

Developer's Network Component Library
Version 1.0

Programmer's Manual

BreZaSoft, LLC
Copyright © 2003

License Agreement

You should carefully read the following terms and conditions before using the Developer's Network Component Library software. Your use of the Developer's Network Component Library software indicates your acceptance of this license agreement. Do not use the Developer's Network Component Library software if you do not agree with this license agreement.

Disclaimer of Warranty

The Developer's Network Component Library software and the accompanying files are distributed and licensed "as is". BreZaSoft, LLC disclaims all warranties, either express or implied, including, but not limited to implied warranties of merchantability and fitness for a particular purpose. Should the Developer's Network Component Library software prove defective, the licensee assumes the risk of paying the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will BreZaSoft, LLC be liable for any damages whatsoever (including without limitation damages for loss of business profits, business interruption, loss of business information and the like) arising out of the use or the inability to use the Developer's Network Component Library software even if BreZaSoft, LLC has been advised of the possibility of such damages.

Copyright

The Developer's Network Component Library software is protected by copyright, as well as other intellectual property laws and treaties. The Developer's Network Component Library software is licensed not sold. Title to the Developer's Network Component Library software shall at all times remain with BreZaSoft, LLC.

Version

This version of the Developer's Network Component Library (the "software") software is a fully functional, nonrestrictive compilation.

Subjected to the conditions in this license agreement:

- You may use the unmodified version of the Developer's Network Component Library software without charge, provided you are using the software for personal applications only. You will be required to purchase a license if:
 - You are using the software in an enterprise environment.
 - You are using the software in commercial applications.
 - You are using the software in applications written for financial gain.

- You may redistribute the unmodified version of the Developer's Network Component Library software, provided:
 - You do not charge for the software.
 - You acknowledge the developer of the software as BreZaSoft, LLC.
 - You include the unmodified Programmer's Manual (this manual) with the package.

- You may embed (or instantiate in any way) the unmodified version of the Developer's Network Component Library software (or part of it), in a product and distribute the product, provided you do not charge for the product. If you are charging for the product, you will be required to purchase a license.

Ordering Information

Please refer to the BreZaSoft, LLC Web site at <http://www.BreZaSoft.com>

Support Contact

Please refer to the BreZaSoft, LLC Web site at <http://www.BreZaSoft.com>

Table of Contents

<i>Introduction</i>	6
Welcome to BreZaSoft's Developer's Network Component Library	6
<i>Installation</i>	7
Setup	7
Sample Projects.....	7
Testing the Developer's Network Component Library Installation	7
Troubleshooting the Developer's Network Component Library Installation.....	7
<i>Coding with Whois</i>	8
Overview	8
Properties	8
Methods.....	8
Sample Code	9
<i>Coding with Telnet</i>	10
Overview	10
Properties	10
Methods.....	10
Sample Code	11
<i>Coding with NSLookup</i>	13
Overview	13
Properties	13
Methods.....	14
Resource Records.....	14
Sample Code (Simple)	15
Sample Code (Detailed).....	16

Introduction

Welcome to BreZaSoft's Developer's Network Component Library

BreZaSoft's Developer's Network Component Library classes are compiled COM objects. The application developer instantiates an instance of a Network Component class from their web page or application via a call to CreateObject. From there they populate different properties and call a method that queries the appropriate server/service. The Developer's Network Component Library consists of the following classes:

- Whois
- Telnet
- NSLookup

Installation

Setup

To install the Network Component Library, simply run the accompanying setup executable (setup.exe) file. This will launch and guide you through the rest of the process.

Sample Projects

The Network Component Library comes with a number of Sample Projects. These projects serve as good tutorials on how to use the Developer's Network Component Library software. While the Sample Projects include both ASP pages as well as Visual Basic 6 projects, for simplicity's sake, all examples in this document are ASP pages.

Testing the Developer's Network Component Library Installation

The easiest way to verify that the Network Component Library was setup correctly is to simply run one of the Sample Projects packaged with the Library. If the sample runs correctly, it can be confirmed that the Network Component Library was installed successfully.

Troubleshooting the Developer's Network Component Library Installation

When testing your installation of the Developer's Network Component Library, please take the following into consideration:

- If you are planning on utilizing the Library from ASP web pages, ensure that the Windows account that IIS is running under has sufficient privileges to access the file that houses the Developer's Network Component Library classes (BZSNetComp.dll)

Coding with Whois

Overview

This component sends queries to any RFC954 whois server and captures the results to its Result property.

Properties

Property Name	Description
Host	This is a string property that holds the name of the whois server that this class will query in response to a call to its Query method. This property has a default value of “whois.crsnic.net”.
Port	This is an integer property that holds the TCP port number that this class will connect in response to a call to its Query method. This property has a default value of 43.
Recurse	This is a Boolean property. If the server response to a query contains the text, “Whois Server: {hostname}”, the class will then requery {hostname} with the same criteria if this property is set to TRUE. The default value of this property is FALSE.
Result	This is a string property that contains the results of a query to a whois server. This property has a default value of “” (blank).
Timeout	This is an integer property that holds the length of time (in seconds) that the class should wait for a response from the whois server before giving up. This property has a default value of 10.

Methods

Method Name	Arguments	Description
Query	Criteria, <i>Optional</i> Host	This method initiates a whois query to a whois server and waits for a response. If the timeout value specified in the timeout property expires, this method will raise an error. It takes one mandatory argument and one optional argument. “Criteria” is the domain you’d like to gather whois information on. The optional “Host” argument can be used to override the Host property.

Sample Code

```
<% @ Language=VBScript %>
<%

Option Explicit

Dim Whois
Dim Successful

Set Whois = Server.CreateObject( "BZSNetComp.Whois" )
Whois.Timeout = 30
Whois.Host = "whois.networksolutions.com"
Whois.Port = 43
Whois.Recurse = True

Successful = Whois.Query( "BreZaSoft.com" )

If Not Successful Then
    Response.Write "There was a problem querying the whois server."
    Response.End
End If

%>
<HTML>
<BODY>
<PRE>
<% =Whois.Result %>
</PRE>
</BODY>
</HTML>
<%

Set Whois = Nothing

%>
```

Coding with Telnet

Overview

This component is an ActiveX user interface to the TELNET protocol.

Properties

Property Name	Description
ConnectionOpen	This is a Boolean, read-only property. It is TRUE if there is an established connection. It is FALSE otherwise.
Host	This is a string property that is used to identify the host that the class is either connected to or will be connecting to once the Connect method is called. This property has a default value of "" (blank).
Port	This is an integer property that holds the TCP port number that this class will attempt to connect to in response to a call to its Connect method. This property has a default value of 23.
Received	This is a string, read-only property. It is a buffer that holds the responses from the host that this class is connected to. If the TruncateOnRead property is set to TRUE, this property will become "" (blank) after each access to it. (See TruncateOnRead). This property has a default value of "" (blank).
Timeout	This is an integer property that holds the length of time (in seconds) that the class should wait for a connection to be established when connecting to a host. This property has a default value of 5.
TruncateOnRead	This is a Boolean property. It tells the class whether or not to truncate (blank out) the Received property between reads. If this property is set to TRUE, it will "blank out" the Received property every time it is accessed, thus only showing the latest responses. If it is set to FALSE, the Received property will always contain a complete history of all responses from the host since the initial connection. This property has a default value of TRUE.

Methods

Method Name	Arguments	Description
Connect	<i>Optional</i> Host, <i>Optional</i> Port	This method attempts to establish a TCP connection to the host/port identified in their respective properties. If an optional host and port are specified as arguments to this method, they will override their property counterparts. If the

		timeout expires, this method will raise an error.
Disconnect		This method disconnects from the previously connected host.
Send	Stream	This method will send the specified Stream of characters to the connected host.
Wait	<i>Optional Seconds</i>	This method makes the class wait for a response from the connected to host. It will block until one of two things occurs: 1. the connection is closed by the host or 2. the number of seconds specified in the Seconds argument have elapsed.
WaitForText	<i>Optional Criteria, Optional Seconds</i>	This method is a variation of the Wait method. It can be used when you know that a lot of responses will be coming from the connected to host, but are waiting for a specific response. It will block until one of three things occurs: 1. the connection is closed by the host, 2. the number of seconds specified in the Seconds argument have elapsed or 3. the Received property contains the text found in the Criteria property.

Sample Code

```

<% @ Language=VBScript %>
<%

Dim Telnet

Set Telnet = Server.CreateObject( "BZSNetComp.Telnet" )

Telnet.Host = "fictionalUNIXServer.ficDomain.org"
Telnet.Port = 23
Telnet.Timeout = 10

'Look for drive utilization.
Telnet.Connect
Telnet.WaitForText "login:", 5
Telnet.Send "myUsername" & Chr(10)
Telnet.WaitForText "password:", 5
Telnet.Send "myPassword" & Chr(10)
Telnet.WaitForText "%", 5
Telnet.Send "df" & Chr(10)
Telnet.Wait 3
Telnet.Send "exit" & Chr(10)
Telnet.Disconnect
%>
<HTML>
<BODY>

<PRE>
<%

Telnet.Received

%>
</PRE>

</BODY>
</HTML>

```

<%

Set Telnet = Nothing

%>

Coding with NSLookup

Overview

NSLookup is an ActiveX component to query Internet domain name servers. It has two modes: simple results and detailed results. In the simple results mode, NSLookup will store query results as a string in the Result property. In detailed results, NSLookup will store a collection of Resource Records (RR) in the Result property. NSLookup conforms to the Network Working Group's RFC1035.

Properties

Property Name	Description
QueryClass	<p>This is an integer property that holds the QCLASS field of the query to be performed when the Query method is called. Valid values for this property include:</p> <ul style="list-style-type: none"> • 1 (IN - the Internet) • 2 (CS - the CSNET class) • 3 (CH - the CHAOS class) • 4 (HS - Hesiod [Dyer 87]) <p>For more information on these values, see RFC1035. These values are enumerated in the BZSNetComp.inc include file. This property has a default value of 1.</p>
QueryType	<p>This is an integer property that holds the QTYPE field of the query to be performed when the Query method is called. Valid values for this property include:</p> <ul style="list-style-type: none"> • 1 (A – a host address) • 2 (NS – an authoritative name server) • 3 (MD – a mail destination) • 4 (MF – a mail forwarder) • 5 (CNAME – the canonical name for an alias) • 6 (SOA – marks the start of a zone authority) • 7 (MB – a mailbox domain name) • 8 (MG – a mail group member) • 9 (MR – a mail rename domain name) • 10 (NULL – a null RR) • 11 (WKS – a well known service description) • 12 (PTR – a domain name pointer) • 13 (HINFO – host information) • 14 (MINFO – mailbox or mail list information)

	<ul style="list-style-type: none"> • 15 (MX – mail exchange) • 16 (TXT – text strings) <p>For more information on these values, see RFC1035. These values are enumerated in the BZSNetComp.inc include file. This property has a default value of 1.</p>
Result	The type of this property is dependent upon the value of the SimpleResults property. If the SimpleResults property is set to TRUE, upon querying a server, the Result property will contain a string with the answer to the query. If the SimpleResults property is set to FALSE, then the Result property will contain a collection of Resource Records (RRs). In either case, this property is read-only. It has a default value of "" (blank).
Server	This is a string property that is used to identify the IP address of the host that NSLookup will query for DNS results. This property has a default value of "" (blank). A valid value must be supplied for this property in order for the Query method to work.
SimpleResults	This is a Boolean property that is used to put NSLookup into one of two modes: SimpleResults or DetailedResults. If this property is set to TRUE, then NSLookup will run in SimpleResults mode. Otherwise it will run in DetailedResults mode. This property has a default value of TRUE.
Timeout	This is an integer property that holds the length of time (in seconds) that the class should wait for a response from the DNS server. This property has a default value of 5.

Methods		
Method Name	Arguments	Description
Query	Criteria	This method formulates a DNS request message and sends it to the DNS server specified in the Server property. It then blocks until it either receives a response from the server or the timeout value expires. Upon receiving a response from the DNS server, it translates the response, depending on the value of the SimpleResults property to either a string or a collection of RR's and stores in the Result property.

Resource Records
If an NSLookup query is initiated with the SimpleResults property set to FALSE, Query will put a collection of Resource Records into the Result property. (Conversely, if the SimpleResults property is set to TRUE, this section does not apply.) This collection can be traversed like any other collection (for example, you could traverse it via a For Each loop). The format of the Resource Record class is as follows:

Resource Record	
NAME	String
R_TYPE	Integer
R_CLASS	Integer
TTL	Long
RDLENGTH	Integer
RDATA	Object

The fields of the Resource Record correspond to the Resource Record definition in RFC1035. When NSLookup processes the response from a DNS server, it will generate RDATA objects and point the RDATA field of the Resource Record object to them. There are a number of RDATA objects, each one corresponding to RR definitions found in RFC1035. Listed below are their definitions:

RDATA SOA	
MNAME	String
RNAME	String
SERIAL	Long
REFRESH	Long
RETRY	Long
EXPIRE	Long
MINIMUM	Long

RDATA MB	
MADNAME	String

RDATA MD	
MADNAME	String

RDATA MF	
MADNAME	String

RDATA MR	
NEWNAME	String

RDATA PTR	
PTRDNAME	String

RDATA MINFO	
RMAILBX	String
EMAILBX	String

RDATA A	
ADDRESS	String

RDATA CNAME	
CNAME	String

RDATA HINFO	
CPU	String
OS	String

RDATA MG	
MADNAME	String

RDATA MX	
PREFERENCE	Integer
EXCHANGE	String

RDATA NS	
NSDNAME	String

Sample Code (Simple)

```
<%@ Language=VBScript %>
<%
Option Explicit

Dim NSLookup
Dim Success
```

```

Set NSLookup = Server.CreateObject( "BZSNetComp.NSLookup" )

NSLookup.Server = "216.68.1.100"

Success = NSLookup.Query( "www.brezaSoft.com" )
If Not Success Then
    Response.Write "Error performing DNS query."
    Response.End
End If

%>
<HTML>
<BODY>

<P>
www.brezaSoft.com resolves to <% =NSLookup.Result %>
</P>

<%

'Let's try a reverse lookup
Success = NSLookup.Query( "66.218.71.92" )
If Not Success Then
    Response.Write "Error performing DNS query."
    Response.End
End If

%>
<P>
66.218.71.92 resolved to <% =NSLookup.Result %>
</P>

</BODY>
</HTML>
<%

Set NSLookup = Nothing

%>

```

Sample Code (Detailed)

```

<% @ Language=VBScript %>
<%

Option Explicit

Dim NSLookup
Dim DomainName
Dim Success
Dim RR

Set NSLookup = Server.CreateObject( "BZSNetComp.NSLookup" )

NSLookup.Server = "216.68.1.100"
NSLookup.SimpleResults = False
NSLookup.QueryType = 6 'SOA Query

DomainName = "brezaSoft.com"

Success = NSLookup.Query( DomainName )
If Not Success Then
    Response.Write "Error performing DNS query."
    Response.End
End If

%>
<HTML>

```

```

<BODY>
<P>Response for query on <% =DomainName %> contained <% =NSLookup.Result.Count %> record(s).</P>
<TABLE BORDER="1">
  <TR><TH>RR Type</TH><TH>Values</TH></TR>
<%
For Each RR In NSLookup.Result
  Response.Write "<TR><TD>"
  Select Case RR.R_TYPE
    Case 1:
      Response.Write "A Record"
    Case 2:
      Response.Write "NS Record"
    Case 6:
      Response.Write "SOA Record"
    Case Else:
      Response.Write "Other (" & RR.R_TYPE & ")"
  End Select
  Response.Write "</TD><TD>"
  Select Case RR.R_TYPE
    Case 1:
      Response.Write "ADDRESS: " & RR.RDATA.ADDRESS
    Case 2:
      Response.Write "NSDNAME: " & RR.RDATA.NSDNAME
    Case 6:
      Response.Write "MNAME: " & RR.RDATA.MNAME & "<BR>"
      Response.Write "RNAME: " & RR.RDATA.RNAME & "<BR>"
      Response.Write "SERIAL: " & RR.RDATA.SERIAL & "<BR>"
      Response.Write "REFRESH: " & RR.RDATA.REFRESH & "<BR>"
      Response.Write "RETRY: " & RR.RDATA.RETRY & "<BR>"
      Response.Write "EXPIRE: " & RR.RDATA.EXPIRE & "<BR>"
      Response.Write "MINIMUM: " & RR.RDATA.MINIMUM & "<BR>"
    Case Else:
      Response.Write "(Other)"
  End Select
  Response.Write "</TD></TR>"
Next
%>
</TABLE>
</BODY>
</HTML>
<%
Set NSLookup = Nothing
%>

```