Fall 2010

Exploring the Benefits of Buying Industry Leading IT Infrastructure in a Healthcare Setting

Frank Biondolillo
Regis University

Follow this and additional works at: http://epublications.regis.edu/theses
Part of the Computer Sciences Commons

Recommended Citation

This Thesis - Open Access is brought to you for free and open access by ePublications at Regis University. It has been accepted for inclusion in All Regis University Theses by an authorized administrator of ePublications at Regis University. For more information, please contact repository@regis.edu.
Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the “fair use” standards of the U.S. copyright laws and regulations.
Professional Project:

Network Performance Monitor

and Electronic Mail/SMTP Gateway

for the Regis Academic Research Network

Regis University School for Professional Studies
Systems Engineering and Development Practicum, 2006b

Prof. Dan Likarish, Advisor

Steven F. Martin,

2006b SEAD Practicum Team Lead,

Systems and Access Team

May 2007
Acknowledgements

I am grateful to Regis University for permitting me to study and learn under one of the great learning institutions in America. I wish to thank all of the facilitators, professors, and professional staff at the School for Professional Studies for their assistance, patience, perseverance, and dedication to the enrichment of student learning. I couldn’t have done it without you.

I also wish to thank my family, including my wife Erin, sons Matthew, Joshua, Elijah, Samuel, and daughter Sarah Lynn, for their patience during my academic career, and for helping me to see it all through. Without you I could not have completed my objectives, and although my pursuit of higher education had personal and family motivations, I dedicate this honor to you: I shall endeavor to continue the rest of my life in altruistic service.
# Table of Contents

**Acknowledgements** .................................................................................................................... 2  
**Table of Contents** ......................................................................................................................... 3  
**Chapter I – Introduction** .......................................................................................................... 4  
  - Problem Statement ...................................................................................................................... 4  
  - Project Relevance ......................................................................................................................... 4  
  - Barriers .......................................................................................................................................... 5  
  - Proposal and Scope .......................................................................................................................... 6  
**Chapter II – Research** ............................................................................................................. 10  
  - Information Gathering .................................................................................................................. 10  
    - LITERATURE ............................................................................................................................... 10  
    - WORLD WIDE WEB .................................................................................................................... 11  
    - EXPERIENCE ............................................................................................................................. 11  
  - Assumptions and Unknowns ....................................................................................................... 11  
  - Lexicon ......................................................................................................................................... 12  
    - WHAT IS A DOMAIN? .................................................................................................................. 12  
    - WHAT IS EXCHANGE? .................................................................................................................. 13  
    - WHAT IS A MAIL GATEWAY? .................................................................................................... 13  
    - WHAT IS SERVERS ALIVE? ...................................................................................................... 14  
    - WHAT IS A NETWORK MONITOR? ........................................................................................... 14  
**Chapter III - Methodology** ..................................................................................................... 16  
  - Preparation .................................................................................................................................. 16  
    - SOFTWARE REQUIREMENTS: ................................................................................................. 16  
    - HARDWARE REQUIREMENTS: ................................................................................................. 16  
    - PRE-INSTALLATION: .................................................................................................................... 17  
  - Installation of Exchange .............................................................................................................. 19  
  - Configuration of Exchange 2003 ................................................................................................. 43  
  - Installation and Configuration of Servers Alive ......................................................................... 51  
  - Testing ......................................................................................................................................... 83  
**Chapter IV – Analysis** .............................................................................................................. 89  
  - Elements ..................................................................................................................................... 89  
  - Hypothesis ................................................................................................................................... 90  
  - Lessons Learned .......................................................................................................................... 90  
  - Recommendations ....................................................................................................................... 91  
**References** ................................................................................................................................. 95
Chapter I – Introduction

This section discusses the project details including the original problem statement, relevance, barriers, and scope.

Problem Statement

The Academic Research Network (ARN) is in need of a means of monitoring servers and other nodes on the network. That need is driven by the requirement to remotely monitor and support the sprawling network resources between campuses, provide real-time network status reports, and immediate reports of downed nodes. An additional need is electronic mail (e-mail) to serve the practicum students and administrators, but primarily to provide a mail gateway through which Internet mail can be transported in order to provide the aforementioned alerts.

Project Relevance

From the beginning of the author’s experience with the practicum, several server outages were observed with little or no notice. As work commenced on servicing the objectives in the Systems Engineering and Application Development (SEAD) practicum, there was considerable confusion among the various team members as to the availability of assets and their status. It became apparent that more information and control was necessary, as would be the case with any populous network such as the ARN.

As the Academic Research Network (ARN) continues to grow, the requirement for real-time monitoring is coming to the forefront. In each practicum we have students assigned to various roles supporting the enterprise, including roles of server and other asset monitoring and support. The ability to monitor and respond quickly to server
outages is important, and it provides an additional layer of support which can allow a higher service level agreement uptime on the ARN.

The addition of an e-mail service on the ARN brings new potential for other projects also. The mail gateway created by the addition of the e-mail system will bring new capabilities that will be used by the monitoring system, but may also be used for other applications, including student communication.

**Barriers**

There are several barriers observed by the author. The foremost issue is adoption by the practicum. Inherent in the practicum is the constant rate of 100% turnover in key staff positions, which occurs twice annually. The turnover results in a considerable amount of confusion as roles are transitioned to new students and projects become standard systems.

Although this system is being implemented in the SEAD 2006b practicum, its use and growth will be determined by future practicum students. Therefore, adequate documentation and proof-of-concept instruction must be propagated to the appropriate team members which each new practicum, including the 2007a.

The second barrier is the existing infrastructure. Microsoft *Exchange* can require considerable server-based resources, such as CPU and memory utilization, especially when running the *Outlook Web Access* service. For *Exchange* to function properly, it will also require network-based resources including Internet access, and a DMZ (‘demilitarized zone’, a military term used colloquially by the IT community to mean outside the firewall but still within the control of the group) web server in order to be able to access mail from without the ARN. Ideally, the server on which Exchange and the network monitor are
installed would be server-class hardware, or multiple PC-class platforms. In the ‘Recommendations’ section this is discussed much more fully.

The third barrier is cost. While licenses for Exchange are already in possession by the practicum via its MSDN subscription, additional recommended software such as Solar Winds Orion are costly, and must be justified given the mission and expectations of the practicum. Initially, free software will be installed to do the network performance monitoring chores, although only simple tools are available at no cost.

**Proposal and Scope**

The project is broken down into two main sections: Mail System and Monitoring System. The first will deal with the installation and configuration of the mail system software, including the pre-installation preparation. The second concerns the installation of a standard, low-cost network performance monitoring system. Both sections will be followed by thorough proof-of-concept testing.

The scope of the project will be as follows:

1. Prepare the **arn-regis.local forest** and domain for Microsoft Exchange
3. Configure the Exchange installation for use on the ARN, including Outlook Web Access and the SMTP (Simple Mail Transport Protocol) gateway
4. Test all functionality and document results
5. Install Woodstone **Servers Alive** network performance monitor

* Please see Lexicon starting on page 12
6. Configure *Servers Alive* to utilize the maximum of ten nodes (per the free license agreement) and begin monitoring

7. Test all functionality and document results

8. Document the existing system and make recommendations for enhancements

In summary, the proposal introduces a mail gateway system to the ARN and provides a new and immediate ability to scope on critical network nodes and identify status of same. All research, development, and testing will be performed by the author. The total cost of the implementation will be $0 (not including any existing infrastructure costs and requirements). The results of the project will be an immediately functioning server monitoring solution and a solid basis for future expansion on the theme.
The Network Monitor software sends alerts to specified Internet recipients via the Mail (SMTP) Gateway.

Outlook Web Access (OWA) is provided for convenient access to e-mail outside the ARN.

Figure 1.1 – Proposed Function (Microsoft Visio 2003)
The network monitor process polls the specified nodes at a configurable interval.

In the event a configurable condition exists on one of the servers, a pre-configured alert message is dispatched via the SMTP gateway to specified email addresses, which can end up on a mobile device and/or PC-based email client.

Figure 1.2 – Proposed Function (Microsoft Visio 2003)
Chapter II – Research

This section covers the areas of research and information concerning the project-specific requirements.

Information Gathering

This project was conceived in the author’s mind from his own experiences in the Information Technology field. The concept is not new: proactively monitor existing systems in an effort to maximize uptime and minimize recovery time. The e-mail portion is an additional benefit; however the mail gateway provided by the e-mail solution is essential. In researching the concept and how best to apply it to the existing problem on the ARN, various sources were used to bring the concept’s broad strokes into finer resolution.

LITERATURE

There were several sources including books, journals, and periodicals that helped with researching the technical aspects of this project. These consisted of Microsoft training material, ComputerWorld Magazine and NetworkWorld Magazine. The training material was loaned to the author from a colleague who had recently completed an Exchange administration training course. The periodicals were from the author’s own subscriptions.

In particular, an article in ComputerWorld entitled, “How to Choose a Network Management System”, demonstrated the different types of products available, their features and pricing, and benefits. This article did not mention the Servers Alive application ultimately settled on. A very attractive product listed, Solar Winds Orion, is comparatively expensive, but the author’s experiences with the product have proven it to be worthwhile. The OpenNMS solution is perfectly free, and fairly well-equipped, but
requires Linux or UNIX, which would have added additional and unnecessary scope to the project.

**WORLD WIDE WEB**

The bulk of research was done on the vast World Wide Web of knowledge available on the Internet. Many sources were consulted, including sites that specialized in Microsoft *Exchange* deployments, online articles, and technology white papers.

Perhaps the single greatest influence in this area was the Stanford Linear Acceleration Center (SLAC) website, a Stanford University website. This site has a huge repository of information, articles, and links for products related to network performance monitoring. This site is supervised and supported by the ESnet Network Monitoring Task Force (NMTF), another Stanford educational entity, whose mission is primarily WAN monitoring. As near as one can surmise, the SLAC is an educational environment akin to Regis University’s SEAD practicum.

**EXPERIENCE**

Personal experiences with network monitoring and e-mail systems, several different platforms and scales, have all helped the author to build a good conceptual understanding of this project. The author has deployed monitoring tools like HP’s *OpenView* and Solar Winds *Orion* were deployed using SMTP gateways provided by Sun *Solaris* (UNIX) (*sendmail*) and Microsoft *Exchange*.

**Assumptions and Unknowns**

It is assumed that the server hardware provided for this project will be able to sustain the burden of running *Exchange* and the monitoring software. The systems involved are accessible to the point that SNMP (Simple Network Management Protocol)
can be installed on the managed devices, and that security protocols allow a community name (e.g. ‘public’) to be established in order to allow information retrieval from the managed device. SNMP is a common management protocol that allows probing of systemic health in areas such as network, CPU, storage, and memory utilization, as well as status (up or down), and more.

It is unknown whether future practicum will be able to build on the basis of this project by adding a commercial (and much more robust) network performance monitor suite. The project is being pursued on the basis of need and it is assumed it will be maintained going forward, but it is unknown if training and persistence of purpose will be carried forward to future students.

Lexicon

WHAT IS A DOMAIN?

A domain in Internet terminology is the name of an addressable host that can be accessed using a familiar name. An example of a domain is “regis.edu”. The “.edu” portion is a so-called top-level domain, and the “regis” portion is considered the host name of the particular domain and can be considered to be a sub domain of the top-level domain. Other sub domains can exist to facilitate more refined access, such as “arn.regis.edu” and “mail.regis.edu”.

In Microsoft terminology, a domain is a grouping of computers controlled through a specialized server, called a “domain controller”. The domain controller manages all of the various aspects of the active directory (a directory service), DNS (domain name system), servers, workstations, and other aspects of the domain. Windows domains often are synonymous with an Internet domain name, such as “regis.edu” and “arn-regis.org”. Both
of these domain names are Regis Windows domains but also have a corresponding presence on the World Wide Web.

**WHAT IS EXCHANGE?**

Microsoft *Exchange* is a commercial e-mail solution in use for many years in large and small environments. *Exchange* competes with products such as Lotus *Notes* and *cc:Mail*, Novell *GroupWise*, and myriad Linux and UNIX implementations, to name a few.

*Exchange* integrates seamlessly into an Active Directory (*Windows Server*) environment, and provides an easy-to-manage interface and toolset with which to manage mail for the user community. *Exchange* also refers to the server counterpart to the Microsoft *Outlook* client. *Outlook* retrieves mail, and provides other features such as calendar, tasks, notes, contacts, and other productivity aids. With *Exchange*, *Outlook* users can manage multiple mail accounts in one client, as well as create and share calendars and public folders. *Exchange* licenses include the *Outlook* client licenses.

An additional feature of *Exchange* is the *Outlook Web Access* component, which essentially serves up via web server a web-based version of *Outlook* for remote access to mail. As of version 2003 and forward, the OWA client mimics very closely the conventional client software.

**WHAT IS A MAIL GATEWAY?**

A mail gateway, interchangeably called an SMTP gateway, is a software mechanism termed to be a mail transport agent, and is designed to transport e-mail messages between two computers. A practical use of a mail gateway is to send MIME-compatible messages from a third-party application through the gateway out to e-mail accounts or mobile devices with Internet access. Another obvious use is the process of sending e-mail from
within the local area network (LAN) out through the wide-area network (WAN) to computer(s) outside the LAN.

**WHAT IS SERVERS ALIVE?**

*Servers Alive* is a product vended by a company called Woodstone IT Consulting, LLC. It comes in several different licensing models, including a free version that allows ten nodes to be monitored concurrently. SA is a simple monitor system that looks simply at ping response, if desired, or can read more complex information concerning a given server via the SNMP. The other versions provide the ability to monitor more nodes and some additional features. It provides the ability to send notifications of outages via e-mail, instant message, and SMS protocols.

**WHAT IS A NETWORK MONITOR?**

By definition, a network monitor is a combination of hardware and software solutions combining to provide up-to-the-minute information on the performance of specified nodes in a given network. Nodes being monitored can be network switches, routers, servers, storage devices, even workstations. Any device that can respond to SNMP requests can be monitored for performance of duties, including CPU utilization, storage capacity, network traffic intensity, or just a simple heartbeat to show that the device is responding. Simply stated, anything approaching configured thresholds for a given node should be reportable to a technical resource for remediation (Orloff 2006).

A network monitor wouldn’t be of much use without the ability to send alerts and other reports out to the desired distribution concerning the nodes being monitored. For example, whenever a server goes down, the technical services group of an organization
would be notified instantly of the outage and can respond in kind. Further, more complex network monitors issue reports on performance of the nodes, for example reporting the top ten servers with high CPU utilization, or low disk space, and so forth.

It should be noted that monitoring can be performed in a variety of ways, using different protocols. SNMP is a protocol designed specifically for this purpose, but other methods include using PING to test the response of a network interface (alive or not), or checking websites via http or https request methods, for example.

Network monitors as discussed in this project do not include monitoring for overall network performance (i.e. bandwidth utilization), or malicious or errant traffic (such as virus or hacker intrusions), as such tools have their own category and usually go by names such as Network Traffic Monitors and Intrusion Detection Systems (IDS).
Chapter III - Methodology

In this chapter we discuss the actual implementation process for the project, including all steps leading up to, and following installation of the chosen software.

Preparation

To begin, an evaluation of the various servers available is necessary in order to make sure we meet Microsoft’s recommended hardware minimums for installation of Exchange. From Microsoft (Microsoft 2005), the requirements are as follows:

SOFTWARE REQUIREMENTS:
- Domain controllers are running Windows 2000 Server Service Pack 3 (SP3) or Windows Server 2003.
- Global catalog servers are running Windows 2000 SP3 or Windows Server 2003.
  The global catalog is a distributed data repository that contains a searchable, partial representation of every object in every domain in a multi-domain Active Directory forest.
- Domain Name System (DNS) and Windows Internet Name Service (WINS) are configured correctly.
- Servers are running Windows 2000 SP3 or Windows Server 2003 Active Directory.

HARDWARE REQUIREMENTS:
- Intel Pentium or compatible 133 megahertz (MHz) or faster processor
- 256 megabytes (MB) of RAM recommended minimum, 128 MB supported minimum
- 500 MB of available disk space on the drive on which you install Exchange
- 200 MB of available disk space on the system drive
• CD-ROM drive

• SVGA or higher-resolution monitor

These last two items, the CD-ROM and the SVGA monitor are really optional since we can load *Exchange* directly from a disk-based location, and many servers do not have a monitor attached at all, nor require one. Both conditions are true in our case. Indeed, the server we have chosen meets the hardware requirements, and the ARN domain environment is also satisfactory.

**PRE-INSTALLATION:**
Before we can even begin the installation process there are certain prerequisite software that must already exist on the server. They are ASP.net (web interface), NNTP (Network News Transport Protocol, used for newsgroups), and SMTP (Simple Mail Transport Protocol, used for transportation of mail across the Internet). Our server does not have these installed by default, so we must install them.

1. In Control Panel, we go to Add/Remove Programs (or just type `appwiz.cpl` in the `Start > Run` box) and select Add/Remove Windows Components. We are presented with the following interface:
2. We choose Application Server and click Details, where we are presented with our first element to install, ASP.NET:
3. Select ASP.NET and Internet Information Services, and with IIS highlighted, we click Details.

![Internet Information Services (IIS)](image)

4. Here we make sure NNTP and SMTP are checked and then click OK. We will be prompted for the Windows Server CD or disk files. Once these items are installed we’re ready to move to the main installation phase for Exchange.

**Installation of Exchange**

Although Microsoft Exchange 2003 comprises a complex installation, Microsoft provides a deployment tool to aid in the process. It includes a checklist allowing the administrator to procedurally move through the eight installation steps. Microsoft recommends that the administrator download and use the latest deployment tool from their website at http://www.microsoft.com/downloads/details.aspx?familyid=271e51fd-fe7d-42ad-b621-45f974ed34c0&displaylang=en. Doing so provides the most recent
version which was posted in September 2004. The deployment tool is really just a
collection of help files presented via an .HTA (HTML Application) page. It does not
require installation and runs from the folder into which it was extracted. When we launch
the exdeploy.hta file, we see the following:

![Exchange Server Deployment Tools](image)

We select “Deploy the first Exchange 2003 server” to begin the appropriate
checklist, as shown in the next screenshot.
On this screen we want to select “New Exchange 2003 Installation” because this is our first mail installation on the domain. The next dialog shows us the checklist we will work from to complete the Exchange 2003 installation.

---

**Deploy the First Exchange 2003 Server**

To begin, select the appropriate deployment process based on your current environment:

<table>
<thead>
<tr>
<th>If this is your current environment</th>
<th>Follow this process</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are running Exchange 5.5 and have not yet connected Exchange to Active Directory.</td>
<td>Coexistence with Exchange 5.5</td>
</tr>
<tr>
<td>You are running Exchange 2000 and Exchange 5.5, and you are using Active Directory Connector (ADC) to synchronize the directories.</td>
<td>Coexistence with Mixed Mode Exchange 2000 and Exchange 5.5</td>
</tr>
<tr>
<td>You are running Exchange 2000 in native mode and you want to upgrade a server or install the first new Exchange 2003 server.</td>
<td>Upgrade from Exchange 2000 Native Mode</td>
</tr>
<tr>
<td>You are not running Exchange.</td>
<td>New Exchange 2003 Installation</td>
</tr>
</tbody>
</table>
Notice we have already checked off the first two because these relate to the minimum requirements already discussed. We are ready for step 3, which points to Windows support tools already existing on our server. If they were not installed, we can download them from Microsoft here:


By default the Support Tools package will install to c:\program files\ and we must add this to the PATH system variable under My Computer > Advanced > Environmental Variables. Once we have verified the installation of these tools, we can check off Step 3 and move to Step 4.
Open a command prompt, Start > Run: cmd <enter>, and type the following command:

dcdiag /f:c:\dcdiaglog.txt /s:dc0larnregis.arn-regis.local

The command calls the Support Tool \texttt{DCDIAG} to assess that the domain controller passes our pre-installation testing. The output of the command is redirected to a text file as noted in the command arguments. The output is as follows:
Performing initial setup:
   Done gathering initial info.

Doing initial required tests
   Testing server: Default-First-Site-Name\DCOLARNREGIS
     Starting test: Connectivity
               .................. DCOLARNREGIS passed test Connectivity

Doing primary tests
   Testing server: Default-First-Site-Name\DCOLARNREGIS
     Starting test: Replications
               .................. DCOLARNREGIS passed test Replications
     Starting test: NCDesc
               .................. DCOLARNREGIS passed test NCDesc
     Starting test: NetLogons
               .................. DCOLARNREGIS passed test NetLogons
     Starting test: Advertising
               .................. DCOLARNREGIS passed test Advertising
     Starting test: KnowsDomainHolders
               .................. DCOLARNREGIS passed test KnowsDomainHolders
     Starting test: KdcManager
               .................. DCOLARNREGIS passed test KdcManager
     Starting test: MachineAccount
               .................. DCOLARNREGIS passed test MachineAccount
     Starting test: Services
               .................. DCOLARNREGIS passed test Services
     Starting test: ObjectsReplicated
               .................. DCOLARNREGIS passed test ObjectsReplicated
     Starting test: Tpssysvol
               .................. DCOLARNREGIS passed test Tpssysvol
     Starting test: TrsEvent
               .................. DCOLARNREGIS passed test TrsEvent
     Starting test: KceEvent
               .................. DCOLARNREGIS passed test KceEvent
     Starting test: SystemLog
               .................. DCOLARNREGIS passed test SystemLog
     Starting test: VerifyReferences
               .................. DCOLARNREGIS passed test VerifyReferences

Running partition tests on : ForestDNSZones
   Starting test: CrossRefValidation
               .................. ForestDNSZones passed test CrossRefValidation
   Starting test: CheckSDRefDom
               .................. ForestDNSZones passed test CheckSDRefDom

Running partition tests on : DomainDNSZones
   Starting test: CrossRefValidation
               .................. DomainDNSZones passed test CrossRefValidation
   Starting test: CheckSDRefDom
               .................. DomainDNSZones passed test CheckSDRefDom

Running partition tests on : Schema
   Starting test: CrossRefValidation
               .................. Schema passed test CrossRefValidation
   Starting test: CheckSDRefDom
               .................. Schema passed test CheckSDRefDom

Running partition tests on : Configuration
   Starting test: CrossRefValidation
               .................. Configuration passed test CrossRefValidation
   Starting test: CheckSDRefDom
               .................. Configuration passed test CheckSDRefDom

Running partition tests on : arn-regis
   Starting test: CrossRefValidation
               .................. arn-regis passed test CrossRefValidation
   Starting test: CheckSDRefDom
               .................. arn-regis passed test CheckSDRefDom

Running enterprise tests on : arn-regis.local
   Starting test: Intersite
               .................. arn-regis.local passed test Intersite
   Starting test: FsmoCheck
               .................. arn-regis.local passed test FsmoCheck
With no errors seen we now check off Step 4 and move to Step 5.

Once more we will use the command line to execute the Support Tool **NETDIAG**, which will make an assessment of the networking environment. We type the following command:

```
netdiag >c:\netdiaglog.txt
```
The command line argument redirects output to a text file. When we view the text file for the results, we see the following:
COMPUTER NAME: EXO1ARMREGIS
DNS Host Name: exo1armregis.arn-regis.local
System info: Microsoft Windows Server 2003 (Build 3790)
Processor: x86 Family 15 Model 1 Stepping 2, GenuineIntel

Netcard queries test . . . . . . . : Passed

Per interface results:

Adapter: Local Area Connection

Netcard queries test . . . . . . . : Passed

Host Name . . . . . . . . . . . . : exo1armregis.arn-regis.local
IP Address ................. : 192.168.1.72
SUBNET Mask ............. : 255.255.255.0
Default Gateway ......... : 192.168.1.252
DNS Servers ............... :
192.168.1.71
192.168.1.143
192.168.224.1
205.171.2.65
205.171.3.65

Auto Configuration results . . . . . . : Passed

Default gateway test . . . . : Passed

NetBT name test ............. : Passed

Global results:

Domain membership test ........ : Passed

NetBT Transports test ........ : Passed

List of NetBT transports currently configured:

NetBT_TCP\Ip={22A39C8D-678C-4FDB-85B8-2CB11026B8F2}
1 NetBT transport currently configured.

Autonet address test ........... : Passed

IF loopback ping test ........... : Passed

Default gateway test ........... : Passed

NetBT name test ............... : Passed

Winsock test ................. : Passed

DNS test ..................... : Passed

Redirect and Browser test ....... : Passed

List of NetBT transports currently bound to the Redirect
NetBT_TCP\Ip={22A39C8D-678C-4FDB-85B8-2CB11026B8F2}
The redirect is bound to 1 NetBT transport.

List of NetBT transports currently bound to the browser
NetBT_TCP\Ip={22A39C8D-678C-4FDB-85B8-2CB11026B8F2}
The browser is bound to 1 NetBT transport.

DC discovery test .............. : Passed

DC List test .................. : Passed

Trust relationship test ........ : Passed

Secure channel for domain 'ARN-REGIS' is to '\\dc02arnregis.arn-regis.local'.

Kerberos test .................. : Passed

LDAP test ..................... : Passed

Bindings test .................. : Passed

WAN configuration test ........ : Skipped

No active remote access connections.

Modern diagnostics test ........ : Passed

IP Security test ............... : Skipped
As with the other Support Tool, if we encounter any errors in the log, we resolve them and then continue. In a clean, working environment we should not encounter any problems. Check off step 5 and move to Step 6.

Step 6 irreversibly prepares the forest in which the Exchange server will reside by extending the Active Directory schema with new classes and attributes germane to Exchange 2003. In general we always want to do installations like this under the auspices of an account with membership in the ‘Domain Administrators’ group, but at a minimum this tool must be run under a domain account that is a member of the ‘Schema Admins’ group.

As we continue, we enter the path to the Exchange 2003 setup file and click the link for “Run ForestPrep now”. When running the ForestPrep tool we receive a warning about
the current version of Windows:

We receive the warning because we are running Windows Server 2003 Service Pack 1 or later, which is incompatible with the non-service-packed version of Exchange 2003. We will be installing the Exchange Service Pack 2 later.

We check “Don’t display this message again”, and then “Continue”. We click “Next” on the introductory screen, then click “Next” to accept the Microsoft EULA (End User License Agreement).
The next screen above will become familiar as we enter different stages of installation. In this stage we are installing ForestPrep, as noted in the ‘Action’ column, and we are installing to the default location of \c:\program files\exchsrvr. We click “Next” to continue.

On the next screen we choose the default Exchange administrator account. Since we are logged in as the domain Administrator, we accept the default and click “Next”.
As the ForestPrep routine runs, we will see progress bars as above. Once the process is complete we click “Finish” and return to the Deployment Tool checklist.
In step 7 we are going to prepare the domain for the *Exchange* deployment in the same manner we did the forest earlier. We enter the source for the *Exchange* setup files again, same as for step 6, and click the ‘Run DomainPrep now’ link.

We click “Next” on the introductory screen, then click “Next” to accept the Microsoft EULA (End User License Agreement).
In this stage we are installing **DomainPrep**, as noted in the ‘Action’ column, and we are installing to the default location of `c:\program files\exchsrvr`. We click “Next” to continue.
As the DomainPrep routine runs, we will see progress bars as above. Once the process is complete we click “Finish” and return to the Deployment Tool checklist.

We have now completed all of the preparatory steps prior to the installation of Exchange itself, and we are now ready for the final step in the Deployment Tool checklist.
Finally, we are on to installing *Exchange* itself. We enter the path to the source files as before and click the “Run Setup now” link. We click “Next” on the introductory screen, then click “Next” to accept the Microsoft EULA (End User License Agreement).
On this sheet we examine the options a bit more and make sure we are installing only what we need. The defaults are fine for our installation so we click “Next” to proceed.
On this sheet we select “Create a new Exchange Organization” because we don’t have one yet. On the next sheet we’ll enter the name of our organization.
We enter the appropriate name, “ARN-Regis”, and proceed. This name will be visible in the Exchange System Manager and all Exchange servers will be placed under this organization name.
On this sheet we are confirming that we have the appropriate number of per-seat licensing for Exchange; click “Next”.
We are now presented with the last sheet before the installation begins. We click "Next" to proceed.
We watch as installation proceeds. It can take some time depending on system speed and conditions.
Eventually, the installation completes. At this point we click “Finish” and close the Deployment Tool checklist. We are now ready to install the Exchange 2003 Service Pack 2 to bring the installation current. The service pack can be downloaded from Microsoft here: http://www.microsoft.com/downloads/details.aspx?FamilyId=535BEF85-3096-45F8-AA43-60F1F58B3C40&displaylang=en.

The installation of the service pack is fairly straightforward with no configuration or administrative choices to make. Once the installation of Service Pack 2 is complete, we are ready to begin configuration.
Configuration of Exchange 2003

There are many configuration possibilities for Exchange 2003, for many different environments and needs, including multi-server, multi-domain environments. Fortunately for basic operation there are only a few things we have to address.

We begin by opening the primary configuration tool, the Exchange System Manager, on the Exchange server. Doing so displays the following Microsoft Management Console snap-in:

We right-click on the topmost object, the ARN-Regis organization object, and select “Properties”.
On this sheet we have the option of switching the Operation mode from mixed mode to native mode. Microsoft recommends that we change to native unless we plan on supporting existing Exchange 5.5 installations. Going to native mode enables features not available unless we do so, including the ability to move mailboxes between servers, and so forth. Primarily, however, the move is designed to provide maximum stability. We click on “Change Mode” and are presented with the following dialogue:
We click “Yes” to proceed, and “OK”. This is an irreversible process and is instantaneous.

Once again we right-click on the topmost object, the ARN-Regis organization object, and this time we select “Internet Mail Wizard”.

With this tool we can configure our environment to be able to send and receive Internet mail via our Exchange installation. By default, from our existing installation we can send mail through our Internet gateway but we cannot receive mail. Since this scope of this project is not concerned with e-mail per se, we will skip this step. Essentially, however, configuring this option will require us to have an Internet IP address assigned to
our *Exchange* server, or setup a secondary *Exchange* server in a network DMZ (demilitarized zone).

We next are ready to begin adding eligible user accounts for Exchange. This is important because we need an account from which to send alerts and also to prepare for the ability to send and receive mail between internal participants.

We open the Active Directory Users and Computers MMC to begin administering *Exchange* at the user level. We select a user by right-clicking on the name, and then select “Exchange Tasks”. The first time this is run on a user we see the following dialogue:
We choose the “Create Mailbox” task and click “Next”.
We enter the alias, in this case mimicking the AD account name, and click “Next”.
Inevitably we should see the success sheet shown above. We click “Finish” to exit the dialogue.
We can now see several Exchange-related tabs in our Properties sheet that allow us to further customize Exchange usage at the user-level, though most configurable options would be handled at the group level, such as attachment limitations, mailbox sizes, and so forth. No further configuration is necessary at this point.

We have completed the installation and configuration of Exchange 2003 in our monitoring environment and we are now ready to move on the next phase.
Installation and Configuration of *Servers Alive*

Our chosen monitoring system, *Servers Alive* is a software tool that is easy to install, simple to configure, and free. Of course, there are limitations to free in that the free version allows us to monitor only ten nodes. If we need to monitor more than that, we have to pay for the commercial license, which is still quite reasonable at $199, or $299 for a fully featured version that allows us to monitor up to 5,000 nodes. Still, the limited information we can gather from *Servers Alive* leads to us to believe we would be better off with a richer monitoring tool like Solarwinds *Orion*. We will discuss this at greater length in the next chapter.

We begin by downloading the latest version of Servers Alive from the Woodstone IT Consulting website, at http://www.serversalive.com/salive/download.asp. As of this writing the current release is version 6, and this is what we will be installing.

We download the small 15mb installation pack and execute, accepting the very remedial settings as default throughout the brief installation. It should take less than a minute to complete the installation and then a reboot of the host server is required. When we come back, we begin the setup and configuration.
Upon returning from boot we will see a shortcut on the desktop for *Servers Alive*. Launching it produces the interface shown above, and we can also see an orange tray icon has been installed (in the lower right corner). When we minimize *Servers Alive*, it is minimized to the tray rather than the taskbar, reducing clutter. Clicking once on the tray icon will again produce the interface.

The next few steps involve configuring servers to be monitored, and then users and groups to alert when a server is down. To begin adding nodes we want to monitor, we click the “Add” button and we are instantly presented with the “Entries” interface.
On this interface we will walk through the various tabs and set up each server we want to monitor. On the General tab, we enter the server name (DNS) or IP address. Notice that we can also monitor Novell Netware servers if we had any.
For our first server we enter our first server of import, the domain controller. We add in the fully qualified domain name of the server, although we could have simply entered the short name without the domain suffix. The “Pretty Name” field is for what will be shown in monitor alerts and should usually be descriptive. No other entries are required on this tab, though we entered a Remark to be thorough.
Next is the “Check” tab. On this tab we configure what methodology we prefer in monitoring our nodes. We have seventeen different options here, including SNMP, URL (for checking the status of websites), Ping, and so forth. In truth we can configure this monitoring tool to use several different methods of monitoring a single node, thereby providing multiple points of detection, such as disk space running low, websites going down, or other decision support elements. For our configuration we will use simple Ping, since we want to detect when a server goes down and comes up. With this free version we can only configure ten host Entries, but we can configure many different alert types per Entry.

Clicking on a given methodology offers different configuration options for that method. For example:
Monitoring for an NT service running on the host will produce options such as which service to monitor and the ability to authenticate into the host as required.

There are two checkboxes for additional options on each of the check configuration options. Each is unchecked by default. The “Invert Status” checkbox is available on all of the check configuration options. This option reverses the status at the end of the check; for example, the status would be reversed from up to down or from down to up. The “Second Knock” option is also available on all of the check configuration options. If a host is seen as down during a check cycle and you have clicked the check box, the software rechecks the host at the end of the check cycle to verify that the host is down.
Monitoring the host for disk space allows us to establish a threshold and alert when the available disk space falls below it.
For purposes of this project we shall deploy the **Ping** method. We can configure the size of the TCP/IP packets (5 frames) sent to the host, large or small, and we can configure how many of the requests (80%) must come back within the specified timeout period (5 seconds). This can afford us the ability test not only if a server is alive, but bandwidth too, if we configure the packets to be larger and increase the percentage of return-required frames. Notice we have selected the option to take a second look at a host that fails the check, to be sure it is really down and not a victim of network traffic.
On the “Alert” tab we will configure who will receive alerts and when. We begin by clicking the “ADD” button on the left side of the panel.
We are again presented with many options for alerting on the Check we are configuring. These include sending SMTP mail, SMS (Simple Messaging System) messages which can be received on cellular telephones and handheld devices, network pages, numeric pages, web page postings, instant messages through popular agents like ICQ, MSN and AIM, and more. For our project we have devised the multi-faceted SMTP server and we will use that methodology.
A new interface appears and we can configure in great depth who will receive alerts and when, and any constraints that override our base settings. On the first tab we are shown the message information, such as to whom the mail will be sent, the subject line with variables (%p is the “Pretty Name” and %s is the status), and an accompanying message body with the same variables or any more information we want to present to the recipient based on how we are configuring this alert.
By clicking on the “Add” button next to the “To:” form field, we are presented with options to send to either a “Person” or “Team”. In the SEAD we are all about teamwork so we select “Team”.

![Add/edit alert window](image)
In the “TEAM” dialogue we are presented with any teams we have configured, or the default. We’ll choose default for now and configure teams later.
Moving to the next tab, “When”, we can configure what circumstances are required before sending an alert. For simplicity we leave the default option chosen, “When 1 times down”, which means that when it is detected only once that a host has gone down the alert will be given.
The third and final tab, “Schedule” is for configuring when the alert will be allowed in terms of hours and days. The text on this sheet state that this is only available in the ENTERPRISE version, but also works in the FREE version. A bit contradictory, but we don’t really need to configure this sheet because we want alerts any time there is a problem.

With the Add/edit Alert dialogue completed, we click “OK” to return back to the “Entries” panel.
And we can now see our alert has been configured and is in place. The next five tabs allow additional configuration of our Entry that are not required. They include posting Entry alert history on a web page, logging, scheduling of the Entry monitoring similar to the scheduling interface on the Alerts panel, custom responses, and network routing options. None of these are relevant to our project so we can click the lower “Add” button to complete our Entry.
We have returned to our main interface, and we see that we have an Entry! We must now configure the method we chose for delivery of alerts, SMTP, and then we'll add users and teams to our roster of alert recipients.

To begin, we click on “File” from the toolbar at the top of the panel and select “Setup”. We will be presented with a large interface with a multi-configuration tree.
Clicking the “+” next to any of these items expands the tree to reveal more detail and many options. We’ll expand the “Alerts” branch, then the “SMTP” branch, and then the “Primary” branch. This provides the interface shown above. Here we will specify the “Mail host”, which in this case is the same machine we are working on, that is our Exchange server. We specify the “From” address as something generic and descriptive. Whether we create the mailbox we specify here is irrelevant. We will not be expecting return mail from the outgoing alerts. We have to be sure to check the “Enable primary SMTP mail” box as this is the option we chose when configuring the alert. We can choose to test the mail gateway by clicking the “Test it” button. A test message is sent and we check our inbox to see the message arrive.
Success! This means our mail gateway and our configuration of Servers Alive is working up to this point.

Still on the “Setup” panel, we can configure an alternate mail gateway to send mail through, however for purposes of this project we have but one. There is only one more option we need to configure before moving to users and teams.
We move down to the “General” branch and expand it to reveal several more options. We’re interested in the “Startup settings” and clicking on this option reveals the panel above. We check the box for “Load startup info from registry” and include a hostfile where we will be saving all of our Entries and Alerts later. We choose a simple filename stored in the root drive, “C:salive.txt”. We click “OK” and we’re done with setup.
Now back at the main screen again, we click on “Edit” from the toolbar and we see a dropdown list of options. Locate “People” near the bottom and select it.
From installation we have one person created, “default”. We begin to add alert recipients by clicking the “Add” button next to the “Select person” field.
We type the name of the new alert recipient in the box provided and click “OK”.

When we return to the People panel we can see our new person has been added.
We check the box for “Person is active” to make this person able to receive alerts. If this person goes on holiday or no longer wishes to be in the alert group, we can simply uncheck this box without removing the recipient altogether. We next configure the default alert delivery mechanism for our project, the SMTP tab.
On this tab we enter the person’s e-mail address and can select the schedule for which the person is on-shift. In this case we leave the entire schedule green for “Send”. There are many configurable options in this software that make it fairly well-rounded for the price. We are finished with this user so we click “OK’ to close out the panel, and we are taken back to the main Servers Alive screen. From here we continue to follow the preceding steps until we have added all of the recipients desired.
Our next step will be to add the alert recipients we just added to one or more teams and then add those teams to our alerts. From the main screen we click “Edit” from the toolbar and select “Teams”.

The first thing we notice is in the “People” box we have the alert recipients we configured in the preceding steps. We will use these to populate our new team.

From installation we have only one team configured: default. We could configure this team but we would rather create our own so that we can establish multiple teams for different functions going forward. To do this we click the “Add” button and a dialogue appears.
We will type the name of our new team in this box, SEAD, and then click “OK” to finish. We can now see in our “Name” box the team we just added. We’ll now add people to our team.
We add people from the left box to the right “Member of the team” box by highlighting a name and clicking on the arrow pointing toward the “Member” box. Also note that it is important we check the “Active team?” box or this team will never receive alerts. We click “OK” to close the dialogue, and we are taken back to the main *Servers Alive* screen.

Getting nearly done now we have to go back into the first Entry we added and add the right team to the alert. We do this in the same manner as when we originally set up the Entry and alerts. When finished we will be returned to the main page. We need to follow all of preceding steps pertaining to adding Entries to add the other servers we want to monitor. When we have finished, we’ll have a screen full of Entries.
For each of the Entries shown above, an alert was created to notify the team “sead” whenever one of the specified hosts went down, and when it came back up again (two alerts per host).

Next we want to make sure we do not lose all of this configuration information, so we will save our work in the “salive.txt” file we specified in the “Setup” section earlier.
We perform the save by clicking on “File” from the toolbar and then “Save as”. We then get the explorer browse dialogue to specify the filename and location.
We type in our filename, `salive.txt`, and click “Save”. We are then returned back to the main *Servers Alive* panel. We have now completed the installation and configuration of our *Servers Alive* monitoring tool. We must click the “Start” button on this panel to begin the actual monitoring process. Once we start it, we can minimize the application and it will disappear to the tray, however it continuously monitors at the standard interval of 5 minutes (configurable).
Testing

The testing process is fairly straight-forward. During the installation of our *Servers Alive* monitoring tool, we learned that our *Exchange* mail gateway was working. We now only need to test if the alerts work as promised, delivering information about downed and resumed servers. Following are several e-mail alerts sent during the testing cycle:

The SQL server went down.
One minute later the server was found to be up. Because we are double-checking the down status of our hosts during the Servers Alive check process, we can confirm that the node was down for the better part of one minute, and resumed. This is presumed to be a deliberate reboot of the SQL server.
The MS Project server is discovered to be down.
Just like the SQL server, this machine went down and came up within a minute, presumably from a deliberate reboot.
The SQL server again.
And approximately three minutes later, the SQL server is back up and running, according to our alerts.

All in all, the testing is successful. We have established that our continuous server monitoring is producing the desired alerts by sending messages out via our SMTP gateway server.
Chapter IV – Analysis

In this chapter we will discuss the project in retrospect, examining the elements of the project, reviewing the hypothesis which begat the project, and looking at lessons learned during the execution.

Elements

The elements of this project include the network architecture (hardware and software), Microsoft Exchange, Servers Alive, and very importantly, the response team to who alerts and management reports will be sent.

The network architecture consists of hardware and software assets, including various x86 (microprocessor architecture) Dell servers running the Microsoft Windows Server 2003 operating system. Ultimately, as specified, little change to existing network infrastructure and architecture was necessary. The changes that did take place to the existing architecture included permanent modification of the arn-regis.local forest and domain to prepare for Exchange, and the addition of the SMTP server itself.

The two new software components installed were Microsoft Exchange 2003 and Woodstone Servers Alive. The Microsoft product was purchased and licensed through the MSDN program subscription for the University. Microsoft Exchange 2003 licenses include licenses for the Microsoft Outlook mail client, although the most likely utilization will occur via the Outlook Web Access medium, a web-based mail client. The Servers Alive product is a free license product from Woodstone Consulting, although the product is limited in the number of nodes that can be monitored.

The third and perhaps most important element of this project are the human resources that will receive and react to the alerts transmitted by the project. Currently
there are three individuals (the program facilitator and two practicum students) who have been configured to receive alerts. Naturally, in order to provide qualitative support, this abbreviated user group must be expanded to include members of the new practicum on an ongoing basis. An appendix document, “Configuring Servers Alive”, has been assembled with contains the majority of the setup instructions for Servers Alive, and this can serve as documentation for the system for any new participating practicum students.

**Hypothesis**

The hypothesis that a monitoring system was required and could be installed at no cost to the University has been proven. Testing revealed that the monitoring system presented an accurate portrayal of monitored network nodes and their status. Post-installation regression testing consisted of rebooting all affected servers on the domain and attempting previous functionality such as logging in, sharing files, accessing web pages hosted by affected servers, and use over time. As of this writing the implemented changes have been in production for approximately six weeks will no negative effects reported.

**Lessons Learned**

There were two types of learning throughout this project. First, the author’s fairly broad experience did not include the performance of a complete installation of Microsoft Exchange 2003, or Servers Alive. Much consultation with printed and electronic resources resulted in a successful implementation. Practical learning through implementation almost always leaves artifacts behind that aid in future, even unrelated implementations.
Second, empirical knowledge was gained about how to establish objectives for a project of this scope and stick to those objectives. In several instances there were opportunities to expand the project to implement features beyond the original scope, and while those features might have added contributed in a positive way, they would have needlessly occluded the basic purpose behind the project.

**Recommendations**

In conclusion to the project, there are several recommendations to enhance the project.

1. **Solarwinds Orion Network Performance Monitor**

   This tool, actually used in production and preferred by the author, is a very robust network performance monitoring tool. It is relatively inexpensive with license and maintenance coming in around $2500 to monitor 100 nodes. The *Orion* product provides detailed information about every node being monitored, including current CPU/network/storage utilization, alerts coming from the operating system, and of course status. Just like *Servers Alive*, it sends alerts whenever certain thresholds are crossed, such as running out of disk space, the node going down, and so forth. The chief difference between the installed product and this recommended product is the interface and capabilities. The *Orion* suite is configured via an easy-to-use graphical user interface on its host server, but then, via web pages, serves up all of the content in graphs and images that are perfect for immediate problem identification.

   Some screenshots of *Orion* in action:
Fig 4.1 - This is the web-based Top-10 list of problems by area.
Fig 4.2 - This is a drilled-in view of one of the servers being monitored.

The *Orion* tool provides a tremendous amount of visibility to the infrastructure of an organization to afford its users with a high degree of proactive information. Administrators can respond to problems before they become crises and even see systemic issues before they are really such.

2. Establish Full-Use Internet Mail for Students

With the capabilities of Microsoft *Exchange* to allow users to receive nearly the full functionality of Microsoft *Outlook* via a web page, and the cost being so low to maintain, it could be a great enhancement to allow practicum student access to the e-mail system.
They could use it to communicate on practicum business wherever they are located, and use its common calendaring and tasks features to help their team stay on track.

There are several items that must be addressed before this can take place. First, the mail server must be configured to deliver mail via the external-facing domain name, which is arn-regis.org. Second, a second Exchange server running Outlook Web Access must then be configured to talk to the primary Exchange server built during this project. This server would sit on the external domain and relay communications back and forth. Finally, the practicum would need to establish usage guidelines and policies for the use of the mail system by the practicum students, and establish procedures for maintenance of mailboxes and the systems by the practicum.
References


http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9000849


http://www.quepublishing.com/articles/article.asp?p=418014&rl=1
