

## ARTICLES

[home](#) » [articles](#) » 6

## Controlling Windows Vista from Linux

Written by [Raimo](#), 26 Apr 2010

This is a simple "getting started guide" to show you how to control a Windows XP/Vista machine from Linux (or *vice versa*).

*Most of what follows is true for any platform to which Erlang has been ported - Linux and Windows XP/Vista are used in the examples because these operating systems are familiar to most people.*

Below follows a program which manipulates the Windows Registry from Linux. This is an example of a program that could make a system administrators life a *lot* easier. In principle a single Linux machine could be used to remotely administer a large network of Windows machines.

Before you can do any of this you will have to install Erlang on both Windows and Linux.

To do this fetch the Linux and Windows versions of Erlang from [www.erlang.org](http://www.erlang.org) and follow the installation instructions.

## Starting two Erlang nodes on the same LAN

This example shows how to start two machines (a Linux machine and a Windows XP/Vista machine) on the same LAN.

On the Linux machine start the system with the command:

```
erl -sname m1
```

You will see something like the following:

```
Erlang R13B (erts-5.7.1) [smp:4:4] [rq:4] [async-threads:0] [kernel-poll:false]
Eshell V5.7.1 (abort with ^G)
(m1@mylinuxhost)1>
```

Here the prompt shows the Erlang node identifier **m1** and the host name **mylinuxhost** (the combination **m1@mylinuxhost** we call the node name)

On the Windows machine give the command:

```
C:\Program Files\erl5.7.1\bin\werl -sname x
```

Assuming that you have installed Erlang in the directory "C:\Program Files\erl5.7.1" where erl5.7.1 is dependent on the version of Erlang you installed.

On the Windows machine a new window pops up containing:

```
Erlang R13B (erts-5.7.1) [smp:2:2] [rq:2] [async-threads:0]
Eshell V5.7.1 (abort with ^G)
(x@mywindowshost)1>
```

If you got this far then you now have two Erlang nodes on the same LAN but they will not necessarily be able to talk to each other.

Before the two nodes can talk to each other they have to be authenticated, Follow the instructions in the next section to set up the authentication system.

Once authenticated the nodes can "ping" each other, this can be used as a final test that everything works. To do this, start two nodes as described above make sure they have the same cookies and then try the command **net\_adm:ping/1** as follows:

```
(m1@mylinuxhost)> net_adm:ping(x@mywindowshost).
pong
```

It worked!

## Authentication

Distributed Erlang authentication is based on the idea of shared secrets and challenge response algorithms. The shared secrets are called cookies. In order for two Erlang nodes to talk to each other they have to agree on the value of a cookie. The commands **erlang:set\_cookie/2** and **erlang:get\_cookie/0** can be used to set and read the the value of a cookie. For example:

```
1> erlang:set_cookie(node(), abcdef123XYZ).
true
2> erlang:get_cookie().
abcdef123XYZ
```

If two distributed node cannot ping each other (with the command **net\_adm:ping(NodeName)**), then the first thing to do is to use the **get\_cookie** command to see if both nodes have the same cookie.

The cookie can be permanently set up by editing the file **HOME/.erlang.cookie** where the value of HOME can be found out by evaluating **init:get\_argument(home)**

## Problems with distribution

How do you want to browse the articles?

- [by year](#)
- [by author](#)
- [by tag](#)

A lot of things can go wrong when setting up distributed Erlang. Most often these are due to errors in the way DNS etc. have been set up. If you have problems in setting up distributed Erlang read the on-line documentation section [Getting Started](#) for help.

## Manipulating the win32 registry from linux

The module **win32reg** can be used to manipulate the windows registry.

The program `print_registry.erl` is used to remotely access the windows registry. Read the comments in the code to see how this works. See also the output obtained when running this program:

**Exercise:** Extend the program so you can read and write files on the Windows machine that were sent from the Linux machine.

hint:

`show source`

Implement a complete remote management system for Windows machines, that is, a system that will allow an administrator to remotely add/remove files and tweak entries in the windows registry. Add security and authentication to your program. [Hint you'll find everything you need MD5 etc. in the Erlang distribution].

## print\_registry.erl

`show source`

## Output from print\_registry.erl

`show source`

~~Translated from previous erlang.org win95demo.html.~~

Tags: [ [examples](#) ]

-----

Copyright 2011 erlang.org - Contact [webmaster@](mailto:webmaster@erlang.org) - RSS Feeds - Powered by Erlang Web