

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

Nel caso fosse necessario centralizzare i log di attività in unico repository, è possibile inviare messaggi di log direttamente ad un syslog di server LINUX o di un devices Linux embedded attraverso una comunicazione udp o tcp da C#.

Prima di tutto è necessario configurare il syslog del Server Linux, in questo caso per semplicità utilizzerò UBUNTU che, essendo una distribuzione basata su Debian, ha come syslog di default rsyslog:

modificare il file `/etc/rsyslog.conf` eliminando i commenti alle righe seguenti:

```
$ModLoad imudp
```

```
$UDPServerRun 514
```

modificate la porta da 514 ad un valore maggiore di 1024 (a causa di problemi legati alla sicurezza, io ho utilizzato la porta 2000)

salvate il file e ravviate il servizio:

```
# service rsyslog restart
```

Lato Visual Studio create un progetto console ed utilizzate i namespace seguenti:

```
using System.Net;    using System.Net.Sockets;
```

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

Avviamo ora una sessione UDP tramite il sockets inserendo l'ip address del server UBUNTU e la porta corrispondente al SYSLOG in attesa di input, in questo caso la porta 2000.

Creiamo una classe Client.cs e inseriamo la funzione Send:

```
public class Client
{
    private IPEndPoint ipLocalEndPoint;
    private IPHostEntry ipHostInfo;
    private IPAddress ipAddress;
    private string _sysLogServerIp = null;
    private int _port = 2000;

    public Client()
    {
        ipHostInfo = Dns.GetHostEntry(Dns.GetHostName());
        ipAddress = ipHostInfo.AddressList[0];
        ipLocalEndPoint = new IPEndPoint(ipAddress, 0);
    }

    public int Port
    {
        set { _port = value; }
        get { return _port; }
    }

    public string SysLogServerIp
    {
        get { return _sysLogServerIp; }
        set
        {
            if ((_sysLogServerIp == null))
            {
                _sysLogServerIp = value;
            }
        }
    }

    public void Send(SendToSyslog.Message message)
    {
```

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

```
        Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
        ProtocolType.Udp);

        IPAddress serverAddr = IPAddress.Parse(_sysLogServerIp);

        IPEndPoint endPoint = new IPEndPoint(serverAddr, _port);

        int priority = message.Facility * 8 + message.Level;
        string msg = System.String.Format("<{0}>{1} {2} {3}",
            priority,
            DateTime.Now.ToString("&quot;MMM dd
HH:mm:ss&quot;),
            ipHostInfo.HostName,
            message.Text);
        byte[] bytes = System.Text.Encoding.ASCII.GetBytes(msg);

        sock.SendTo(bytes, endPoint);
        sock.Close();// chiudo il socket

    }

}
```

notiamo un valore priority che , per convenzione del syslog di linux, è determinato da alcuni valori numerici come segue:

```
int priority = message.Facility * 8 + message.Level;
```

Nella classe creata inseriamo l'enumerazione dei due valori:

```
public enum Level
```

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

```
{  
    Emergency = 0,  
    Alert = 1,  
    Critical = 2,  
    Error = 3,  
    Warning = 4,  
    Notice = 5,  
    Information = 6,  
    Debug = 7,  
}
```

```
public enum Facility  
{  
    Kernel = 0,  
    User = 1,  
    Mail = 2,  
    Daemon = 3,  
    Auth = 4,  
    Syslog = 5,  
    Lpr = 6,  
    News = 7,  
    UUCP = 8,  
    Cron = 9,  
    Local0 = 10,  
    Local1 = 11,  
    Local2 = 12,  
    Local3 = 13,  
    Local4 = 14,  
    Local5 = 15,  
    Local6 = 16,  
    Local7 = 17,  
}
```

notiamo che come argomento c'è una classe message così composta:

```
public class Message  
{  
    private int _facility;  
    public int Facility  
    {  
        get { return _facility; }  
    }  
}
```

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

```
        set { _facility = value; }
    }
    private int _level;
    public int Level
    {
        get { return _level; }
        set { _level = value; }
    }
    private string _text;
    public string Text
    {
        get { return _text; }
        set { _text = value; }
    }
    public Message() { }
    public Message(int facility, int level, string text)
    {
        _facility = facility;
        _level = level;
        _text = text;
    }
}
```

inseriamo anche questa nella nostra Classe Client.cs

ora dal main del programma richiamiamo la funzione Send così:

```
static void Main(string[] args)
{
    SendToSyslog.Client c = new SendToSyslog.Client();
    try
    {
        c.Port = 2000;
        c.SysLogServerIp = "192.168.20.65"; // syslogd on local machine
        int facility = (int)SendToSyslog.Facility.User; // Local5
        int level = (int)SendToSyslog.Level.Warning; // Debug;
        string text = (args.Length > 0) ? args[0] : "Hello, Syslog.";

        c.Send(new SendToSyslog.Message(facility, level, text));
    }
}
```

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38

```
    }  
    catch (System.Exception ex1)  
    {  
        Console.WriteLine("&quot;Exception! &quot; + ex1);  
        Console.ReadKey();  
    }  
}
```

Ora Potete verificare con un programma di log reader su Ubuntu (io uso KsystemLog) il risultato.

C# - Inviare Messaggi a Syslog remoto

Scritto da Administrator

Giovedì 06 Marzo 2014 09:00 - Ultimo aggiornamento Giovedì 06 Marzo 2014 09:38
