP Associate method described in *RFC 1928 - SOCKS Protocol Version 5*.

1. Create a `TidSocksInfo` object and assign it to the `TidUDPClient.TransparentProxy` property.
2. Set the `TidSocksInfo` properties as required.
3. Call `TidUDPClient.OpenProxy` just before you use any receive or send calls.
4. After completing all of your receive and send calls, call `TidUDPClient.CloseProxy`.

or if you are using the optional `TidUDPClient.Connect` and `TidUDPClient.Disconnect` methods, you can simply:

1. Create a `TidSocksInfo` object and assign it to the `TidUDPClient.TransparentProxy` property.
2. Set the `TidSocksInfo` properties as required.
3. Call the `TidUDPClient.Connect` method and it will open a connection to your proxy and call `TidUDPClient.Disconnect` to close the connection to the proxy.

8.8 How do I use Indy with a NAT (Network Address Translator)?

Most of the time, Indy based clients should work transparently with a NAT meaning that there is special coding or special settings required.

However, there may be problems using [FTP with SSL](#)^[23] behind a NAT. This is discussed [here](#)^[14].

In other servers, you need to be careful that you never generate any output which contains an internal IP address because those addresses will not work outside of the internal network and a NAT may not recognize the IP address at all.

See also:

- [How do I use FTP with SSL behind a NAT?](#)^[14]
- [How do I use FTP with SSL?](#)^[23]

8.9 How do I use FTP with SSL behind a NAT?

The problem is that the FTP protocol is really not too friendly with many firewall/proxy configurations. The problem is that there's two connections between the client and server (the control connection and the data channel) and the data channel is established using commands (PORT and PASV) that communicate IP addresses. Computers behind a NAT have IP addresses that only make sense in the Internal network and not the Internet.

This is not usually a problem with unencrypted FTP because most NAT's will fix the connection by opening random ports and changing values being communicated in the PORT and PASV commands. Unfortunately, SSL makes it nearly impossible for a NAT to make such a correction.

For Indy 10 TIdFTP

A [FTP client using SSL](#) ^[23] should use PASV transfers (Passive := True) instead of PORT transfers (Passive = False) because a NAT can not translate the IP address given in a PORT command on the control connection.

If you must use PORT transfers, you should do the following:

1. Configure your NAT to forward a range ports to your computer behind the NAT.
2. Specify that port range with the DataPortMin and DataPortMax properties.
3. Set the ExternalIP property to the NAT's external Internet IP address.

You may also wish to consider clearing encryption from the control channel by setting the UseCCC property to True as a NAT might be able to intercept the IP address specified in the PORT command. However, that does not work on some NAT's, you may not want to send unencrypted FTP commands, and some FTP servers do not support the CCC command.

If you are making a FTP client that will be deployed widely, you should make configuration options for port ranges to use with data channels, permitting the user to enter an external Internet IP address for their NAT's, and permitting the user to clear the command connection with the CCC command.

For Indy 10 TIdFTPServer

We do not recommend placing a [FTP server using SSL](#) ^[23] behind a NAT at all. The NAT can not translate the IP address given as a reply to the PASV command on the control connection into an address that works outside of the internal network. Unfortunately, PASV transfers are the default in some FTP clients.

If you must use a FTP server using SSL behind a NAT, you should do the following:

1. Configure your NAT to forward a range ports to your server.
2. Specify that port range with PASVBoundPortMin and BoundPortMax properties.
3. Set the IP address given in PASV replies to your NAT's external Internet IP address using the OnPASVReply event but do not change the IP address for clients that are also on the internal network.

If you are writing a FTP server that will be deployed widely, you should make configuration options for port ranges, for specifying an external Internet IP address for connections from clients outside the network, and for permitting the CCC command. You can permit the CCC command globally by setting the TIdFTPServer.FTPSecurityOptions.PermitCCC to True or you can permit the CCC command on a per-user basis by setting UserSecurity.PermitCCC property in the TIdFTPServerContext object after the user logged in.

See also:

- [How do I use Indy with a NAT \(Network Address Translator\)?](#)^[13]
- [How do I use FTP with SSL?](#)^[23]

8.10 What should I do if I do not know what proxy a user will need to use and what the correct settings are?

This is actually a common problem where software is deployed on a wide scale which developers should consider addressing.

Be sure that your program can use all of various proxy support we offer and you should read all of our Knowledge Base items concerning using Indy with Firewall and Proxies.

The topics you should read are:

- [How do I use Indy with a Proxy?](#)^[9]
- [How do I use a TIdFTP with a FTP Proxy?](#)^[10]
- [How do I use a TIdFTP with a SOCKS proxy?](#)^[10]
- [How do I use a TIdHTTP with a HTTP Proxy?](#)^[11]
- [How do I use a TIdTCPClient with a HTTP Tunneling Proxy?](#)^[12]
- [How do I use a TIdTCPClient with a SOCKS Proxy?](#)^[12]
- [How do I use a TIdUDPClient with a SOCKS Proxy?](#)^[13]
- [How do I use Indy with a NAT \(Network Address Translator\)?](#)^[13]
- [How do I use FTP with SSL behind a NAT?](#)^[14]

For TCP Clients, we recommend that you enable SOCKS proxy support and if you are using Indy 10, be sure that you offer configuration settings for using a HTTP Tunneling Proxy. If you are using TIdHTTP, be sure that you offer your users settings for using a HTTP proxy. If you are using functionality in TIdFTP, you should offer your users configuration settings for FTP proxies and you should permit a user to choose between using PORT (TIdFTP.Passive = False) and PASV (TIdFTP.Passive = True) data transfers. For UDP clients in Indy 10 build 51, you should offer your users configuration settings for SOCKS version 5 Proxies. Your program defaults should be for users who do not require proxy support at all because most users on dial-up PPP Internet Service Provider Accounts, some network configurations, and usually, NAT's do not require any special settings unless you are using [FTP with SSL behind a NAT](#)^[14].

The key is making the options available through easily edited configuration files, command-line switches, or configuration dialog-boxes where settings are saved in configuration files or the registry in Win32. Just be careful that you do not intimidate users with all sorts of proxy/firewall options initially in a configuration dialog box or offer them in an initial program configuration wizard. You may also wish to state in your documentation and configuration dialog-boxes that if the user does not know what their proxy configuraiton is that they should contact their administrators or have the administrator configure the proxy settings in the program. It is also helpful to tell the user that if they are using a dial-up PPP Internet Service Provider account, that they probably should just leave your program with the default settings.

9 Server

9.1 How can I send a record using Indy?

Check out the "SendReceiveRecords" demo.

9.2 How can I send a file using Indy?

Check out the ImageServer and TCPDataset demos in Indy.

9.3 Is it safe to access a database connection in a TIdTCPServer.OnExecute event?

Since the server triggers the OnExecute event in the context of a server connection thread, you need to maintain an individual TSession for each thread. However, you need to select a database driver which is thread-safe meaning it can be used for concurrent access in several threads. Some database drivers may **not** be thread safe so please ask the database engine vendor about this before using it.

If you are using third-party BDE replacements, you need to consult your documentation or ask the vendor if their product is thread-safe and follow their instructions for using the component in multi-threaded servers.

9.4 How do I get the Peer's IP Address in the TIdTCPServer?

For Indy 8.0

Use code such as this:

```
procedure TfrmMain.TCPServerConnect(AThread: TIdPeerThread);
var PeerIP : String;
begin
    PeerIP := Thread.Connection.Binding.PeerIP;
end;
```

For Indy 9.0

Use code such as this:

```
uses IdIOHandlerSocket;

procedure TfrmMain.TCPServerConnect(AThread: TIdPeerThread);
var PeerIP : String;
begin
    PeerIP := TIdIOHandlerSocket(AThread.Connection.IOHandler).Binding.PeerIP;
end;
```

For Indy 10.0

Use code such as this:

```
uses IdContext, IdIOHandlerSocket;

procedure TfrmMain.TCPServerConnect(AContext: TIdContext);
var PeerIP : String;
begin
    PeerIP := AContext.Connection.Socket.Binding.PeerIP;
end;
```

10 Ping, ICMP (Internet Control Message Protocol), Raw Sockets

10.1 I can not ping with Indy on Windows NT or Windows 2000. What do I do about this?

In Windows NT and Windows 2000, Indy's TIdICMPClient requires administrative privileges to use raw sockets. The command line PING application does not exhibit this behavior because Microsoft chose to use undocumented calls in ICMP.DLL and its winsock implementation to get around the issue. In short: they cheated. Indy does not use ICMP.DLL to maintain Linux compatibility, and allow use of both Raw sockets and ICMP sockets.

In single-user operating systems such as Windows 95, 98, and ME, any user can ping with Indy.

10.2 I can not ping with Indy using Linux. What do I do about this?

In Linux, Indy's TIdICMPClient requires root privileges to use raw sockets. The command-line ping can be used by most Linux users because the SUID bit is set and the binary is owned by "root".

Before telling you how to set this properly, we should point out that any executable file with the SUID bit set would cause the file to be executed with the privileges of the person that owns the file. Because the file is owned by "root", the SUID bit would cause the program to have absolute privileges over everything and thus, could potentially cause unlimited damage to a system (be it accidentally or on purpose) and nothing would stop this. Many troublemakers will try to abuse such programs to gain root privileges for themselves. Thus, those types of programs have to be designed with extreme caution and security in mind. Typically, this involves taking a lot of care with the code, reducing dependencies on external libraries, keeping the amount of code to a minimum, and verifying all input provided by a user.

Often, many programs can do things without root privileges simply by using executing standard SUID programs and by piping input and output to that SUID program.

Please think very carefully about all of this. If you still have to make your program SUID, do the following as root:

```
chown root.root myprogram
chmod u+s myprogram
```

11 SMTP (Simple Mail Transport Protocol) and E-Mail

11.1 How can I send / receive HTML email?

This is shown in MailClient demo in Indy and Remy Lebeau made an [entry in his blog](#) about this.

11.2 How can I send or receive mail attachments?

See the MailClient Indy demo.

11.3 Whenever I do a LoadFromFile or a LoadFromStream, I get an EldReadTimeout exception. What am I doing wrong?

Your message files need to be properly terminated for the Indy LoadFromFile and LoadFromStream methods. You must terminate the message with a "<CR><LF>.<CR><LF>". The Indy helpfile notes this behavior.

Both LoadFromFile and LoadFromStream are designed to handle messages written using the SaveToStream or SavedToFile methods in the Indy TIdMessage component. These methods create an encoded version of the message contents that matches the protocol transmissions used when the message is transmitted using TIdSMTP including the message terminator characters (CRLF.CRLF). If you are trying to import .EML message files from MS Outlook Express, or another email client, make sure that the text file contains the message terminator sequence. An exception will be raised if LoadFromFile or LoadFromStream are used to load files that do not include the message terminator sequence.

11.4 How do I send mail directly to a recipients mail server?

To send mail directly to another SMTP server, you do the following:

1. Do a MX (Mail eXchange) DNS lookup on the domain name.
2. Connect to the machine address returned by the MX records you consulted.
3. Submit the E-Mail using the SMTP protocol.

In Indy 8 and 9, the SMTPRelay Indy demo illustrates how to do this using Indy.

In Indy 10, you may wish to use the TIdSMTPRelay component.

Some people will have problems with sending mail directly to a recipient's mail server and thus, would have to use their ISP's mail server. Those problems are:

- Some Internet Service Providers firewall users at port 25 to make it impossible to send E-Mail directly to a recipient's server. This is a security measure that Internet Service Providers take so users can not bypass SMTP server logging that Internet Service Providers uses to trace users that spam.
- The IP address being used by a PPP dial-up user may be blacklisted. This means that the a SMTP server using the blacklist may refuse to accept E-Mail from that machine.
- Many Internet users do not know the address of their DNS server. Often, this information is provided by the Internet Service Provider when the user logs into their PPP dial-up account. Some ISP's also provide users with some type of automated setup utility that automatically configures the appropriate settings without the user knowing what the settings are.

11.5 How do I send emails through a SMTP server that require authentication?

Before answering this question, we do well to state that Authentication was never in the original SMTP protocol. When RFC 821 was written in 1982, third-party SMTP servers and gateways had to relay messages between other hosts and the other hosts would act as an intermediaries between the originating SMTP servers and the intended destination. That was how E-Mail worked at the time. As time went on, the need for intermediate SMTP servers to relay messages between an originating server and a destination server faded and as the internet opened up to the general public, the capability to relay messages from a client on a different network to a destination outside of the SMTP server's network only became a security hazard that was exploited by spammers and others out to abuse E-Mail.

There are several ways that the security threat was addressed.

- The SMTP server was configured to simply send mail to other networks that originated only from specific I.P. Addresses in the server's network.
- The SMTP servers would require the client sending the outgoing E-Mail to authenticate by logging into the POP3 (Post Office Protocol 3) server and immediately afterwards, send the E-Mail to the SMTP server.
- The third-way is by adding some extensions to the SMTP protocol that would use various Simple Authentication and Security Layer (SASL) mechanisms to ensure that the client is authorized to send outgoing E-Mail and these extensions and SASL mechanisms were formalized into standards.

Since the first two methods are relatively straightforward, we will only discuss how to use Indy's RFC 2554 support.

For Indy 8.0 and 9.0

Indy's TIdSMTP component has only limited RFC 2554 support in that it can only support one SASL mechanism called LOGIN. There are other SASL mechanisms that can be used such as CRAM-MD5, Digest MD5, Kerberos 4, GSSAPI (Generic Security Service Application Program Interface), and S/Key. Those SASL mechanisms are not supported in that version.

To use Indy's LOGIN SASL support, simply set the TIdSMTP UserName property to the username, and password property to the user's account password, and the AuthenticationType property to atLogin. Then send the E-Mail as you normally would. You can also use Indy to determine if LOGIN authentication support is available on the SMTP server and using it if it is available by using the AuthSchemesSupported property with code like this:

```
IdSMTP1.AuthenticationType := atNone;
IdSMTP1.Connect;
try if
  IdSMTP1.AuthSchemesSupported.IndexOf('LOGIN')>-1 then
  begin
    IdSMTP1.AuthenticationType := atLogin;
    IdSMTP1.Authenticate;
  end;
  IdSMTP1.Send(IdMessage1);
finally
  IdSMTP1.Disconnect;
end;
```

For Indy 10

You can use simple AUTH LOGIN similarly to Indy 9.0. Set TIdSMTP.AuthType to atDefault. Then set the Username and password properties in TIdSMTP.

This provided for backwards compatibility. For many programs which are distributed to the public or on a mass scale, we recommend that you use Indy 10's SASL framework for authentication so your program can support a wide variety of SMTP servers. Our expanded SASL framework supports TIdSMTP, TIdPOP3, and TIdIMAP4. This framework has plug-in SASL mechanism components for CRAM (Challenge-Response Authentication Mechanism) MD5 SASL (RFC 2195), Anonymous SASL (RFC 2245), External SASL (RFC 2222), Login, One-Time-Only-Password - OTP (RFC 2444), Plain SASL (2595), and S/Key (RFC 2222). You can also write your own SASL mechanism component by writing a descendent of TIdSASL (located in IdSASL). To use this with framework with TIdSMTP:

1. Drop a SASL mechanism component on your form. Those components are located on the new *Indy SASL* tab in the component palette.
2. In TIdSMTP, set the AuthType property to atSASL.
3. In TIdSMTP, click the button beside the SASLMechanisms property. An editor will appear.
4. Select the SASL mechanism components you wish to use with TIdSMTP from the *available* listbox.
5. Click the Right arrow to move those to the *Assigned* listbox.
6. Click Ok when you are finished.

11.6 I am using the TIdSMTP but when I try to send mail I get an error message such as: "521 potserver.organization.com access denied" or "572 Relay Not Authorized."

It means that you have been denied access to that SMTP server by its administrator. Administrators prevent abuses of their SMTP relay servers by requiring:

- That you send E-Mail only from a computer connected to their network.
- That you use some type of [authentication](#) ^[19].

Talk to the server's administrator and ask what are the requirements for connecting to the server. Sometimes, the server may need to be reconfigured.

11.7 I am using the TIdSMTP but when I try to send mail I get an error message such as: "530 Must issue a STARTTLS command first"

This message means that the SMTP server requires you to use TLS when authenticating and sending E-Mail. The server expects you to use explicit TLS SMTP. For this, you connect to port 25, issue an EHLO command and get a reply, send a STARTTLS command, and after getting a success reply, you then negotiate TLS with the server.

For Indy 8 and 9

Indy 8 and Indy 9 do not support explicit TLS with SMTP. If the SMTP server supports implicit TLS with SMTP, you may able to use implicit TLS with SMTP by stting the TIdSMTP.IOHandler to TIdSSLIOHandlerSocket and the port to the IdPORT_ssmtip constant in IdAssignedNumbers (465).

For Indy 10

Indy 10 supports SMTP with explicit TLS. In the Indy 10's TIdSMTP, you simply have to assign a TIdSSLIOHandlerSocketBase descendent to the TIdSMTP.IOHandler property and set the UseTLS property to either utUseRequireTLS if you do not want to use an unencrypted connection or utUseExplicitTLS if you are willing to use an unencrypted connection.

11.8 In C++Builder, a class methods such as TIdSMTP.QuickSend require an extra parameter. What parameter should I pass?

C++Builder requires that you pass the TMetaClass as the first parameter even if the actual code ignores that parameter. Thus, Delphi code such as:

```
TIdSMTP.QuickSend('Host', 'Subject', 'Recipient', 'From', 'Message');
```

in C++Builder is:

```
TIdSMTP::QuickSend(__classid(TIdSMTP), "Host", "Subject", "Recipient", "From", "Message");
```

11.9 Where can I find examples for using the SMTP / POP3 server components?

Demos for both SMTP and POP3 servers are available in the Indy Demos. They are however, rudimentary examples of implementing a limited set of functionality. Please consult the RFCs for SMTP (RFC 821) and POP3 (RFC 1939) for more information related to complete server implementations.

11.10 Why doesn't the MAPS Demo work anymore?

The MAPS (Mail Abuse Prevention System) demonstration program no longer works because MAPS has become a paid subscription service that is only available to subscribers. More information about this service is at <http://www.mail-abuse.org/>. We have decided to discontinue the MAPS demonstration program for that reason and because several other blacklists are no longer in service.

11.11 A SMTP server is treating E-Mail from my program as if it was spam or the server rejects it with a 554 error. I don't have any problems with Outlook. How do I fix this?

The reason this is occurring is because some spam filtering software is screening E-Mail for the "X-library:" header. This header is sent by Indy 8.0 and the release versions of Indy 9.0. In and of itself, this is not bad. However, some malware such as spam bulk E-Mails and even a worm (W32/Nicehello@MM), were built with Indy meaning that they sent the same "X-Library:" header that most Indy-based E-Mail programs do.

For this reason, we have decided to remove the X-Library header from Indy 9.0 code and you can obtain it from our VCS. Instructions are at:

<http://www.indyproject.org/download/DevSnapshot.html>

If you are using Indy 8.0 and you can not upgrade to this code, you can remove the X-Library header by editing IdMessage and removing line 439 which reads:

```
Values['X-Library'] := gsIdProductName + ' ' + gsIdVersion ; {do not localize}
```

and then rebuild Indy using the FULL*.BAT file for your system.

11.12 But that still does not help with SpamAssassin. What now?

Some filters such as [SpamAssassin](#) have other tests they use for filtering E-Mail and a score is generated for a particular message. Administrators then use a given score as a "cut-off" when filtering E-Mail spam. Unfortunately, in the process of filtering E-Mail from certain spamware programs, SpamAssassin is also filtering out E-Mail from any Indy-based program (a false positive).

One test SpamAssassin uses is detecting the pattern in the MIME message part boundary which the Indy code has generated. In Indy 10, we have addressed this by making the boundary more random.

In addition, we have made sure that the from header will always contain a name if the name property is blank. We have also forced the In-Reply-To header to always be generated. This helps reduce the score SpamAssassin assigns. We have recently sent E-Mail and received a Spam Assassin score of 0.

Please keep in mind that these aren't the only test SpamAssassin does. There are some other tests involving the content of the E-Mail, some header values, some blacklists such as MAPS, and other filtering technologies. Some legitimate E-Mail messages can generate false-positives with those tests.

12 SSL (Secure Sockets Layer) or TLS (Transport Layer Security)

12.1 Can I legally use and/or export software that uses Indy's SSL capabilities?

It depends upon the laws in your country. Some countries outlaw the use of strong encryption while other nations regulate or have no restrictions at all on strong encryption.

We do not give you any inherent right to use SSL in your applications. Indy allows you to do SSL by providing the code, but we do not provide a license to use SSL. If you use SSL, it is up to you to make sure that you are legally permitted to use it in the country you are in and for your application.

12.2 Where can I download Indy's SSL support .DLL's and source-code?

Because of export restrictions we do not provide Indy's SSL support DLL's and code for Indy. But they are available at Intelicom in Slovenia <http://www.intelicom.si/?newlang=eng>. Select the Download link.

12.3 I need SSL help, where can I find answers?

There is a SSL forum hosted by Intelicom at <http://www.intelicom.si/forum/list.php?f=3>.

12.4 How do I use FTP with SSL?

In Indy 8.0 and Indy 9.0, you can not do this. The File Transfer Protocol requires some extensions that those Indy versions do not support them. The extensions provide encrypted or clear PORT and PASV data channels plus provide a command for starting negotiation with explicit TLS. If you need to use SSL with FTP, you need the current Indy 10 source-code which is still in the alpha stage of development.

For Indy 9.0

A commercial SSL implementation called [SecureBlackbox](#) includes a FTP client with TLS. We do not endorse or support this product.

For Indy 10

In Indy 10, both the FTP client and server both already supports SSL.

In the Indy 10's TIdFTP, you simply have to assign a TIdSSLIOHandlerSocketBase descendant to the TIdFTP.IOHandler property and set specific properties for SSL. Those properties are:

DataPortProtection - set to clear (ftpdpsClear) for unencrypted clear data transfers (PBSZ 0 and PROT C) or set to private (ftpdpsPrivate) for encrypted TLS data transfers (PBSZ 0 and PROT P).

Set UseTLS to one of the following values:

- utNoTLSSupport - you do not wish to use TLS at all
- utUseImplicitTLS - You wish to use [implicit TLS](#)^[24]. In implicit TLS FTP, you connect to port 990, start TLS negotiation, and the entire session is encrypted. IETF has deprecated implicit TLS FTP so this setting should only be used with FTP servers that do not yet support [explicit TLS](#)^[24].
- utUseRequireTLS - You wish your session to use [explicit TLS](#)^[24]. In explicit TLS FTP, you connect to port 21 like the unencrypted version of FTP, issue a special command (AUTH TLS, AUTH SSL, AUTH TLS-P, or AUTH TLS-C), start TLS negotiation, and encryption lasts until you disconnect or issue the reinitialize command (REIN). If UseTLS is utUseRequireTLS, no attempt is made to continue the FTP session if TLS negotiation fails. This setting is for situations where security is more important than interoperability.
- utUseExplicitTLS - You wish your session to use [explicit TLS](#)^[24] if the FTP server supports it.

Indy 10's TIdFTPServer supports both either [implicit TLS](#)^[24] FTP or [explicit TLS](#)^[24] FTP. Assign the IOHandler property to a TIdServerIOHandlerSSLBase descendant and then set the UseTLS property to one of these values:

- utNoTLSSupport - you do not want your server to use TLS at all
- utUseImplicitTLS - You want your server to use [implicit TLS](#)^[24]. In [implicit TLS](#)^[24] FTP, you connect to port 990, start TLS negotiation, and the entire session is encrypted. IETF has deprecated [implicit TLS](#)^[24] FTP so this setting should only be used with older FTP clients that do not yet support [explicit TLS](#)^[24].
- utUseRequireTLS - You want your server to require the client to use [explicit TLS](#)^[24] by not accepting unencrypted usernames and passwords. The FTP Server will disconnect a client that attempts to if the client tries to authenticate without encryption. This setting is for situations where security is more important than interoperability.
- utUseExplicitTLS - You want your server to support [explicit TLS](#)^[24] and also accept unencrypted logins. This setting is for situations where interoperability may be more important than security.

If you would like more information about TLS FTP, please consult the current IETF (Internet Engineering Task Force) working draft document titled [Securing FTP with TLS](#).

See also:

- [What is the difference between implicit TLS and explicit TLS?](#)^[24]
- [How do I use FTP with SSL behind a NAT?](#)^[14]

12.5 What is the difference between implicit TLS and explicit TLS?

These are two different approaches to TLS securing protocols.

In implicit TLS, the following is done:

1. A client connects to a different port than for the regular non-TLS version of the protocol. For example, the non-encrypted POP3 clients use port 110. For implicit TLS with POP3, you connect to port 995.
2. You immediately negotiate TLS with the server.
3. Except for the TLS encryption, your session is exactly the same as a regular POP3 session.
4. The TLS encryption is used throughout the entire connection.

Often implicit versions of protocols are known as POP3S (or SPOP3), SNEWS, HTTPS, and FTPS (or FTSPS).

In explicit TLS, the following is done:

1. You connect to the POP3 server on port 110 exactly as you would with non-encrypted POP3.
2. You issue a command to the server indicating that you wish to being TLS negotiation. In POP3, you issue a STLS command and get an +OK reply.
3. You then negotiate TLS with the server.
4. The TLS often lasts throughout an entire session. In [explicit TLS FTP](#)^[23], a REIN will reset the TLS encryption and the control channel returns to a non-encrypted state.

Currently, the IETF (Internet Engineering Task Force) favors explicit TLS protocols because most implicit TLS protocols require a separate port and the reserved port numbers (1-1024) have already been allocated for various protocols. There are some implicit TLS protocols in use today such as SNEWS and there are still software implementations based on implicit TLS protocols that are still in use today even though explicit TLS protocol versions are available.

See also:

- [How do I use FTP with SSL?](#)^[23]

13 FTP (File Transfer Protocol)

13.1 With some servers, when I try to abort a transfer, the client is automatically disconnected from the server, instead of aborting just the transfer. Is this an Indy bug?

It probably is not an Indy bug at all. Some servers will close the control connection when the data port connection is closed unexpectedly.

13.2 How do I obtain the size of a file I am about to download with TIdFTP?

Use the TIdFTP.Size method passing the file name as a parameter. That method will return the size of a file or -1 if the server does not support the SIZE command.

You should never use any data from the TIdFTP.DirectoryListing collection because, on some servers, the most you can only get is an estimate of the file size because the file system is based on records or blocks instead of individual bytes. You then might be able to estimate a file size with that information. Sometimes, you can not even calculate an estimate of the file size from that information in the directory listing. Some FTP directory list formats do not include any file size information at all. Even when emulating Unix, some mainframes may only return estimated file sizes.

13.3 I want to download files only if they have been modified on the server by comparing the date and time. How do I do this with Indy?

The best way to compare the date and time is to use the MDTM FTP command. This FTP command will give the last modified date stamp based on Universal time. This will only work if the FTP server supports the MDTM command.

Indy 9.0 does not support MDTM although the functionality is easy to add. The MDTM command is documented in the an IETF draft titled [Extensions to FTP](#).

Indy 10 has support for MDTM in the TIdFTP.FileDate method.

There are some unreliable ways that files are sometimes synchronized between a server and client. We do not recommend these at all and mention them only to discuss their problems.

Some clients might attempt to use a date and time given by the server in the output from the list command. This is problematic for several reasons.

- Many servers will give timestamps relative to their local time-zone. Unfortunately, you may always know what that time zone is for comparisons and if you did know, you would have to convert the time into the user's local time-zone for accuracy. The only list formats that give time-stamps based on Universal Time (in the U.S., commonly known as Greenwich Mean Time) are EPLF (Easily Parsed List Format), and Distinct FTP Server plus the output from the MLST or MLSL commands.
- Novell Netware and Unix (including systems emulating the Unix directory listing format) will either report either a year or a time but not both. Novell Netware only reports a time for files less than one year old. Most Unix servers only give a time stamp when a file is less than 6 months old and unfortunately, this behavior has to be emulated by a majority of FTP servers in use today just to work with web-browsers and a vast majority of FTP code. Serv-U, some FreeBSD systems, and Indy 10's FTP Server component support a -T switch for getting a complete timestamp with the LIST command but that is still based on the local time-zone, is not supported by most FTP servers, and other Unix systems may treat the -T switch far differently than FreeBSD.
- A few FTP servers may not give a date in their FTP directory listings at all. This includes Xercom MicroRTOS, HP3000, Cisco IOS, TOPS20, IBM MVS JES interfaces, and VSE Power Queues and VSE root directories.
- There are still many servers in use that do not support the MLST and MLSL commands.

Some clients might use file sizes given by the directory listing to determine if a file was modified. This is problematic for the reasons given in the Knowledge Base item, "[How do I obtain the size of a file I am about to download with TIdFTP?](#)"²⁵

13.4 In TIdFTP, is there a method to parse the contents of the List result, so I can use a ListView component to display the directory?

The original FTP RFC 959 never standardized a common directory format for the LIST command. A FTP server was free to list files in whatever format it wanted. It often is dependant on the platform the computer is running on although some FTP servers will emulate Unix.

For Indy 8.0

In Indy 8.0, you have to write your own parsing routines for different types of FTP servers. We recommend that you upgrade to Indy 9.0 because that Indy version has directory parsing support for Unix and MS-DOS.

If you only want file or directory names, you can use the Details = False parameter such as:

```
IdFTP1.List(MyStrings, '', False);
```

In Indy 9.0

Most of the time, you can get a directory listing by using the TIdFTP.DirectoryListing property to access a collection of TIdFTPListItem objects. Please see the Indy Help File documentation for TIdFTPListItem. The Indy 9.0 directory parsing code only supports Unix and MS-DOS directory list formats. This is not as bad as it seems because many FTP servers emulate Unix including some modern mainframes, MacIntosh servers (e.g. AppleShare IP FTP Server, NetPresenz), current AMOS servers, and most freeware/shareware Windows FTP Servers. You may have to upgrade to the current Indy 10 development versions or write your own parsing code if you need more capability.

In Indy 10 Development Versions

In the current Indy development versions, we have added parsing code for the following servers:

- Bull GCOS 7 or Bull DPS 7000
- Bull GCOS 8 or Bull DPS 9000/TA200
- Cisco IOS
- Distinct FTP Server
- EPLF ([Easily Parsed List Format](#))
- HellSoft FTP Server for Novell Netware 3 and 4
- HP 3000 or MPE/iX including HP 3000 with Posix
- IBM AS/400, OS/400
- IBM MVS, OS/390, z/OS
- IBM OS/2
- IBM VM, z/VM
- IBM VSE
- KA9Q or NOS
- Microware OS-9
- Music (Multi-User System for Interactive Computing)
- NCSA FTP Server for MS-DOS (CUTCP)
- Novell Netware
- Novell Netware Print Services for UNIX
- TOPS20
- UniTree
- VMS or VMS (including Multinet, MadGoat, UCX)
- Wind River VxWorks

- WinQVT/Net 3.98.15
- Xecom MicroRTOS

and starting with Indy 10 build 51, support has been added for:

- BayNetworks WfFTP
- Tandem NonStop Guardian
- Stratus VOS
- Netmanage Chameleon
- SuperTCP
- NCSA FTP Server for MacIntosh
- Unisys ClearPath Enterprise Server (OS2200 or MCP)
- PC-NFSD

Even with this code, you still may encounter servers that give directory listings that Indy does not support so you may still have to write your own code and this new code is far from perfect because sometimes, we could only find scant data to work with. On the other hand, most FTP servers developed today use the Unix System FTP list format and many servers are Unix-like systems such as Linux, HP UX, FreeBSD, Solaris, and AIX.

The MLST and MLSD commands described in the IETF Draft, *Extensions to FTP* ([draft-ietf-ftpext-mlst-16](#)) require a standardized listing format. While some servers support them, there are still many servers still do not support this at all and in one case, the data is not as good as the information from the standard LIST command. To enable TIdFTP to use the MLSD command, set the UseMLIS property to true and TIdFTP will use those commands if the server supports them.

13.5 When using the TIdFTP.SetModTime function, why do I get a "no such file or directory" but the file exists.

The TIdFTP.SetModTime uses a FTP command called MDTM. That command requests the last modified date on a file and this is what *Extensions to FTP* ([draft-ietf-ftpext-mlst-16](#)) says. Unfortunately, some servers and some clients also use (I should say abuse), that command to set the last modified file date. Indy's FTP client and server do support this abuse of the MDTM for compatibility reasons. If the server does support a "MFMT" command, Indy will use that instead of MDTM but most servers do not support that command.

The syntax for getting a file date is:

```
MDTM MyFileName
```

and to set the file date, the syntax usually is something like:

```
MDTM filedate MyFileName
```

If a server doesn't support setting the file date with MDTM, the server may interpret the filedate as the start of the filename. Since a file with such a name probably does not exist, you get that error about a file not existing.

So all I can say is that server does NOT support setting the file date with the MDTM command. The confusion is the consequences of someone else's poor design.

The ultimate solution to this problem is if more servers support the commands from *The 'MFxx' Command Extensions for FTP* ([draft-somers-ftp-mfxx-01](#)) IETF Internet Draft. That draft also provides a solution to another problem I see with FTP but that's another blog entry I may make someday.

14 HTTP (Hypertext Transfer Protocol)

14.1 I am getting a 403 Forbidden error with some sites while they work fine with Opera, Netscape Communicator, or Internet Explorer. How do I fix this?

This happening because some administrators are now filtering HTTP requests from programs that send the HTTP User-Agent header with the value "Mozilla/3.0 (compatible; Indy Library)". This is the default value provided by TIdHTTP. Unfortunately, some people used TIdHTTP in their malware (denial of service programs, E-Mail address web-harvesters, password cracking programs. and ill-behaved web-robots). To counter this, some administrators block such programs from accessing their website using the HTTP User-Agent header value meaning that not only will they block those malware programs but other Indy based applications which send the default value.

We can not change this default value because we provided it for a historical reason. At one time, Netscape Navigator was extremely popular and had some high-end features such as Java, Java Script, and frames as a well as a few proprietary HTML tags. Some webmasters started filtering HTTP requests from non-Netscape web-browsers because they couldn't work on their websites. To bypass this restriction, Microsoft Internet Explorer would send "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)" and other programs started sending similar strings. When doing Indy 8.0, we were well aware of this and created a default user agent value of "Mozilla/3.0 (compatible; Indy Library)".

You need to change the TIdHTTP.Request.UserAgent property from the default value to your own value. When doing this, the string should start with the text "Mozilla/3.0 (compatible; " followed by the name of your program, followed by a ")".

15 IP Multicasting

15.1 TTIdIPMCastClient and TIdIPMCastServer do not work on Linux?

You may have to enable multicast support in the Linux Kernel. To enable multicast routing on linux, a route needs to be added manually.

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0 //for first nic
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth1 //for second nic
```

16 Migrating to Indy 10

16.1 Where is the Intercept property?

The Intercept property has been moved to the IOHandler.

16.2 TIdFTP.DirectoryListing is returning nothing for my FTP server. Why?

We restructured the way FTP directory listing is done. The code is now a plug in system with various parsers in their own units which register themselves into a framework. To add support for a particular type of FTP server, you simply add the parser unit to one of your program unit's uses clause.

To add the Unix directory list parser, add `IdFTPListParseUnix` to your unit's clause. To add support for Microsoft Windows NT IIS FTP server, add `IdFTPListParseWindowsNT` to your uses clause. You can also add many other parsers we include in the Indy packages. By convention, they are named *IdFTPListParser* followed by the server type. You do not need to add `IdFTPListParseBase` to your uses clause because `IdFTP` uses that unit. If you wish to use all of Indy's FTP List parser classes, just add the `IdAllFTPListParsers` unit to your uses clause.

We admit this new design is not as intuitive as the older one but this has many advantages for us such as:

- The FTP list parsing code is now much easier to maintain than the same code in a non-modular design in one unit which would be at least 150KB total.
- You now can add support for only the FTP servers you want your program support to support.
- You can now write your own custom parser classes and plug those into the standard FTP parsing list framework.

16.3 Where is the TIdPOP3.APOP property?

The property is now:

`AuthenticationType`

For `TIdPOP3`, that has three values and those are:

- `atAPOP` - use APOP
- `atUserPass` - use the classic POP3 USER PASS commands
- `atSASL` - use one of the SASL mechanism components that is listed in the `SASLMechanisms` property. The mechanisms will be tried in the order set by the property editor.

17 Miscellaneous Topics

17.1 Why did the chicken cross the road?

Some people say, "To get to the other side." We however are not really sure why the chicken crossed the road. It shall remain a mystery to the world.

18 For Further Reading

18.1 Indy Introductory Articles

Title	Description	Link
Introduction to Indy by Chad Z. Hower	Indy uses blocking socket calls. Blocking calls are much like reading and writing to a file.	http://www.swissdelphicenter.ch/en/showarticle.php?id=4
Introduction to Sockets by Chad Z. Hower	This article is an introduction to socket (TCP/IP sockets) concepts. It's not meant to be a complete coverage of all socket topics; it's meant as a primer to educate the reader at a level at which socket programming can be easily communicated	http://www.swissdelphicenter.ch/en/showarticle.php?id=3
Overview of Internet Direct by Chad Z. Hower	Full contents of Chapter 20 of "Building Kylix Applications" by Cary Jensen. Chapters by Chad Z. Hower a.k.a "Kudzu". While this article was written for Kylix, it also applies to Windows.	http://www.swissdelphicenter.ch/en/showarticle.php?id=5
Using Internet Direct by Chad Z. Hower	Full contents of Chapter 21 of "Building Kylix Applications" by Cary Jensen. Chapters by Chad Z. Hower a.k.a "Kudzu". While this article was written for Kylix, it also applies to Windows.	http://www.swissdelphicenter.ch/en/showarticle.php?id=5
Indy in Depth by Chad Z. Hower and Hadi Hariri	Indy in Depth is an e-book written by the Indy experts themselves. Indy in Depth covers from the very basic introduction to sockets, all the way up to advanced usage. Whether you are an absolute beginner and have never programmed sockets before, or you are an advanced socket developer, Indy in Depth has material suited for you.	http://www.atozed.com/indy/book/

18.2 Socket Programming

Title	Description	Link
Appendix C: Error Reference by Bob Quinn and Dave Shute	This is a hypertext version of Appendix C in the text Windows Sockets Network Programming, by Bob Quinn and Dave Shute. It contains a detailed description of all the errors defined by the WinSock specification, with suggestions on how to deal with them when they occur in an application.	http://www.sockets.com/a_c.htm
Introduction to Sockets by Chad Z. Hower	This article is an introduction to socket (TCP/IP sockets) concepts. It's not meant to be a complete coverage of all socket topics; it's meant as a primer to educate the reader at a level at which socket programming can be easily communicated	http://www.swissdelphicenter.ch/en/showarticle.php?id=3
What is TCP/IP? by Shiv Kumar	What is TCP/IP?	http://www.matlus.com/scripts/website.dll/Tutorials?DelphiTCPIP&TCPIPWhats&1
What is TCP/IP? Part 1 by Shiv Kumar	Part 1- Understanding Client / Server	http://www.matlus.com/scripts/website.dll/Tutorials?DelphiTCPIP&FirstTCPIP&2
What is TCP/IP? Part 2 by Shiv Kumar	Part 2 - Building the telnet client for the server	http://www.matlus.com/scripts/website.dll/Tutorials?DelphiTCPIP&TCPIPSimple&3
What is TCP/IP? Part 3 by Shiv Kumar	Part 3 - Building the Server for the Telnet Client	http://www.matlus.com/scripts/website.dll/Tutorials?DelphiTCPIP&TCPIPDataClient&4

18.3 RFCs (Request for Comments) and Internet Drafts

Title	Description	Link
The Internet Engineering Task Force	Most current Internet Drafts are located here.	http://www.ietf.org/
RFC Editor: Homepage	This has a nice searchable engine for the RFC's.	http://www.rfc-editor.org/

18.4 Indy Exceptions

Title	Description	Link
Connection Closed Gracefully by Chad Z. Hower	What is the EIdConnClosedGracefully exception? Why do I see this exception in my code? Many Indy users are annoyed by the EIdConnClosedGracefully exception that is raised with Indy servers, especially the HTTP and other servers. EIdConnClosedGracefully is an exception signaling that the connection has been closed by the other side intentionally.	http://www.swissdelphicenter.ch/en/showarticle.php?id=1

18.5 HTTP (Hypertext Transfer Protocol)

Title	Description	Link
Automatic file version check over the web (example program) by Corbin Dunn	A complete source code example of how to check for the latest version of a program over the web using the Indy socket components. Simple and easy example.	http://codecentral.borland.com/codecentral/ccweb.exe/listing?id=17521
Uploading Files using Indy (multipart/form-data) by Shiv Kumar	In this tutorial, I'll show you how you can use Indy to upload files to a Web Server (either an ISAPI or ASP) that can accept file uploads using the W3C specified multipart/form-data encoding type.	http://www.matlus.com/scripts/website.dll/Tutorials?DelphiTCPIP&IndyMultiPartFormData&5

18.6 SSL (Secure Sockets Layer) or TLS (Transport Layer Security)

Title	Description	Link
IndySSL - using certificate authentication - Client side issues by Gregor Ibic	This document explains how to use certificate authentication when connecting to a site that requires certificate authentication.	http://www.intelicom.si/domc.ms.nsf/web/en.products001.html

18.7 Web Services

Title	Descriptions	Links
Building a stand-alone Web service with Indy by Dave Nottage	This article shows how to build a Web service using Indy and Delphi 6.	http://community.borland.com/article/0,1410,27513,00.htm!

18.8 Other Indy Websites

AToZed Software maintains Indy Portal at <http://www.atozed.com/indy/>.

Several Indy developers have their own Indy blogs at Indy Core Blogspace (<http://www.indyproject.org/Sockets/Blogs/index.iwp>).

Index

- 1 -

10038 Exception 6, 7
10048 Exception 8

- 5 -

521 Error 20
530 Error 20
554 Error 21
572 Error 20

- A -

A device attached to the system is not functioning 4
Abort 24
Access Denied 20
Address already in use 8
Administrative Privileges 17
Anonymous SASL 19
API Changes 4
APOP3 29
associate 13
associate method 13
atDefault 19
atLogin 19
atSASL 19
Attachments 17
AUTH SSL 24
AUTH TLS 24
AUTH TLS-C 24
AUTH TLS-P 24
Automatic file version check over the web 32

- B -

bind 10, 8
bind method 10
Binding 16
Binding.PeerIP 16
blacklist 18, 21
Blocking Socket 8

blogs 33
Borland C++ Builder 5
 BPK 6
 Class Methods 21
 Installation 4
Borland C++ Builder Installation 4
Borland Delphi
 Installation 2
Borland Delphi 4
 Service Pack 2 3
 Standard Edition 3
Borland Delphi 6
 PE 3
 Personal Edition 3
Borland Delphi Installation 2
BoundIP 8
BoundPort 8
BoundPortMin 8
BPK 6
Building a stand-alone Web service with Indy 33
Building Kylix Applications
 Chapter 20 30
 Chapter 21 30

- C -

CCC 14
CCC Command 14
Challenge-Response Authentication Mechanism
 MD5 19
Chicken
 Cross the Road 29
Compatibility.txt 4
Connection Closed Gracefully 32, 7
CRAM
 MD5 19

- D -

Database
 Connection 16
Database Connection 16
DataPortMax 14
DataPortMin 14
Demo 1
Demo Playground 1
denial of service programs 28

Digest MD5 SASL 19
 DNS 18, 9
 DNS Resolver 4
 Domain Name Service 9
 Domain Name System 18
 Domain Name System Resolver 4
 draft-ietf-ftpevt-mlst-16 26, 27
 draft-somers-ftp-mfxx-01 27

- E -

Easily Parsed LIST Format 26
 EldConnClosedGracefully 32, 6, 7
 EldSilentException 6
 E-Mail
 Attachments 17
 Direct 18
 HTML 17
 HyperText Markup Language 17
 Server 21
 E-Mail address web-harvesters 28
 EPLF 26
 Error Reference 31
 Exception
 10038 Exception 7
 10048 Exception 8
 EldConnClosedGracefully 32, 6, 7
 EldSilentException 6
 explicit TLS 24
 explicit TLS FTP 20, 23
 explicit Transport Layer Security 20, 23, 24
 Export 22
 Extensions to FTP 25, 26, 27
 External SASL 19
 ExternalIP 14

- F -

false-positive 22
 FD_WAIT 8
 File Size 25
 File Transfer Protocol 9
 Abort 24
 AUTH SSL 23, 24
 AUTH TLS 23, 24
 AUTH TLS-C 23, 24
 AUTH TLS-P 23, 24

Directory Format 26
 Directory Listing 25, 26, 29
 File Date 25
 File Size 25
 File Synchronization 25
 LIST 25, 26
 MDTM 25, 27
 MFMT 27
 MLSL 25, 26
 MLST 25, 26
 PASV 10, 12, 13, 14, 15
 PBSZ 23
 PORT 10, 12, 13, 14, 15
 PROT 23
 Proxy 10
 REIN 23, 24
 Secure Socket Layer 13, 14, 23
 SIZE 25
 SSL 13, 14, 23
 TLS 13, 14, 23
 Transport Layer Security 13, 14, 23
 File Transfer Protocol Proxy 10, 15
 fpcmCustomProxy 10
 fpcmNone 10
 fpcmNovellBorder 10
 fpcmOpen 10
 fpcmSite 10
 fpcmTransparent 10
 fpcmUserHostFireWallID 10
 fpcmUserPass 10
 fpcmUserSite 10
 FreeBSD 25
 From 22
 header 22
 From header 22
 FTP 13, 14, 9
 Abort 24
 AUTH SSL 23, 24
 AUTH TLS 23, 24
 AUTH TLS-C 23, 24
 AUTH TLS-P 23, 24
 CCC 14
 data channel 23
 Directory Format 26
 Directory Listing 25, 26, 29
 File Date 25
 File Size 25
 File Synchronization 25

FTP 13, 14, 9
 LIST 25, 26
 MDTM 25, 27
 MFMT 27
 MLSD 25, 26
 MLST 25, 26
 PASV 10, 12, 13, 14, 15, 23
 PBSZ 23
 PORT 10, 12, 13, 14, 15, 23
 PROT 23
 Proxy 10
 REIN 23, 24
 Secure Socket Layer 13, 14
 SIZE 25
 SSL 13, 14
 TLS 13, 14, 23
 Transport Layer Security 13, 14, 23
 FTP Proxy 10, 15, 9
 ftpdpsClear 23
 ftpdpsPrivate 23
 FTPS 24
 FTPSecurityOptions.PermitCCC 14
 FULD6.BAT 3
 FULLC4.BAT 4
 FULLC5.BAT 4
 FULLC6.BAT 4
 FULLD4.BAT 2
 FULLD5.BAT 2
 FULLD6.BAT 2
 FULLD7.BAT 2

- G -

Generic Security Service Application Program
 Interface 19
 GMT 25
 Gopher+ 9
 Greenwich Mean Time 25
 GSSAPI 19
 GSSAPI SASL 19

- H -

header 22
 from 22
 In-Reply-To 22
 How Can I Help 1

HTML 17
 HTTP 11, 32
 Proxy 11
 User-Agent header 28
 HTTP Proxy 11, 15, 9
 HTTP Tunneling Proxy 10, 12, 15, 9
 HTTPS 24, 32
 HyperText Markup Language 17
 Hypertext Transfer Protocol 15, 32
 Proxy 11
 User-Agent header 28
 Hypertext Transfer Protocol Proxy 11, 15
 Hypertext Transfer Protocol Tunneling Proxy 15

- I -

ICMP 17
 IdMessage 21
 IdPORT_ssmtp 20
 IdSASL 19
 IETF 23, 24, 25, 26, 27, 9
 IMAP4 19, 9
 implicit TLS 24
 Implicit TLS FTP 20, 23
 implicit Transport Layer Security 20, 23, 24
 Indy
 API 4
 blogs 33
 Commercial Programs 1
 Copyright 1
 License 1
 Shareware 1
 Support 1
 Teams 1
 Technical Support 1
 Indy 10 28
 Indy API 4
 Indy Core Blogspace 33
 Indy Knowledge Base
 Maintainer 1
 Indy SASL tab 19
 Indy.dcu 3
 IndyC60.bpk 6
 IndySSL - using certificate authentication - Client side
 issues 32
 In-Reply-To
 header 22
 In-Reply-To header 22

Installation

- Borland C+ Builder 4

- Borland Delphi 2

- Intelicom 22

- Intercept 28

- Internet Control Message Protocol 17

- Internet Drafts 31, 9

- Internet Engineering Task Force 23, 24, 25, 26, 27, 31, 9

- Internet Message Access Protocol 4 19, 9

- Introduction to Indy 30

- Introduction to Sockets 30, 31

- IOHandler 20, 23

- IOHandler property 4

- K -

- Kerberos 19

- Kerberos SASL 19

- L -

- Lebeau

- Remy 17

- Legality 22

- Line Printer Deamon 8

- Linux

- Multicast Routing 28

- Ping 17

- Root Privileges 17

- SUID bit 17

- SUID programs 17

- LIST 26

- LoadFromFile 18

- LoadFromStream 18

- Login SASL 19

- LPD 8

- LPR 8

- M -

- Mail Abuse Prevention System 21, 22

- Mail Exchange 18

- malware 21, 22, 28

- MAPS 21, 22

- MDTM 25

- message terminator sequence 18

MIME

- message part boundary 22

- MIME message part boundary 22

- Missing Package Errors 5

- MLSD 26

- MLST 26

- Mugaas

- J. Peter 1

- Multipurpose Internet Mail Extension message part boundary 22

- Multipurpose Internet Mail Extensions 22

- Must issue a STARTTLS command first 20

- MX 18

- N -

- NAT 13, 14, 15

- Network Address Translator 13, 14, 15

- Non-Blocking Socket 8

- O -

- OnConnect 14

- One-Time Password 10

- One-Time-Only-Password 19

- One-Time-Only-Password SASL 19

- OTP 10, 19

- OTP SASL 19

- Overview of Internet Direct 30

- P -

- packages 4

- password cracking programs 28

- PASV 10, 12, 13, 14, 15, 23

- PASVBoundPortMax 14

- PASVBoundPortMin 14

- PBSZ 23

- Peer IP Address 16

- PermitCCC 14

- Ping 17

- Linux 17

- Win32 17

- Windows 95 17

- Windows 98 17

- Windows ME 17

- Windows NT 17

Plain SASL 19
POP3 19, 21, 24, 9
 Server 21
 STLS 24
POP3 Server 21
POP3S 24
PORT 10, 12, 13, 14, 15, 23
Post Office Protocol 3 19, 24, 9
 Server 21
 STLS 24
Procedure Entry Point 3
PROT 23
Proxy
 File Transfer Protocol 10, 15, 9
 FTP 10, 15, 9
 HTTP 11, 15, 9
 HTTP Tunneling 12, 15, 9
 HyperText Transfer Protocol 15, 9
 SOCKS 10, 12, 13, 15, 9

- R -

Raw Sockets 17
REIN 23, 24
Relay Not Authorized 20
Remote Execute 9
Remote Shell 8, 9
Request For Comment 31, 9
Rexec 9
RFC 31, 9
RFC 1928 10, 13
RFC 1928 - SOCKS Protocol Version 5 10, 13
RFC 959 26
RFC 959 - FILE TRANSFER PROTOCOL (FTP)
26
RFC Editor: Homepage 31
robots 28
Root Privileges 17
RSH 8, 9

- S -

S/Key 19
S/Key SASL 19
saNoAuthentication 10
SASL 19, 29
 Anonymous 19

Challenge-Response Authentication Mechanism
19
CRAM MD5 19
Digest MD5 19
External 19
Generic Security Service Application Program
Interface 19
GSSAPI 19
Kerberos 19
Login 19
One-Time-Only-Password 19
OTP 19
Plain 19
S/Key 19
saUsernamePasswor 10
SavedToFile 18
SaveToStream 18
score 22
Secure Socket Layer 20, 22
 DLL 22
 Export 22
 File Transfer Protocol 13, 14, 23
 FTP 13, 14, 23
 Legality 22
 Simple Mail Transfer Protocol 20
 SMTP 20
 Source-Code 22
 Support 22
Secure Socket Layer Support
 DLL 22
 Source-Code 22
Secure Socket Layer Support DLL 22
Secure Sockets Layer 32, 4
SecureBlackbox 23
Securing FTP with TLS 24
Send
 File 16
 Record 15
Serv-U 25
SetModTime 27
Shareware 1
Simple Authentication and Security Layer 19, 29
 Anonymous 19
 Challenge-Response Authentication Mechanism
 19
 Digest MD5 19
 External 19
 Generic Security Service Application Program
 Interface 19

- Simple Authentication and Security Layer 19, 29
 - GSSAPI 19
 - Login 19
 - One-Time-only Password 19
 - OTP 19
 - Plain 19
 - S/Key 19
 - Simple Mail Transfer Protocol 18, 9
 - 521 Error 20
 - 530 Error 20
 - 554 Error 21
 - 572 Error 20
 - Access Denied 20
 - Authentication 19
 - Must issue a STARTTLS command first 20
 - Relay Not Authorized 20
 - Secure Socket Layer 20
 - Server 21
 - SSL 20
 - STARTTLS 20
 - TLS 20
 - Transport Layer Security 20
 - SIZE 25
 - SMTP 18, 9
 - Access Denied 20
 - Must issue a STARTTLS command first 20
 - 521 Error 20
 - 530 Error 20
 - 554 Error 21
 - 572 Error 20
 - Authentication 19
 - Relay Not Authorized 20
 - Server 21
 - STARTTLS 20
 - TLS 20
 - Transport Layer Security 20
 - SMTP Server 21
 - SNEWS 24
 - Socket
 - Blocking 8
 - Error Codes 8
 - Non-Blocking 8
 - Socket Error Codes 8
 - SOCKS 10, 12, 13
 - Associate method 13
 - Bind method 12
 - SOCKS associate method 13
 - SOCKS bind method 10
 - SOCKS Proxy 10, 12, 13, 15, 9
 - SOCKS: A protocol for TCP proxy across firewalls 10
 - spam 18, 19, 21, 22
 - spam filtering 21, 22
 - SpamAssassin 22
 - SPOP3 24
 - SSL 13, 14, 20, 32, 4
 - Export 22
 - File Transfer Protocol 13, 14, 23
 - FTP 23
 - Legality 22
 - Simple Mail Transfer Protocol 20
 - SMTP 20
 - Support 22
 - Support DLL 22
 - Support Source-Code 22
 - SSL Support
 - DLL 22
 - Source-Code 22
 - STARTTLS 20
 - Simple Mail Transfer Protocol 20
 - SMTP 20
 - SUID bit 17
 - SUID programs 17
 - svNoSocks 10
 - svSocks4 10
 - svSocks4A 10
 - svSocks5 10
 - SyncObjs.pas 3
- T -**
- TCP 12, 15, 16
 - The 'MFxx' Command Extensions for FTP 27
 - TIdConnectThroughHttpProxy 10, 12
 - ChainedProxy 12
 - TIdConnectThroughHttpProxy.ChainedProxy 12
 - TIdContext 16
 - TIdContext.Socket 16
 - TIdContext.Socket.Binding.PeerIP 16
 - TIdDNSResolver 4
 - TIdFTP 10, 12, 15, 23
 - Abort 24
 - DataPortProtection 23
 - DirectoryListing 25, 26, 29
 - FileDate 25
 - GetLoginPassword 10

- TIdFTP 10, 12, 15, 23
 - OnCustomFTPProxy 10
 - Passive 10, 12, 13, 14, 15
 - Password 10
 - ProxySettings 10
 - ProxySettings.Host 10
 - ProxySettings.Password 10
 - ProxySettings.Port 10
 - ProxySettings.ProxyType 10
 - ProxySettings.Username 10
 - SetModTime 27
 - Size 25
 - UseMLIS 26
 - UseTLS 23
- TIdFTP.Abort 24
- TIdFTP.DataPortMax 14
- TIdFTP.DataPortMin 14
- TIdFTP.DataPortProtection 23
- TIdFTP.DirectoryListing 25, 29
- TIdFTP.ExternallP 14
- TIdFTP.FileDate 25
- TIdFTP.GetLoginPassword 10
- TIdFTP.IOHandler 23
- TIdFTP.OnCustomFTPProxy 10
- TIdFTP.Passive 10, 12, 13, 14, 15
- TIdFTP.Password 10
- TIdFTP.ProxySettings 10
 - Host 10
 - Password 10
 - Port 10
 - ProxyType 10
 - Username 10
- TIdFTP.ProxySettings.Host 10
- TIdFTP.ProxySettings.Password 10
- TIdFTP.ProxySettings.Port 10
- TIdFTP.ProxySettings.ProxyType 10
- TIdFTP.ProxySettings.Username 10
- TIdFTP.SetModTime 27
- TIdFTP.Size 25
- TIdFTP.UseTLS 23
- TIdFTPSecurityOptions.PermitCCC 14
- TIdFTPServer 14
- TIdFTPServer.FTPSecurityOptions.PermitCCC 14
- TIdFTPServer.OnConnect 14
- TIdFTPServer.PASVBoundPortMax 14
- TIdFTPServer.PASVBoundPortMin 14
- TIdFTPServerContext 14
- TIdFTPServerContext.UserSecurity.PermitCCC 14
- TIdHTTP 15, 28
 - ProxyParams.ProxyPassword 11
 - ProxyParams.ProxyPort 11
 - ProxyParams.ProxyServer 11
 - ProxyParams.ProxyUsername 11
 - Request 28
 - Request.ProxyPassword 11
 - Request.ProxyPort 11
 - Request.ProxyServer 11
 - Request.ProxyUsername 11
 - Request.UserAgent 28
 - UserAgent 28
- TIdHTTP.ProxyParams.ProxyPassword 11
- TIdHTTP.ProxyParams.ProxyPort 11
- TIdHTTP.ProxyParams.ProxyServer 11
- TIdHTTP.ProxyParams.ProxyUsername 11
- TIdHTTP.Request.ProxyPassword 11
- TIdHTTP.Request.ProxyPort 11
- TIdHTTP.Request.ProxyServer 11
- TIdHTTP.Request.ProxyUsername 11
- TIdHTTP.Request.UserAgent 28
- TIdICMPClient 17
- TIdIMAP4 19
- TIdIOHandler 10, 12, 4
 - SocksInfo 10, 12
 - TransparentProxy 10, 12
- TIdIOHandler.SocksInfo 10, 12
- TIdIOHandler.TransparentProxy 10, 12
- TIdIOHandlerSocket 10, 12
 - SocksInfo 10, 12
- TIdIOHandlerSocket.SocksInfo 10, 12
- TIdIOHandlerStack 10, 12
 - TransparentProxy 10, 12
- TIdIOHandlerStack.TransparentProxy 10, 12
- TIdIPMCastServer 28
- TIdLPR 8
- TIdMessage
 - LoadFromFile 18
 - LoadFromStream 18
 - SavedToFile 18
 - SaveToStream 18
- TIdMessage.LoadFromFile 18
- TIdMessage.LoadFromStream 18
- TIdMessage.SavedToFile 18
- TIdMessage.SaveToStream 18
- TIdPeerThread 16
- TIdPeerThread.Connection 16
- TIdPeerThread.Connection.Binding 16

- TIdPeerThread.Connection.Binding.PeerIP 16
- TIdPOP3 19, 29
- TIdPOP3.APOP 29
- TIdPOP3.SASLMechanisms 29
- TIdPOP3Server 21
- TIdRSH 8
- TIdSASL 19
- TIdSASLList 19, 29
- TIdSMTP 19, 20
 - Access Denied 20
 - AuthenticationType 19
 - AuthSchemesSupported 19
 - AuthType 19
 - Must issue a STARTTLS command first 20
 - QuickSend 21
 - Relay Not Authorized 20
 - SASLMechanisms 19
 - UseTLS 20
- TIdSMTP.AuthenticationType 19
- TIdSMTP.AuthSchemesSupported 19
- TIdSMTP.AuthType 19
- TIdSMTP.IOHandler 20
- TIdSMTP.QuickSend 21
- TIdSMTP.SASLMechanisms 19
- TIdSMTP.UseTLS 20
- TIdSMTPRelay 18
- TIdSMTPServer 21
- TIdSocksInfo 10, 12, 13, 4
- TIdSSLIOHandlerSocket 10, 12, 20
 - SocksInfo 10, 12
- TIdSSLIOHandlerSocket.SocksInfo 10, 12
- TIdSSLIOHandlerSocketBase 20, 23
- TIdTCPClient 12
 - BoundIP 8
 - BoundPort 8
 - BoundPortMax 8
 - BoundPortMin 8
 - SocksInfo 12
- TIdTCPClient.BoundIP 8
- TIdTCPClient.BoundPort 8
- TIdTCPClient.BoundPortMax 8
- TIdTCPClient.BoundPortMin 8
- TIdTCPClient.IOHandler 12
- TIdTCPClient.SocksInfo 12
- TIdTCPServer 16
 - OnExecute 16
 - Peer IP Address 16
- TIdTCPServer.OnExecute 16
- TIdUDPCClient 13
 - TransparentProxy 13
- TIdUDPCClient.CloseProxy 13
- TIdUDPCClient.Connect 13
- TIdUDPCClient.Disconnect 13
- TIdUDPCClient.OpenProxy 13
- TIdUDPCClient.TransparentProxy 13
- Time-Zone 25
- TLS 13, 14, 20, 23, 32
 - explicit 24
 - Export 22
 - File Transfer Protocol 13, 14, 23
 - FTP 23
 - implicit 24
 - Legality 22
 - Simple Mail Transfer Protocol 20
 - SMTP 20
 - Support 22
 - Support DLL 22
 - Support Source-Code 22
- TLS FTP 13, 14
 - explicit 23
 - implicit 23
- TLS SMTP
 - explicit 20
 - implicit 20
- TLS Support
 - DLL 22
 - Source-Code 22
- Transmission Control Protocol 12, 15, 16
- Transport Layer Security 20, 32
 - DLL 22
 - explicit 20, 23, 24
 - Export 22
 - File Transfer Protocol 13, 14, 23
 - FTP 13, 14
 - implicit 20, 23, 24
 - Legality 22
 - Simple Mail Transfer Protocol 20
 - SMTP 20
 - Source-Code 22
 - Support 22
- Transport Layer Security Support 22
 - Source-Code 22
- Transport Layer Security Support DLL 22
- TTIdIPMCastClient 28
- Tunneling TCP based protocols through Web proxy servers 9

- U -

UDP 13, 15
Universal Time 25
Uploading Files using Indy 32
User Datagram Protocol 13, 15
UserSecurity.PermitCCC 14
UseTLS 20, 23
UT 25
utNoTLSSupport 20, 23
utUseExplicitTLS 20, 23
utUseImplicitTLS 20, 23
utUseRequireTLS 20, 23
UUEncode 9

- W -

W32/Nicehello@MM 21
Web-Services 33
What is TCP/IP?
 Part 1 31
 Part 2 31
 Part 3 31
Win32
 Ping 17
Windows 95
 Ping 17
 Winsock 2 3
Windows 98
 Ping 17
Windows ME
 Ping 17
Windows NT
 Ping 17
Windows Sockets Network Programming
 Appendix C 31
Winshoes 2
 Indy 2
 Migrating 2
Winsock 2
 Windows 95 3
worm 21
WS2_32.DLL 3

- X -

X-Library 21

- Y -

YEnc 9