Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 62 (2012) 824 – 831

Procedia
Social and Behavioral Sciences

WC-BEM 2012

A New Management Tools For Remote-Access Through Lan (P2P) Using Wmi Technology

Anis Ismail, Mohammad Hajjar, Mazen El-Sayed

Department of Computer Network and Telecommunications EngineeringUniversity Institute of Technology – Saida
Lebanese University Liban, Saida BP 813 anismaiil@yahoo.com m_hajjar@ul.edu.lb mazen_elsayed@yahoo.fr

Abstract

In this paper, we present a new management system permitting to explore remotely , in real time, a computer connected to a local network or Peer to Peer (P2P) Network, from a remote station, without that the user of this computer realizes this exploration. The strength of this system, it requires no previous installation on stations to explore. The requirement is only to have an account for these stations. The exploration happens in a different session, even though the same account is opened on this station. We will show how to have access operating system information, services, and processes running on local remote machine as well as on a remote machine on the network, provided administration rights to them. Also we will show how to start and stop services, terminate processes, and create new processes from WMI. The strength of this system is to permit the possible utilization remotely, through a local network or P2P Network, of a remote computer as if one was there. WMI (Windows Management Instrumentation) is a tool respecting the standard of the administration's domain, a system that permits to get information systems of the following machine. It can also allow managing a machine while modifying directly the well stocked parameters.

Keywords: Exploration, administration, WMI, LAN (Peer to Peer), Visual Basic.NET 2005;

1. Introduction

Computers are coupled together in a network to provide digital communications. Computer networks are critical for carrying digital data used for Internet, IP telephony, video teleconferencing, e-commerce, file transfers, e-mail, databases, etc. And as these networks become bigger, more powerful, and more versatile, the networked computers can be programmed to perform more complex tasks. Accordingly, software applications are likewise becoming quite sophisticated. One problem arises when software has to be distributed across the network (Peer to Peer). For instance, when new software is routinely being installed to give users new functionalities; software upgrades are installed for improved efficiency; and network software is installed to help monitor and administer the performance of the network [1]. In the past, a network administrator would have to physically visit each of the individual computers and manually install the software onto that particular station. This is a time consuming task. Furthermore, it is inefficient because of the associated downtime as the software was being installed. In addition, there is much overhead expended in administering, tracking, and maintaining an accurate and updated log of the software state for each of the computers across the entire network (Peer to Peer). There can also be problems encountered with respect to software compatibility, troubleshooting, and licensing issues.

Different methods and tools frequently used to administer remote Windows systems, and which let you able to perform basic system administration [2], such as, Microsoft Remote Procedure Call "MSRPC" [3] which is an interprocess communication "IPC" mechanism that enables data exchange and invocation of functionality residing in a different process. That process can be on the same computer, on the local area "LAN", on P2P Network (Peer to

doi:10.1016/j.sbspro.2012.09.139

Peer), or across the Internet. The Microsoft RPC mechanism uses other IPC mechanisms, such as named pipes, NetBIOS, or Winsock, to establish communications between the client and the server, Web based remote administration tools which do not require permanent client software or a network change but need web server to take control remote computer, Windows Management Instrumentation (WMI) is an infrastructure that enables you to access and modify standards-based information about objects, such as computers, applications, and network components, in your enterprise environment. Using WMI, you can create powerful administration applications to monitor and respond to specific events in your environment [4] Where we had been used to built our system according to their advantages that are no previous installation on the remote stations, part of windows 2000/XP/vista/7.

A bibliographic research permitted to localize some software doing tasks separated, administration, surveillances, using the WMI technology. We can mention for example: the software Goverlan Remote Administration [5], XR PerfMon Tools [6], BMC Software's [7], WMI Explorer [8] that allow to Explore the full set of WMI management classes, objects and their properties, to Browse through objects and settings on remote machines and to Execute any WQL query and view the result set. The major inconvenience of those softwares is that they are not free.

The objective of this paper is to present a new secure system permitting to explore remotely, in real time, a computer connected to a local network (Peer to Peer), without the user of that computer realizes this exploration. The strength of this system, it requires no previous installation on stations to explore. These interfaces possess the same presentations that the equivalent Windows. Thus, it is possible to explore hard disks, files, indexes, the register, the active applications, in progress processes and services of the remote computer while offering the same functionalities of Window system is to give back possible remote utilization, through the local network (Peer to Peer), of a remote computer as if one was there, without needing knowledge of network or system.

This system is developed using the Visual Basic.NET 2005 language [9], [10] and built on WMI technology (Windows Management Instrumentation) [11]. The WMI architecture is normalized interface permitting to reach resources of a computer under Windows, to interrogate them, to manage them and even to configure them.

The following of this article is organized as following. In the paragraph 2, we present the WMI architecture. The paragraph 3 described the security at WMI level. The paragraph 4 described functionalities of our system. Finally, a general conclusion will be provided in the last paragraph.

2. WMI Architecture

WMI stands for Windows Management Instrumentation [12]. This is the Microsoft implementation to two industry standards of DMTF (Desktop Management Task Force) the first is CIM (Common Information Model) and the second is WBEM (Web-Based Enterprise Management).

WMI core is already a part of windows ME/2000/XP/vista/7. The WMI enables the management capabilities by supplying the standard storage component (CIM), the means to set and get information to and from the storage and the ability to dock third party providers (same as plug-in) to the Providers Manager (CIMOM), shown in Figure 1. WMI Provider is a software component that functions as mediator between the CIM Object Manager and managed objects[13].

In order to expose software component such as a service through the WMI one need to write WMI provider. This plug-in (provider) exposes the service to the WMI and provides the interface to receive information and to interact with the service. Till recently WMI Providers was written as a COM component and now with the emerging of .NET framework it is easier to develop providers [14]. In the WMI architecture, all depends on the CIMOM (Common Information Model Object Manager) (Fig.1). This is not simply a central core; it is a motor. WMI Data Providers retrieve information from managed resources. A managed resource is a logical or physical component that can be accessed and managed using WMI. Examples of standard Windows resources that can be managed by using WMI include the computer system, disks, peripheral devices, event logs, files, folders, etc. WMI providers access the managed resources using the appropriate Application Programming Interfaces “API”, and expose the data to the WMI infrastructure using a standards-based object-oriented data model.

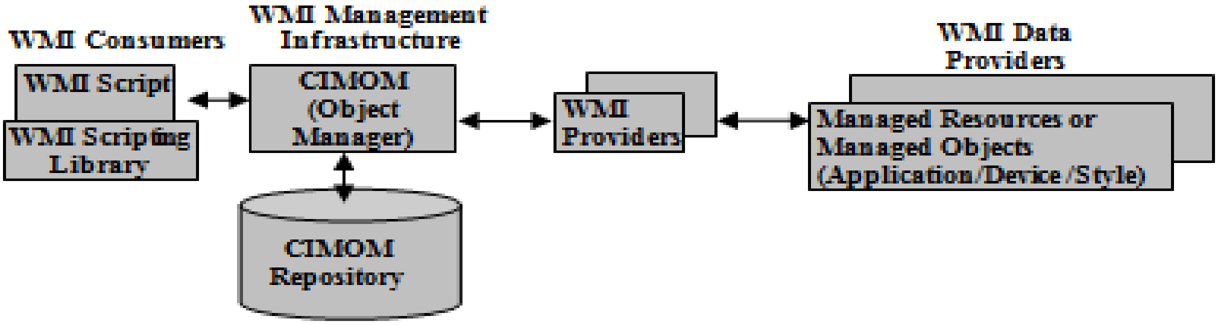


Figure 1. WMI Architecture

The operating system comes bundled with standard providers for accessing standard operating system resources such as processes, registry settings, etc. Software developers can add and integrate additional providers for exposing data and management functionality that are unique to their products. The OpenManage Server Administrator's CIM Reference Guide contains additional information on the data model supported by this provider. WMI Management Infrastructure consists of the Common Information Model Object Manager "CIMOM" or the WMI service. It handles the interaction between consumers and providers. All WMI requests and data flow through the CIMOM. The WMI service is similar to other operating system services.

Management applications, administrative tools and scripts [15] make requests to the CIMOM to retrieve data, subscribe to events or to perform management-related tasks. The CIMOM retrieves the provider and class information to service the request from the CIM repository. The CIMOM uses the information obtained from the CIM repository to hand in the consumer request to the appropriate WMI provider.

The WMI use some queries very similar to the SQL that originates the name of WQL (WMI Query Language). Briefly, WQL is a subset of the standard American National Standards Institute Structured Query Language (ANSI SQL) with minor semantic changes. A basic WQL query remains fairly understandable for people with a basic SQL knowledge. As if the computer was a data base, the WMI executes a query in order to allow again the chosen information only of it. So we create an ObjetQuery object containing our query, here we ask over for the name of the skilled worker and the connected user. Then, we create an ObjetSearcher, which serves to extract a collection of objects of management according to a specified query. Therefore, WQL is dedicated to WMI and is designed to perform queries against the CIM repository to retrieve information or get event notifications.

3. SECURITY at the WMI Level

At the WMI level, security is based on standard NT security descriptors, which are essentially Access Control Lists "ACLs" combined with information about the scope of their application. An ACL contains a series of Access Control Entries "ACEs", each of which associates a user or group with a set of rights. In an NTFS file system, ACEs give users or groups rights to read, write, execute, or otherwise tamper with a file. ACEs that form part of the WMI security descriptors give users or groups various rights to manipulate WMI objects within a namespace.

WMI is Microsoft's implementation of the Common Information Model "CIM" which is a vendor-independent standard for describing the hardware and OS components of computer systems and providing tools that a program can use to both read and modify components. The model provides a schema that describes static objects, those common to all computing devices, to which vendors may add dynamic objects, those specific to their products. The Distributed Management Task Force "DMTF", a consortium of PC hardware and network vendors, was originally published CIM in 1996. In 1998, DMTF absorbed the efforts of another initiative, Web-Based Enterprise Management "WBEM", whose goal was the development of Web-based standards for enterprise administration. WMI describes the Windows operating systems, including file systems, event logs, devices, services, hardware controllers, processing and memory. While Microsoft-specific details are included, WMI is based on the CIMv2

Schema, a vendor-independent model of operating environments. Management applications and scripts can be written that take advantage of WMI.

A namespace in WMI is a logical grouping of objects and classes. A SWbemServices object is always associated with a specific namespace, meaning that it can see all classes and objects within that namespace but nothing beyond it. All the standard WMI classes defined as part of the CIM schema live in a namespace called \root\CIMv2, this is usually the default namespace on a system. Other common namespaces include \root\directory (the LDAP namespace), \root\WMI (the namespace where internal WMI objects live), and \root\SECURITY, where objects controlling WMI security live. This deals almost exclusively with objects that live in \root\CIMv2.

4. System Functionalities

4.1. Generalities

The new secure system permits to explore remotely, in real time, a computer connected to a local network (Peer to Peer), from a remotely station, without the user of that computer realizes this exploration. The exploration takes place through a convivial, efficient, and easy-to- use graphical interface and without needing networks or systems knowledge. The exploration happens in a different session, even though the same account is opened on this station. These interfaces possess the same GUI of Windows interfaces.

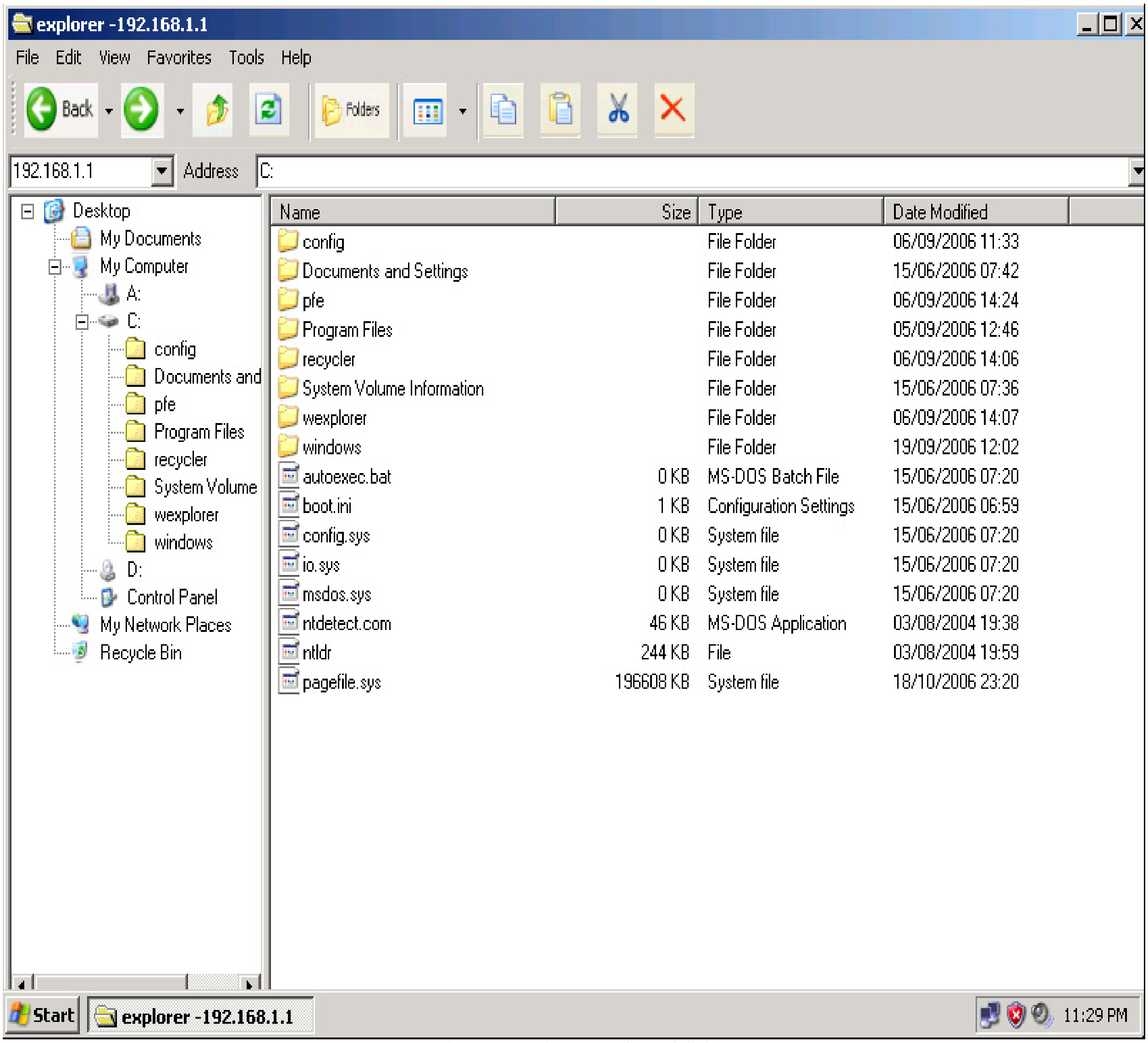
```
Dim myScope  as ManagementScope                                          ‘we create a scope
Dim myQuery as System.Management.ObjectQuery                              ’ we create an object of query
Dim myManagementObjectCollection as ManagementObjectCollection           ’ we create an object collection
Dim myManagementObjectSearcher as ManagementObjectSearcher               ’ we create a Searcher object
Dim nomdemachine as string= "."                                          ‘we specified the address of the station
myScope=newSystem.Management.ManagementScope("\\"&nomdemachine&_ "\root\cimv2")
```

Figure 2. WMI Connection.

The first and the common task before exploring remote computer, is to establish a connection to the service of this computer. The CIMOM allows or no this connection (fig. 2) according to the provided parameters, that correspond to the information of logon to Windows (username and password). First, we create different variables that we will need later. We create a ManagementScope, where more precisely an address on which we will work and to which, we will get data.

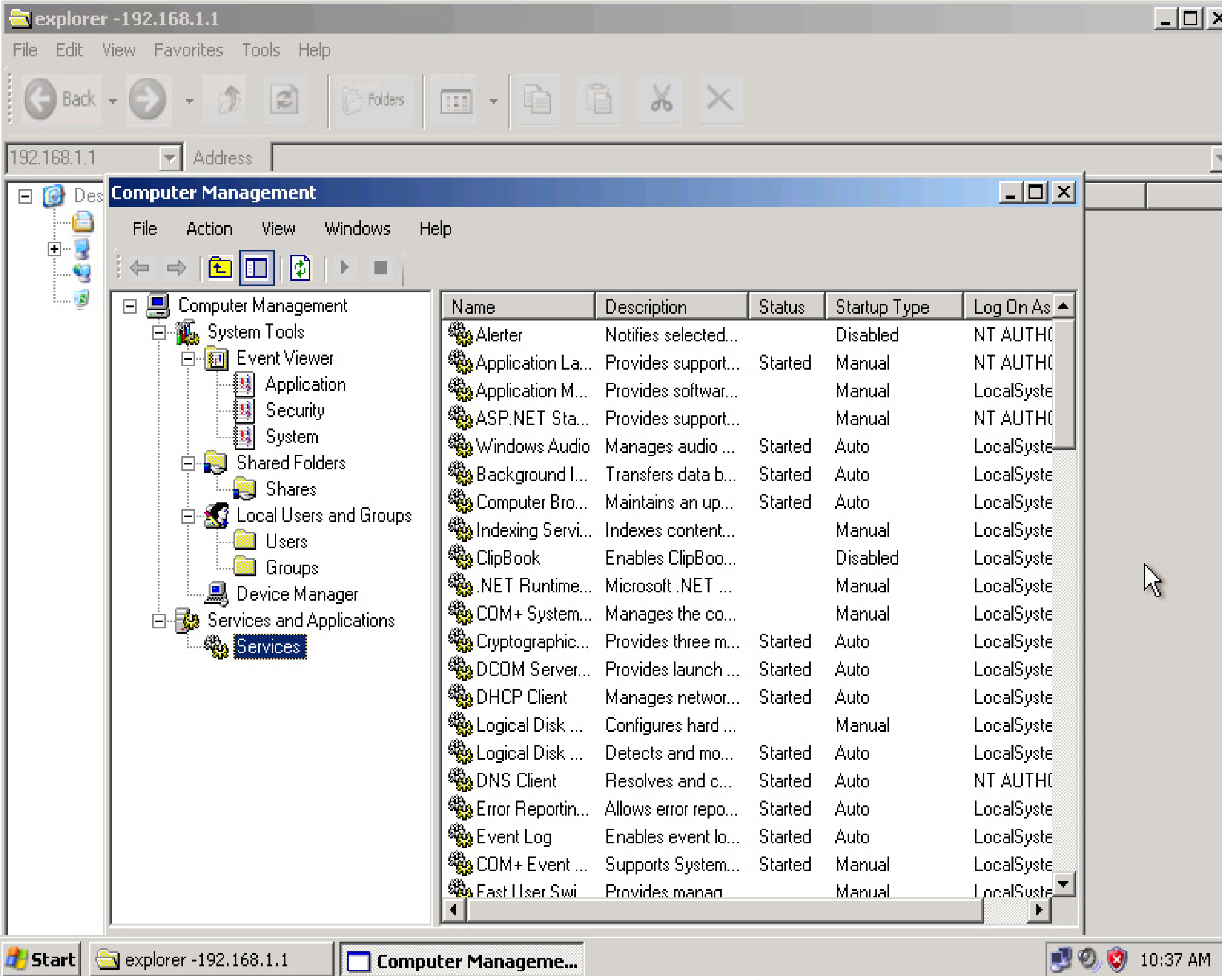
4.2. Explorer

The interface of fig. 3 permits to explore and to manipulate the disk of the Client computer. In the Explorer control, we used the WMI Win32_LogicalDisk class to get all the local and mapped drives on the local machine. To get access to drives information, we need to use the ManagementObjectSearcher class to obtain a ManagementOjbectCollection class containing the drive information we requested. We now have all the available drives' information at our disposal, such as drive name, type, volume, description. When we click on a drive or a directory in the TreeView, we need to check if the drive or directory exists, before proceeding any further. We can get the directories for the current selection in the TreeView by using the Directory class from the System.IO namespace. Calling the Directory.GetDirectories method with the current node path as the parameter, will return an array of directories. We can loop through the directories array to populate sub-nodes under the current selected node. To get the currently selected node's files, we need to call the Directory.GetFiles method with the current node path as the parameter. This will return an array of files for the selected drive or directory. Now we can populate the ListView with the file array by looping through each array element and call FileInfo class to get the file size, creation date, and modified date. The presented information’s in this interface are refreshing regularly. Indeed, this presented Explorer the same form, the same functionalities and the same options of real explorer of a Windows system.



Name	Size	Type	Date Modified
config		File Folder	06/09/2006 11:33
Documents and Settings		File Folder	15/06/2006 07:42
pfe		File Folder	06/09/2006 14:24
Program Files		File Folder	05/09/2006 12:46
recycler		File Folder	06/09/2006 14:06
System Volume Information		File Folder	15/06/2006 07:36
wexplorer		File Folder	06/09/2006 14:07
windows		File Folder	19/09/2006 12:02
autoexec.bat	0 KB	MS-DOS Batch File	15/06/2006 07:20
boot.ini	1 KB	Configuration Settings	15/06/2006 06:59
config.sys	0 KB	System file	15/06/2006 07:20
io.sys	0 KB	System file	15/06/2006 07:20
msdos.sys	0 KB	System file	15/06/2006 07:20
ntdetect.com	46 KB	MS-DOS Application	03/08/2004 19:38
ntldr	244 KB	File	03/08/2004 19:59
pagefile.sys	196608 KB	System file	18/10/2006 23:20

Figure 3. Explorer Interface of disk.



Name	Description	Status	Startup Type	Log On As
Alerter	Notifies selected...		Disabled	NT AUTH(
Application La...	Provides support...	Started	Manual	NT AUTH(
Application M...	Provides softwar...		Manual	LocalSyste
ASP.NET Sta...	Provides support...		Manual	NT AUTH(
Windows Audio	Manages audio ...	Started	Auto	LocalSyste
Background I...	Transfers data b...	Started	Auto	LocalSyste
Computer Bro...	Maintains an up...	Started	Auto	LocalSyste
Indexing Servi...	Indexes content...		Manual	LocalSyste
ClipBook	Enables ClipBoo...		Disabled	LocalSyste
.NET Runtime...	Microsoft .NET ...		Manual	LocalSyste
COM+ System...	Manages the co...		Manual	LocalSyste
Cryptographic...	Provides three m...	Started	Auto	LocalSyste
DCOM Server...	Provides launch ...	Started	Auto	LocalSyste
DHCP Client	Manages networ...	Started	Auto	LocalSyste
Logical Disk ...	Configures hard ...		Manual	LocalSyste
Logical Disk ...	Detects and mo...	Started	Auto	LocalSyste
DNS Client	Resolves and c...	Started	Auto	NT AUTH(
Error Reportin...	Allows error repo...	Started	Auto	LocalSyste
Event Log	Enables event lo...	Started	Auto	LocalSyste
COM+ Event ...	Supports System...	Started	Manual	LocalSyste
Fast User Swi...	Provides manag...		Manual	LocalSyste

Figure 4. Computer management Interface.

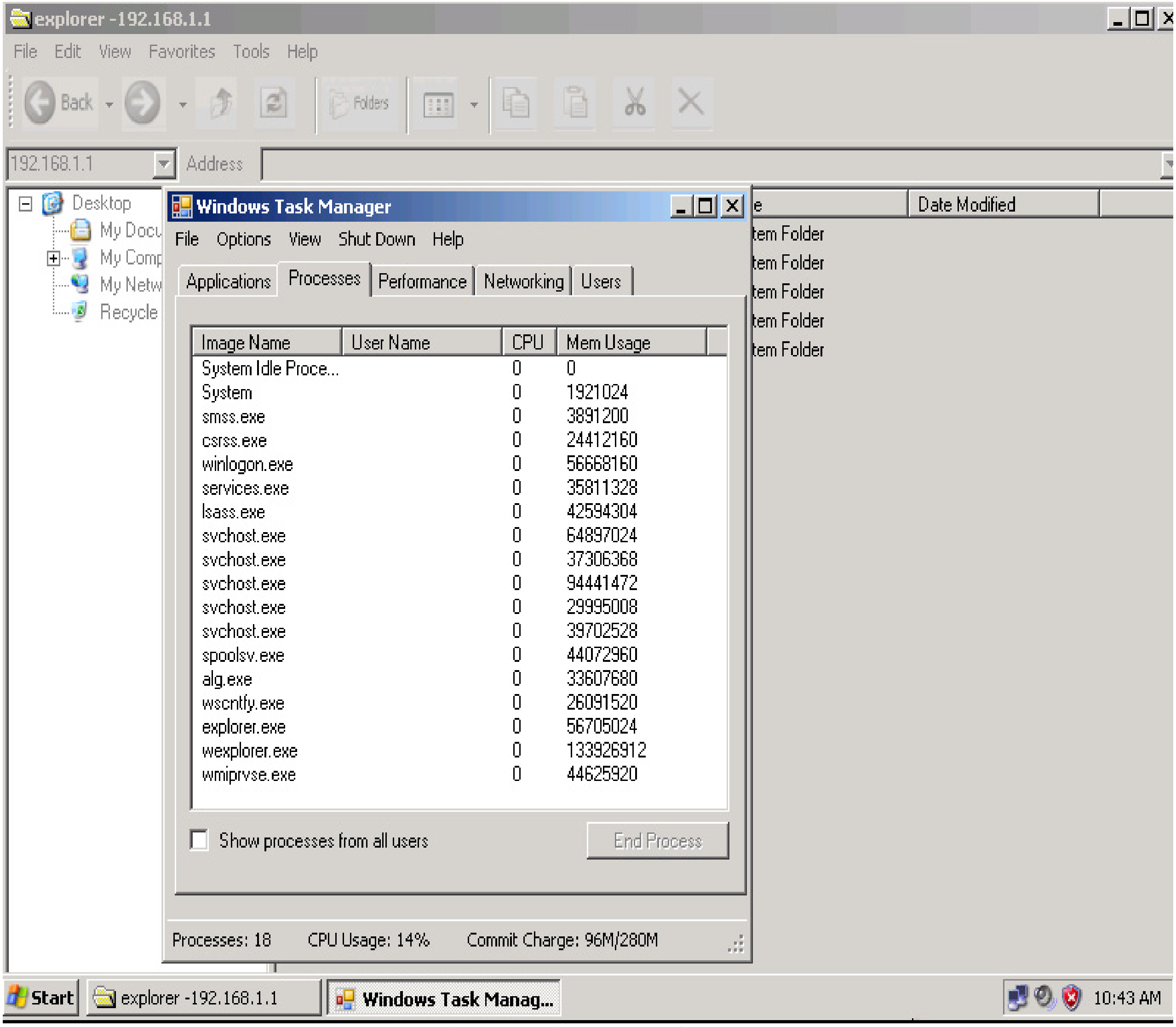


Image Name	User Name	CPU	Mem Usage
System Idle Proce...		0	0
System		0	1921024
smss.exe		0	3891200
csrss.exe		0	24412160
winlogon.exe		0	56668160
services.exe		0	35811328
lsass.exe		0	42594304
svchost.exe		0	64897024
svchost.exe		0	37306368
svchost.exe		0	94441472
svchost.exe		0	29995008
svchost.exe		0	39702528
spoolsv.exe		0	44072960
alg.exe		0	33607680
wscntfy.exe		0	26091520
explorer.exe		0	56705024
wexplorer.exe		0	133926912
wmiprvse.exe		0	44625920

☐ Show processes from all users

Processes: 18 CPU Usage: 14% Commit Charge: 96M/280M

Figure. 5. Task manager Interface.

4.3. Computer Management

The interface of the fig. 4 permits to explore the computer management. The presented information's in this interface are refreshing regularly. The computer management presents the same form, the same functionalities and the same options of a real computer of a Windows system (Compute Management).

4.4. Task Manager

The interface of the fig. 5 permits to supervise and to control the different active applications in progress processes and services of the Client computer. This administrative presents the same form, the same functionalities and the same options of Task Manager of a Windows system. The administrator can also modify the list of processes and applications that the user of the client station is executing. This interface permits to have some useful information on the client computer as: Memory usage, type of Bios, CPU utilization, version of Windows, user's name, computer name, Windows folder, the index System. This interface also permits either to shutdown or to start again the client computer.

5. CONCLUSION AND PRESPECRTIVES

In this paper, we presented a new secure system permitting to control, to manage and to supervise, remotely, in real time, a computer connected to a local network (Peer to Peer). The strengths of this system is to permit the possible utilization remotely, through the local network, of a remotely computer as if one was there, it requires no previous installation on stations to explore. Our software is tested and has delivered the desired level of performance to meet all of the specified requirements for Software verification phase which contains consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed, and provides support for a subsequent conclusion that software is validated. The first of perspectives is the extension of our application so that it will function in a larger context: the Internet. In this case it is necessary to stay up to the security of data that circulates between the customer and the remote computer using a powerful system of encrypting and authentication of remotely secure session.

Acknowledgements

This work has been done as a part of the project "Extraction automatique d'information à partir des documents Arabes".

References

[1] Harrison, B., Lee, K.W. & Rogers, C.M. (2006). Remote execution of software using windows management instrumentation, United States Patent 7035920.
[2] Ismail, A., Hajjar, M. & Hajjar, H. (2008). Remote Administration Tools: A Comparative Study, Journal of Theoretical and Applied Information Technology, pp. 140-148.
[3] Schauer, H., & Baptiste, J., (2006). Windows network services internals.
[4] WMIC: A New Approach to Managing Windows Infrastructure from a Command Line, (2006).
[5] PJ Technologies. (2009). Goverlan Security Information.
[7] Heather Real BMC Software. (2006), http://www.bmc.com/
[8] WMI Explorer 1.00. (2006). http://www.sharewareplaza.com/WMI-Explorer-download_36422.html
[9] Tunstall, C., (2011). Developing WMI Solutions: A Guide to Windows Management Instrumentation.
[10] Jones, D. (2004). Managing Windows® with VBScript and WMI, Publisher: Addison.
[11] Lavy, M., & Meggitt, A. (2011). Windows Management Instrumentation (WMI), 2011.
[12] WMI (Windows Management Instrumentation) : vue d'ensemble de la prise en charge internationale. (2006).
[13] Tunstall C. (2002). Developing WMI Solutions, Publisher: Addison Wesley, ISBN: 0201616130.
[14] Lissoir, A. (2003). Understanding WMI Scripting with Windows and Exchange, Publisher: Butterworth-Heinemann, ISBN: 1555582664.
[15] Ed Wilson, E. (2005). Microsoft® Windows® Scripting with WMI: Self-Paced Learning Guide, Publisher: Microsoft Press, Print ISBN-13: 978-0-7356-2231-9.