



ダイヤモンドブリコット電話研究所
山崎 はるか
 YAMAZAKI, Haruka
<http://www.nda.co.jp/>

第10回 DNSにクエリーを投げる

DNSに“もしもし”

.NETプログラミングにおいて、名前解決（ドメインの正引き／IPからの逆

Level



Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

Samples

この記事で取り上げたソースコードおよびサンプルプログラムは、
<http://www.shoetisha.com/mag/windev/>
 からダウンロード可能です。

引き)を意識することはほとんどない。TcpClientクラス(Connectメソッド)やWebRequestクラスなど、主要な通信クラスは、名前解決を自動で行なっているからだ。

強いて正引き／逆引きを行なうときも、Dnsクラスを使う。本誌2005年9月号の本稿を見ればわかるように、正逆引きに限っていえば、通常のアプリケーションで使うならDnsクラスの機能があれば充分である。

しかし、ネットワークを利用するだけでなく「ネットワークそのものを調査する」といった局面だと、これでは物足りないことがある。たとえば、

- ・使用するネームサーバーを指定したい(既存クラスやDnsクラスでは割り当て済みのDNSしか使えない)
- ・MXレコードが引きたい(メーラーの実装に不可欠)

というとき。

こういった場合は、DNSに直接／ク

エリー(問い合わせコード)を投げることになる。

DNS(Domain Name Server)は、ポート53、TCPとUDPを用いている。

TCPはDNS同士のゾーン転送(プライマリ／セカンダリ)に使用され、UDPはクライアントや外部DNSからのクエリー問い合わせに用いられる。

一般にクライアントが使うのは、UDPである。DNSにとっても、到来頻度の多い「クエリー受付／応答」はオーバーヘッドの少ないUDPを使ったほうが、効率的で安全だからだ。

今回は、UDP/53を用いた、DNSクライアントを作り、DNSの動き方を体験してみよう。

nslookupは何をしているか

まず最も身近なDNSクライアント、「nslookup」の動きを観察してみよう。簡単にモニターするために、サンプル1(リスト1)を用意したので、これを起

リスト1：サンプル1 (UDP53モニター)

```
'名前空間のインポート -----
Imports System.Threading ' Threadクラス
Imports System.Net.Sockets ' UdpClientクラス
Imports System.Net ' IPEndPointクラス
' -----

Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
    Dim receive As New Thread(AddressOf ReceiveThread)
    receive.Start()
End Sub

Private Sub ReceiveThread()
    ' 設定 & 接続
    Dim udpClient As New UdpClient(53) ' DNSポート53番でリスニングする

    Dim riep As New IPEndPoint(IPAddress.Any, 0)
    Dim bary As Byte()

    Dim a As Integer
    Dim s As String

    While True
        ' 受信
        bary = udpClient.Receive(riep)
        ' 表示
        For a = 0 To bary.Length - 1
            's = bary(a).ToString.ToUpper
            s = Convert.ToString(bary(a), 16).ToUpper
            ' 1桁の場合は0?hのように'0'を追加
            If s.Length <= 1 Then s = "0" + s
            TextBox1.Text += s + " "
        Next

        TextBox1.Text += vbCrLf
    End While
End Sub

Private Sub Form1_Closed(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles MyBase.Closed
    End
End Sub
```

図1：サンプル1のnslookupコンソール

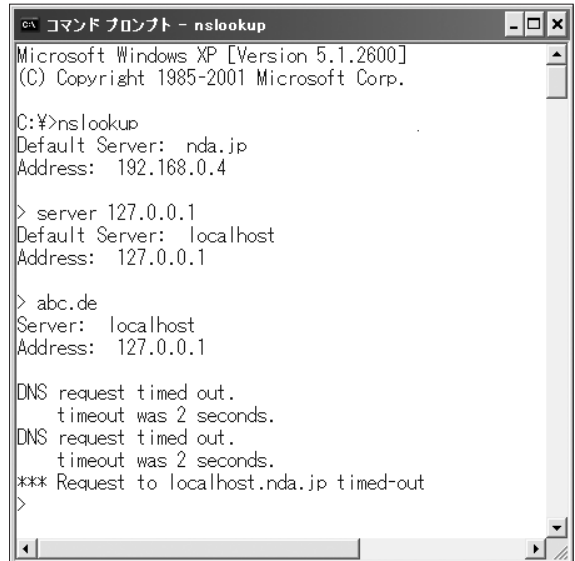


図2：16進文字列が表示された

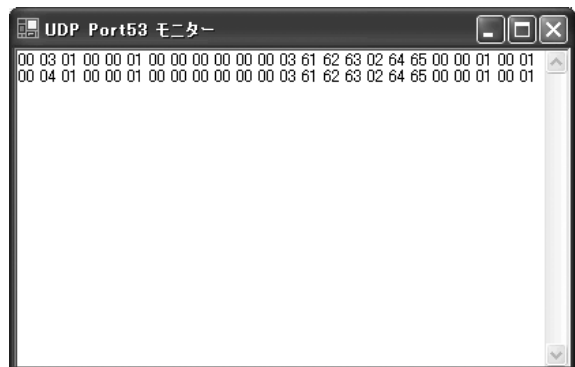
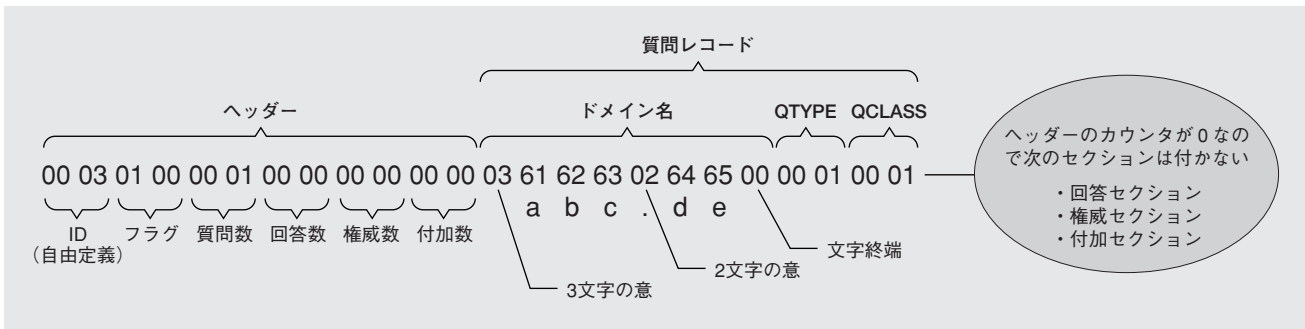


図3：送出されたクエリーの意味



動してほしい。そして、コマンドプロンプトからnslookupを立ち上げ、次のように入力する (図1)。

```
> server 127.0.0.1
> abc.de
```

すると、サンプル1の画面には、なにかや16進文字列が表示される (図2)。これが「クエリー」の正体である。この例では、わずか24バイト。この短いクエリーをDNSに投げることで「abc.

de」というドメイン名の正引きが行なわれるわけである。このクエリーには、図3のような意味がある。この規則については、次の項で解説する。