

Monitoring heterogeneous environments

(Version 1) Using Perl and XML to use Web pages to monitor heterogeneous hosts and to obtain configuration information from them

Olutoyin Oshin, Angel Rivera

TR 29.3459

DISCLAIMER:

This technical report is not an official publication from the e-Business Tools support group. The authors are solely responsible for its contents.

e-Business Tools Technical Support Team
IBM Software Solutions
Research Triangle Park, North Carolina, USA

© Copyright 2001, IBM Corp.

Monitoring heterogeneous environments

Monitoring heterogeneous environments

ABSTRACT

This technical report provides a methodology, Perl scripts and an XML file that system administrators can implement to have in a single simple web page (in table format), predetermined configuration information from selected heterogeneous hosts, such as status of the host (if not available, then the keyword FAILED appears in red), the actual version of the operating system, the amount of available disk space, and if enabled, a link to a file (whoAmI.html) that provides more configuration details.

The method uses available tools such as IBM HTTP Server or Apache HTTP Server, scripts written in Perl, an XML file with the list of hosts to be monitored and information provided in HTML. Because the scope of this method is simple, there is no need for sophisticated system management tools and the associated complexity and long learning curve.

This technical report is the first version of this methodology and tools. While developing this methodology, we identified some improvements that we want to include in a future second version.

ITIRC KEYWORDS

- Configuration information
- System administration
- Web
- Perl scripts

Monitoring heterogeneous environments

ABOUT THE AUTHORS

Olutoyin Oshin

Mr. Oshin is an software engineer for e-business tools technical support. He currently provides level 2 and 3 support to customers of WebSphere Studio.

Mr. Oshin started with IBM in 2000 and worked as an application developer, then he provided technical support for IBM VisualAge TeamConnection.

Mr. Oshin has a B.S. in Computer Science from North Carolina State University at Raleigh, and is currently a M.S. student in Computer Science at North Carolina State University.

Angel Rivera

Mr. Rivera is an advisory software engineer and is the serviceability architect and RAS focal point for IBM e-Business Tools.

He joined IBM in 1989 and since then had worked in the development and support of library systems. He was the technical lead for CMVC Version 2 for many years. He was also the team lead of the technical support team for VisualAge TeamConnection.

Mr. Rivera has an M.S. in Electrical Engineering from The University of Texas at Austin, and a B.S. in Electronic Systems Engineering from the Instituto Tecnologico y de Estudios Superiores de Monterrey, Mexico.

Monitoring heterogeneous environments

ABSTRACT	- 3 -
ITIRC KEYWORDS	- 3 -
ABOUT THE AUTHORS	- 4 -
Introduction	- 8 -
Motivation behind this methodology	- 8 -
Structure of this technical report	- 9 -
Why not simply use Tivoli or other systems management tools?	- 9 -
Caveats	- 10 -
Disclaimer	- 10 -
How to get the most up to date version of this technical report	- 10 -
Example of a working solution	- 11 -
Accessing the URL that has the table of hosts and their information	- 11 -
Grouping the hosts by platform	- 12 -
Finding out details of specific hosts via the file whoAml.html	- 12 -
Why are some hostnames underlined?	- 12 -
Finding out the version and the operational status of a host	- 13 -
Finding out the details of the disk space in a host	- 15 -
Summary of the processing to generate the table	- 16 -
Summary of the characteristics of the table	- 16 -
Table entry when a Unix host is available	- 16 -
Table entry when a Windows host is available	- 16 -
Table entry when a host is not available	- 16 -
Installation of the prerequisites for both server and client hosts	- 17 -
Web Server (IBM HTTP Server)	- 17 -
Perl	- 17 -
Perl (ActivePerl for Windows)	- 17 -
Note about possible configuration error in Windows	- 18 -

Monitoring heterogeneous environments

Store the scripts and files from this TR in the cgi-bin directory of the Web Server - 19 -

Installation of additional prerequisites in the server

host - 21 -
(Windows only) OPTIONAL: Windows Services For Unix (SFU) - 21 -
nmake for Windows - 21 -
GNU tar for Windows - 21 -
GNU gzip for Windows - 22 -
WinZip or PKZip for Windows - 22 -
Info-zip for Windows/Unix - 22 -
Perl modules - 22 -
Edit hostnames.xml to include all client hosts - 23 -
Use a Web Browser to invoke the main home page for the tool - 24 -

Description of the Perl scripts and the XML file - 25 -

Description of hostnames.xml - 25 -
Description of mhe.pl - 25 -
Description of diskSpaceWin.pl - 26 -
Description of versionWin.pl - 26 -

How to add a client host into the web page - 27 -

Only for Unix clients: create a user id (webcfg) for the automated telnet session - 27 -
Important information on the prompt - 27 -
Editing the XML file "hostnames.xml" - 27 -
Detailed description of the contents of hostnames.html - 28 -
Preparing the whoAml.html file - 29 -

Debugging suggestions - 31 -

How to get the Perl scripts mentioned in this document . - 32 -

FTP site for TeamConnection - 32 -
Obtaining Info-ZIP - 32 -

Monitoring heterogeneous environments

How to unzip files	- 32 -
Copyrights, Trademarks and Service marks	- 33 -

Monitoring heterogeneous environments

Introduction

This technical report provides a methodology, Perl scripts and an XML file that system administrators can implement to have in a single simple web page (in table format), predetermined configuration information from selected heterogeneous hosts, such as status of the host (if not available, then the keyword FAILED appears in red), the actual version of the operating system, the amount of available disk space, and if enabled, a link to a file (whoAmI.html) that provides more configuration details.

The method uses available tools such as IBM HTTP Server or Apache HTTP Server, scripts written in Perl, an XML file with the list of hosts to be monitored and information provided in HTML. Because the scope of this method is simple, there is no need for sophisticated system management tools and the associated complexity and long learning curve.

This technical report is the first version of this methodology and tools. While developing this methodology, we identified some improvements that we want to include in a future second version.

Motivation behind this methodology

Before implementing the methodology described in this technical report, the following was a common scenario in our group: we use several heterogeneous hosts (Windows, AIX, HP-UX, Solaris, and Linux) and frequently we needed to know:

- If these hosts were active: for this task we used the “ping” command.
- Which is the actual level of the operating system for a given host: for this task we used “telnet” or direct usage of the console.
- How much free disk space is available.
- What software products are installed.

In the case of some of the Windows machines, we do not have a Telnet server in all of them, and thus, we needed to walk to another building in order to interact directly with these hosts. We wanted to save time and effort in the activities to monitor these hosts, and therefore, we have a strong set of motivations for developing the methodologies and the scripts described in this document:

- To have in a single simple web page the overall view of all the machines that we have in our group: their host names, their basic operating system, their operational status, etc.
- To reduce the amount of “pings” to be issued, just to find out if the host is operational.
- To reduce the amount of “telnet” sessions to be issued, just to execute basic commands such as finding the level of the operating system and the available file disk space.
- To reduce the walking between buildings in order to access Windows machines and issue simple queries.

Monitoring heterogeneous environments

- To facilitate the displaying of configuration information. For example, finding out the amount of RAM in a Unix machine could be tricky because there are no universal commands in the Unix variations that provide this information; thus, once the system administrator finds this information, it would be nice to place it in a file which can be accessed by a web browser.

The sample scripts were developed in Perl. The reason for using Perl is that it is a scripting language that is available in both Unix and in Windows. If we used only Korn shell scripts in Unix, then we would not have a solution that would include Windows. We decided to use XML to provide the list of hosts to be monitoring because it is a very convenient way to represent the information needed for the scripts (easily describe the data, define the tags with the data and parse/extract the data).

Structure of this technical report

The structure of this document is the following:

- Chapter “Example of a working solution” explains the methodology and the scripts, by describing the table that the end-user will see in a web browser, and how to interact with the different elements from that table.
- Chapter “Installation of the prerequisites for both server and client” explains some important notes related to the installation and the configuration of the common prerequisites for the host that will be the server, as well as the hosts that will be clients accessed by the server.
- Chapter “Installation of the prerequisites in the server host” explains additional notes related to the installation and the configuration of the prerequisites for the host that will be the server.
- Chapter “Description of the Perl scripts and the XML file” explains all the Perl scripts used in the solution and the XML file that has the list of hosts to monitor.
- Chapter “How to add a client host to the web page” describes how to add a new host name into the central table that is displayed to the end users.
- Chapter “How to get the Perl scripts mentioned in this document” explains how to use FTP to get the tools mentioned in this document.

Why not simply use Tivoli or other systems management tools?

There are some systems management tools, such as Tivoli, that provide a wealth of configuration information. The main drawbacks are:

- There are complex to install.
- There are complex to configure.
- They take considerable system resources to execute and deliver information.
- They are expensive.

Monitoring heterogeneous environments

The proposed methodology described in this technical report has the following advantages that answer each one of the drawbacks mentioned above:

- It is easy to install (for a person who is somewhat familiar with the Web server that is being used at their site).
- It is easy to configure and an advanced system administrator can expand the functionality.
- The execution overhead is minimum and it only happens when the web page is accessed for the first time or it is refreshed (on demand). That is, we do not need to have constant feedback on the status of the hosts, simply by refreshing the page being displayed by the web browser, we could get the status on the hosts listed in that page.
- It uses existing applications at the customer site (in case that they already have a Web server) or uses shareware (such as Perl).

Caveats

In our first attempt in writing the tools and this technical report, we wanted to prepare a general solution in which the server Perl scripts would work in both Windows and AIX (we wanted the primary web page to reside in an AIX server).

However, we discover that there is a big difference in the cross-platform implementation of certain Perl modules, and unfortunately, these required modules did not install properly in AIX. We were a bit disappointed because we thought that by using Perl, we could rely on cross-platform functionality of publicly available Perl modules. We did not try other non-Windows platforms.

Because of these unexpected problems, we had to reduce the scope of this technical report to provide a solution in which the server Perl scripts are located only in a Windows server machine.

Disclaimer

This technical report is not an official publication from the e-Business Tools support group. The authors are solely responsible for its contents.

How to get the most up to date version of this technical report

The most up to date version of this technical report can be obtained from the following IBM VisualAge TeamConnection Enterprise Server web site:

<ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/trweb-mhe.pdf>

Monitoring heterogeneous environments

Example of a working solution

This chapter provides to the reader a “taste” of an example of a working solution of the methodology, the sample Perl scripts and the XML file. These scripts are described in detail in the rest of this document.

Accessing the URL that has the table of hosts and their information

First, let’s assume that the web page for this example is ready and all prerequisites are present. The end user will use a web browser and specify the URL that will have the main Perl program (or click on a link already defined for this URL), such as:

```
http://your-server-name/cgi-bin/MHE/mhe.pl
```

Where “your-server-name” is the host name that handles the web server.

WARNING: It may take few minutes for this page to be displayed. The reason is that the current implementation tries to access ALL the hosts in the list before displaying the complete table. This waiting time includes any time-outs for those hosts that are not responding.

A sample graphic snapshot of the table in the web browser is shown below:

Monitoring of heterogenous environment

Platform	Hostname	Version	Diskspace
SOLARIS	oem-sn12	FAILED	Show Disk Space
SOLARIS	oem-sn13	SunOS 5.6	Show Disk Space
AIX	suppserv	AIX 4.3	Show Disk Space
AIX	tcaix02	FAILED	Show Disk Space
AIX	tcaix08	FAILED	Show Disk Space
HP-UX	oemk250a	HP-UX B.10.20	Show Disk Space
HP-UX	oem-hp14	HP-UX B.11.00	Show Disk Space
WINDOWS	tcbuid1	Windows NT Version 4.0	Show Disk Space
WINDOWS	tcbuid2	Windows NT Version 4.0	Show Disk Space
LINUX	taiwo	FAILED	Show Disk Space

Monitoring heterogeneous environments

Grouping the hosts by platform

The first column (called “Platform”) indicates the generic name for the operating system or platform. In our example we have AIX, HP-UX, Solaris, Linux, and Windows hosts.

Finding out details of specific hosts via the file whoAmI.html

The second column (called “Hostname”) indicates the host name. In our example, we chose to show the short name (“tcaix02”) and not to show the fully qualified name (“tcaix02.raleigh.ibm.com”).

Why are some hostnames underlined?

In this column, the host names that are active are shown with an underlined font, which represents a hypertext link. If the host is not active, then there is no purpose in providing an unreachable link and thus, it is shown as plain text. For example, if the user clicks on the host name “oem-sn13”, the web browser will show the contents of the file whoAmI.html which resides in the host oem-sn13; this is a file that is manually created and maintained by the system administrator to display specific information about the host. In this case, the link is for the page:

<http://oem-sn13/whoAmI.html>

By the way, the above file is actually located in the oem-sn13 host in:

`/opt/GoServer/usr/internet/server_root/pub/whoAmI.html`

However, due to the different web servers that we have, we wanted to have an easier way to get to this file. Thus, the “root” user in each host had created a symbolic link at the root level:

```
$ cd /
$ ln -s /opt/GoServer/usr/internet/server_root/pub/whoAmI.html /whoAmI.html
$ ls -dl whoAmI.html
whoAmI.html -> /opt/GoServer/usr/internet/server_root/pub/whoAmI.html
```

An example of the contents of such page is shown below:

Configuration information for host "oem-sn13"

Date last updated (manually): 19-Feb-2001

Location of this file: /opt/GoServer/usr/internet/server_root/pub/whoAmI.html

```
Host name:          oem-sn13
  IP Address:       9.x.y.z
  Primary use:      Development/testing for VA TC

Operating System:  Solaris 2.6 (SunOS 5.6)
Manufacturer:      Sun
Type/Model:        Enterprise 250
Serial Number:     OEM1 580297B
RAM:               512 MB
Disk:              8 GB
```

Monitoring heterogeneous environments

```
DB2 UDB 5.2:
  version:      db2level
                DB21085I Instance "db2inst1" uses DB2 code release "SQL05020" with
                level identifier "02090103" and informational tokens "DB2 v5.2.0.33",
                "c990904" and "U465808".
  home:        /opt/IBMDB2
  DB2 instances:
    db2inst1
    db2as
```

```
Java (JDK):
  version:     java -version
              1.1.8
  home:       /opt/java/jdk1.1.8
```

```
Lotus Domino Go WebServer,
  version:    4.6.2
  home:      /usr/internet -> /opt/GoServer/usr/internet
```

```
Netscape:
  version:   4.5 (en)
  home:     /opt/netscape
```

Notes:

- This file contains some pieces of information that are hard to find via commands, such as the model of the hardware, the serial number, etc.
- The whoAmI.html file could be created automatically during run-time; that is, this file might be generated by a script running in the target host and presented to the web browser when requested. However, there are many commands to process and many different results to parse and format.
- Because the scope of this method is to keep it simple, we decided to create and edit this file manually.

Finding out the version and the operational status of a host

The third column (called “Version”) indicates 2 things:

(1) The specific version of the operating system:

- In Unix, the command “uname -a” is used, and the output is parsed appropriately to get the name of the operating system, the version number and the release number. The output in AIX is different than HP-UX and Solaris, thus, we need to take these differences into account.

* AIX: In an AIX host, the output of “uname -a” could be:

AIX oem-ppc3 2 4 000022559000

Then, the items in underlined font are parsed and the following string is returned:

AIX 4.2

* HP-UX: In an HP-UX host, the output of “uname -a” could be:

HP-UX oemk250a B.10.20 A 9000/800 268748331 two-user license

Monitoring heterogeneous environments

Then, the items in underlined font are parsed and the following string is returned:
HP-UX B.10.20

* Solaris: In a Solaris host, the output of “uname -a” could be:

SunOS oem-sn13 5.6 Generic_105181-05 sun4u sparc SUNW,Ultra-250

Then, the items in underlined font are parsed and the following string is returned:
SunOS 5.6

Notice that to SunOS 5.6 is really Solaris 2.6. Furthermore, the new Solaris 7 is SunOS 5.7 and the new Solaris 8 is SunOS 5.8.

* Linux: In a Linux host, the output of “uname -a” could be:

Linux taiwo 2.2.16-22 #1 Tue Aug 22 16:49:06 EDT 2000 i686 unknown

Then, the items in underlined font are parsed and the following string is returned:
Linux 2.2.16-22

- In Windows, the command “ver” is used, the complete output, such as:

Microsoft Windows 2000 [Version 5.00.2195]

Windows NT Version 4.0

(2) The operational status of the host:

- If the host is operational, then the text in this column will be shown with a green color (similar to the green in a traffic light, which has the meaning of: Go ahead).

- If the host is not operational, then the keyword FAILED will be shown in red and in upper-case (similar to the red in a traffic light, which has the meaning of: Stop).

FAILED

- If the host is AIX and the password has expired, then the following will be displayed:

PASSWORD EXPIRED

Monitoring heterogeneous environments

Finding out the details of the disk space in a host

The fourth column (called “Disk Space”) provides a link to a script that will gather the available disk space for a host. For example, if the user clicks on "[Show Disk Space](#)" for the AIX host “tcaix08”, then something gets executed in the target host (this will be explained later) and the web browser displays the output of the AIX command that shows the information about the disk usage of the file systems in that host (df -k):

tcaix08

Results from the command: df -k

Filesystem	1024-blocks	Free	%Used	Iused	%Iused	Mounted on
/dev/hd4	65536	47592	28%	1308	4%	/
/dev/hd2	2523136	177508	93%	62222	10%	/usr
/dev/hd9var	81920	68108	17%	283	2%	/var
/dev/hd3	180224	174224	4%	147	1%	/tmp
/dev/hd1	4276224	261956	94%	10789	2%	/home
/dev/lv02	8388608	3843992	55%	134	1%	/bakup
/dev/lv03	6537216	1689932	75%	10890	1%	/home4

By the way, in this case, the specific link to “Show Disk Space” for the host tcaix08 is:

http://server-host-name/cgi-bin/MHE/mhe.pl?diskspace_SN_AIX=tcaix08

The command “df -k” provides the same output in AIX and in Solaris, but a very different and rather useless output in HP-UX. Thus, when dealing with HP-UX systems, we have to use a different command that provides a more useful output: bdf. For example, if the user clicks on "[Show Disk Space](#)" for the HP-UX host “oem-hp14”, then something gets executed (this will be explained later) and the web browser displays the output of the HP-UX command that shows the information about the disk usage of the file systems in that host (bdf):

oem-hp14

Results from the command: bdf

Filesystem	kbytes	used	avail	%used	Mounted on
/dev/vg00/lvol3	83733	22663	52696	30%	/
/dev/vg00/lvol11	67733	23614	37345	39%	/stand
/dev/vg00/lvol18	480341	282393	149913	65%	/var
/dev/vg00/lvol17	518549	420644	46050	90%	/usr
/dev/vg00/lvol14	49861	38649	6225	86%	/tmp
/dev/vg00/patch	159509	111385	32173	78%	/patches
/dev/vg00/lvol16	698133	605732	22587	96%	/opt
/dev/vg00/lvol15	797845	581329	136731	81%	/home

By the way, in this case, the specific link to “Show Disk Space” for the host oem-hp14 is:

http://server-host-name/cgi-bin/MHE/mhe.pl?diskspace_HP=oem-hp14

Monitoring heterogeneous environments

Installation of the prerequisites for both server and client hosts

This chapter explains some important notes related to the installation of the prerequisites in the server host and in the client hosts for implementing the solution described in this document.

You may need to install a Web Server and Perl in the client hosts in order to use URLs to handle web requests, such as servicing the whoAmI.html page that describes certain characteristics of the hardware and software of the machine.

Web Server (IBM HTTP Server)

For a Web Server, we use the IBM HTTP Server, which is based on the Apache Web Server.

* Download the IBM HTTP Server (the particular version that we used was 1.3.12.2) for Windows from the following web site:

<http://www-4.ibm.com/software/webservers/httpservers/download.html>

* Install the IBM HTTP Server.

In Windows, do the following:

- Locate the downloaded HTTPServer_us.exe file and execute it.
- The code will be installed in: C:\Program Files\IBM HTTP Server

* Start the IBM HTTP Server.

Perl

The scripts provided with this technical report are written in Perl.

Perl (ActivePerl for Windows)

For Perl, we use ActivePerl for Windows.

* Visit the web site:

<http://www.activestate.com/Products/ActivePerl/Download.html>

Or

<http://aspn.activestate.com/ASPN/Downloads/ActivePerl/>

* Download the appropriate Perl installation file. Notice that at the time we wrote this document, this was the latest available version.

"ActivePerl-5.6.1.629-MSWin32-x86-multi-thread.msi"

Monitoring heterogeneous environments

* If not using Windows 2000, then you need to download the following auxiliary software:

For Windows 95/98, it is required you download InstMsi.exe from the following web site:
<http://download.microsoft.com/download/platformsdk/wininst/1.1/W9X/EN-US/InstMsi.exe>

For Windows NT, download InstMsi.exe from the following web site:
<http://download.microsoft.com/download/platformsdk/wininst/1.1/NT4/EN-US/InstMsi.exe>

* Install ActivePerl in Windows, and ensure the following:

- In the Custom Setup window of the installation, specify the “Location” directory to be C:\usr (the default is C:\Perl)
- In the Choose Setup Options window of the installation, specify to add Perl to the PATH and to associate the file extensions for Perl.

* After the installation, open a new command prompt and check the value of the PATH variable and make sure that it includes the directory “C:\usr\bin”, because this is the directory where Perl.exe will be located.

* It is recommended to reboot your Windows system after you install Perl.

* Ensure that the first line of your Perl scripts should be the following, to guarantee its proper execution:

```
#!/usr/bin/perl
```

The reason for using the above specification is to have portable Perl scripts that can be executed in both Windows and Unix platforms. Thus, it is important to install in Windows the Perl executables in c:\usr instead of c:\perl.

* Note if you install Perl in a driver other than C.

By default, the expression “/usr/bin/perl” in Windows means “c:\usr\bin\perl.exe” by the web server. However, if you install the Perl code in d:\usr (for example), then the execution of the Perl scripts will fail when invoked by the web server. To overcome this problem, you will need to change the first line of the Perl scripts to be:

```
#!d:/usr/bin/perl
```

* Verify the installation of Perl by performing the following:

```
C:  
Cd \usr  
Cd eg  
Perl example.pl
```

The result of running the example.pl script should be:

```
Hello from ActivePerl!
```

Note about possible configuration error in Windows

Monitoring heterogeneous environments

If you failed to install ActivePerl in c:\usr and failed to customize the first line of the Perl scripts, then you will be unable to execute .pl or .cgi scripts and you will get the following errors in the error.log file:

No such file or directory: couldn't spawn child process:

In this particular case, when trying to execute the mhe.pl script, the web browser will say:

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, you@your.address and inform them of the time the error occurred, and anything you might have done that may have caused the error.

More information about this error may be available in the server error log.

There will be entries in the error.log of the Web Server (c:\Program Files\IBM HTTP Server\logs\error.log) such as:

```
[Mon Aug 27 10:57:46 2001] [error] [client 9.x.y.z]
(3)The system cannot find the path specified: could not run script interpreter:
/usr/bin/perl -w "c:/program files/ibm http server/cgi-bin/mhe/mhe.pl"
```

```
[Mon Aug 27 10:57:46 2001] [error] [client 9.x.y.z]
(2)No such file or directory: couldn't spawn child process:
c:/program files/ibm http server/cgi-bin/mhe/mhe.pl
```

Note that you will get the above error if you install Perl in a driver other than C (such as in D:\usr). To overcome the problem you will need to modify the first line of the scripts to be:

```
#!d:/usr/bin/perl
```

Store the scripts and files from this TR in the cgi-bin directory of the Web Server

Now that the Web Server is time to store in the cgi-bin directory structure of the Web Server, the scripts and files from this technical report. See the section "How to get the Perl scripts mentioned in this document" on how to download and unpack these files:

The following files are only used by the server host:

```
mhe.pl
hostnames.xml
```

Monitoring heterogeneous environments

The following file is only used by Windows client hosts (they should be placed also in the server host, because the server should also be a client):

```
diskspaceWin.pl  
versionWin.pl
```

The following file should be placed in all clients:

```
whoAmI.html
```

Once those files are unpacked, you need to copy them into the appropriate cgi-bin subdirectory for the Web Server. In our case, we create a directory named cgi-bin/MHE and under this directory we store the files described in this document.

* For Windows, when using the IBM HTTP Server, store the files in:

```
mkdir C:\Program Files\IBM HTTP Server\cgi-bin\MHE
```

```
Copy the files into C:\Program Files\IBM HTTP Server\cgi-bin\MHE
```

* Copy the whoAmI.html file into the DocRoot directory, such as:

```
Windows: C:\Program Files\IBM HTTP Server\htdocs\whoAmI.html
```

```
Unix: /usr/HTTPServer\htdocs/en_US/whoAmI.html
```

Monitoring heterogeneous environments

Installation of additional prerequisites in the server host

This chapter explains some important notes related to the installation of additional prerequisites in the server hosts. This is in addition to the prerequisites mentioned in the previous chapter.

Due to installation problems with some required Perl modules in AIX, the server hosts described in this chapter are only Windows hosts.

(Windows only) OPTIONAL: Windows Services For Unix (SFU)

The new Windows Services for UNIX (SFU) Version 2, English CD, has a telnet client, a telnet server, over 60 Unix utilities such as cat, grep, ls, ps, and vi natively from Windows NT or Windows 2000.

For specifications, see its home page at: <http://www.microsoft.com/windows2000/sfu/>
To buy it, visit: <http://shop.microsoft.com>

nmake for Windows

The nmake (Make) is needed in Windows in order to compile the Net-Telnet utility (shown later).

- * Download the following file and store it as “nmake.exe” from the following web site:
<ftp://ftp.microsoft.com/Softlib/MSLFILES/nmake15.exe>
- * Select a directory that is included in your PATH and store the utility there.

GNU tar for Windows

The GNU tar utility is used to unpack (untar) files that have a .tar suffix.

- * This utility can be downloaded from:
<http://www.gnu.org/software/tar/tar.html>
- * At the time we wrote this technical report, the following URL had the tar file for Windows:
<ftp://ftp.gnu.org/gnu/tar/tar-1.12.msdos.exe>
- * Select a directory that is included in your PATH and store the utility there; ensure that you rename it to “tar.exe”.

GNU gzip for Windows

Monitoring heterogeneous environments

The gzip utility is needed to uncompress files that have a .gz suffix.

* GNU gzip to handle the compression (but not the unpacking) can be downloaded from:
<http://www.gnu.org/software/gzip/gzip.html>

* Select a directory that is included in your PATH and store the utility there.

WinZip or PKZip for Windows

The gzip can only uncompress the .tar.gz file, rendering it as a .tar file. Now you need to use another utility to unpackage the .tar file. Some options are:

A) WinZip can be used to unpack the uncompressed files that have a suffix of .tar or .zip. Its web page is:

<http://winzip.com/>

B) PKZip can be used to unpack the uncompressed files that have a suffix of .tar or .zip. Its web page is:

<http://www.pkware.com/>

Info-zip for Windows/Unix

Another great zipping utility is Info-Zip. You can take advantage now to download it. For more information see the chapter "How to get the Perl scripts mentioned in this document".

Perl modules

There are several Perl modules that are used in our solution, and it is necessary to download them from www.cpan.org.

+ The Net::Telnet module for Perl is needed to allow Perl scripts to make client connections to a TCP port and perform network input/output, especially to a port using the TELNET protocol. The objective is to access a host via telnet and obtain information from that host via normal commands.

* Download the following Perl Modules from the following web site:

<http://www.cpan.org/modules/01modules.index.html>

This URL could be a bit hard to use. You can search for the category, author and full file name shown below:

module/ category	author/ maintainer	current distribution file	size	uploaded
---------------------	-----------------------	---------------------------	------	----------

Monitoring heterogeneous environments

Net JROGERS Net-Telnet-3.02.tar.gz 30k 27 May 2000

* Uncompress and untar the files

One way to use both the gzip and tar commands to uncompress and unpack files with suffix of .tar.gz (also known as .gtar or .tgz) is the following:

1) Uncompressing the file, which result in a file with a suffix of .tar. The gzip utility only uncompresses; it does not unpack:

```
gzip -d file.tar.gz
```

2.a) Unpacking the file, which will result in a directory Net-Telnet-3.02 with several files. Note: there might be some warnings about changing file permissions.

```
tar -xvf file.tar
```

2.b) Another choice is to use either WinZip or PKZip to unpack the tar file.

* Install the Net::Telnet module in Windows:

```
cd Net-Telnet-3.02
perl Makefile.PL
nmake
nmake install
```

Edit hostnames.xml to include all client hosts

The file “hostnames.xml” is a file that provides all the names and additional data of the client hosts that will be shown by the home page. A template for this file is provided with the zip file with the Perl scripts for this technical report.

The hostnames.xml file needs to be located in the cgi-bin\MHE directory in the Web Server; this is the location for the Perl scripts developed for this technical report. If using IBM HTTP Server, then the location is:

```
C:\Program Files\IBM HTTP Server\cgi-bin\MHE
```

For details on the format of the file “hostnames.xml”, see the next chapter “Description of the Perl scripts and related files“.

Use a Web Browser to invoke the main home page for the tool

Monitoring heterogeneous environments

The main home page for this tool is ready to be used!

Open a web browser and enter the below URL:

`http://your-server-name/cgi-bin/MHE/mhe.pl`

Where your-server-name is the hostname of the machine.

Monitoring heterogeneous environments

Description of the Perl scripts and the XML file

This chapter provides a brief description of the Perl scripts and the XML file that are used to implement the solution described in this technical report:

Description of hostnames.xml

The XML file “hostnames.xml” contains the central table with hostnames, login and password id’s, and platform types.

This is the only file that needs to be edited to add, delete, or modify hostnames and there properties. A sample contents is shown in the next chapter “How to add a host to the central table“.

Description of mhe.pl

This is the main Perl script. It reads the contents of the XML file hostnames.xml (which contains the central table of hosts), obtains the hosts to be monitored and the appropriate properties and performs the following actions:

- 1) When the application is launched, for each hostname it performs the following:
 - 1.a) If the “os” attribute of the host (platform type) is of Unix system (AIX, HP-UX, SOLARIS or LINUX):
 - * The script issues a “telnet” command to the appropriate host, using the login and password id specified in hostnames.xml. Once the telnet session is established, the script executes the “uname -a” command to retrieve the version of the operation system.
 - * If the “Show Disk Space” link is clicked, then the Unix command “df -k” (for AIX, SOLARIS and LINUX) or “bdf“ (for HP-UX) is executed to retrieve the disk space.
 - 1.b) If the “os” attribute (platform type) is WINDOWS:
 - * The script accesses the host via a URL, and executes the “ver” command.
 - * If the “Show Disk Space” link is clicked, then executes the URL corresponding to the diskspaceWin.pl Perl script residing in the client Windows host to retrieve the characteristics of the disk space in that client.
- 2) Retrieves the result of the execution of the appropriate command, parses the result, and displays the appropriate strings in the browser.

Monitoring heterogeneous environments

Description of diskSpaceWin.pl

This is a Perl script installed in client Windows hosts to find out the disk space characteristics. It executes the “dir” command for the C: drive. The result is parsed and displayed in the browser.

Description of versionWin.pl

This is a Perl script installed in client Windows hosts to find out the version of the operating system. It executes the “ver” command. The result is parsed and displayed in the browser.

Monitoring heterogeneous environments

How to add a client host into the web page

This chapter describes the tasks needed to add a client host into the web page.

Only for Unix clients: create a user id (webcfg) for the automated telnet session

The solution described in this document relies on an automated telnet session into the Unix client host in order to execute some commands to obtain information. For security reasons, we suggest to create a unique user id (such as “webcfg”) which is only used for these purposes.

Important information on the prompt

The Perl module that provides the Telnet functions require that the prompt for this userid **MUST** be the plain default “\$” (dollar sign). If you customize the prompt (the PS1 variable), the telnet functions will fail:

* Correctly configured for Perl Telnet:

If PS1 is defined in the .profile as:

```
PS1='$ '
```

Then the prompt will look like this:

```
$
```

* Incorrectly configured for Perl Telnet:

If PS1 is defined in the .profile as:

```
PS1='$LOGNAME@$THIS_SYSTEM:$PWD'
```

Then the prompt will look like this:

```
rivera@suppserv:/home/rivera
```

Editing the XML file “hostnames.xml”

The XML file “hostnames.xml” has the central table of hosts to be accessed. This is the only file that needs to be updated when adding, deleting or modifying entries for the hosts to be monitored. In this file, the hostnames, login and password id’s, and platform types should be changed to the appropriate ones used in your site.

An example of hostnames.xml is given below:

```
<hostname_start>
  <hostname name="LinuxHost">
    <username>webcfg</username>
    <password>webcfg</password>
    <os>LINUX</os>
  </hostname>
  <hostname name="SolarisHost">
```

Monitoring heterogeneous environments

```
        <username>webcfg</username>
        <password>webcfg</password>
        <os>SOLARIS</os>
</hostname>
<hostname name="AixHost">
    <username>webcfg</username>
    <password>webcfg</password>
    <os>AIX</os>
</hostname>
<hostname name="HpuxHost">
    <username>webcfg</username>
    <password>webcfg</password>
    <os>HP-UX</os>
</hostname>
<hostname name="WindowsHost">
    <username>don't remove</username>
    <password>don't remove</password>
    <os>WINDOWS</os>
</hostname>
</hostname_start>
```

Note:

The XML file “hostnames.html” can be validated by opening it in Internet Explorer 5 or later, which can verify an XML file.

Detailed description of the contents of hostnames.html

The beginning and ending tags “</hostname_start>” and “</hostname_start>” list all the client hostnames to be accessed by the web page.

The subsequent entries have the format:

A) Unix:

An example of the XML specification for a Unix host is shown below:

```
<hostname name="AixHost">
    <username>webcfg</username>
    <password>webcfg</password>
    <os>AIX</os>
</hostname>
```

If the platform for the host is AIX, HP-UX, SOLARIS or LINUX, then the username (login), the password for the username, and the os (operating system or platform type) need to be specified within the “hostname” tags. The name of the host needs to be specified too.

Monitoring heterogeneous environments

In the example shown above, we use the specific userid “webcfg” which is created in the client host just for the purposes of getting access to this client host via Telnet in order to execute commands such as “uname -a” and “df”..

The valid operating system (platform type) values for the “os” sub-tag for Unix platforms are: AIX, HP-UX, SOLARIS or LINUX

B) Windows

An example of the XML specification for a Windows host is shown below:

```
<hostname name="WindowsHost">
    <username>don't remove</username>
    <password>don't remove</password>
    <os>WINDOWS</os>
</hostname>
```

If the platform is Windows, a username (login) and password id are not required; however, for consistency's sake, the string “don't remove” should be present for these fields.

Preparing the whoAmI.html file

It is recommended to create a file named “whoAmI.html” which resides in the document root of the Web Server of the client host. The purpose of this file is to capture information that is rather difficult or cumbersome to obtain automatically via commands that are seldom executed, or characteristics of the hardware that cannot be easily found via commands.

An example of preparing such file is shown below:

- 1) Login as root into the desired client host.
- 2) Change to the document root directory for the Web Server:

Unix:

```
cd /usr/HTTPServer/htdocs/en_US
```

Windows:

```
Cd C:\Program Files\IBM HTTP Server\htdocs
```

- 3) Create a file named “whoAmI.html” and a sample contents is shown below:

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title> Configuration information for the AIX host "AixHost"
</title>
```

Monitoring heterogeneous environments

```
</head>
<body>
<h1> Configuration information for the AIX host "AixHost"
</h1>
<p>
Date last updated (manually): 31-May-2001
</p>
<p>
Location of this file: /usr/HTTPServer/htdocs/en_US/whoAmI.html
</p>
<hr>
<!-- The pre tag is used to delimit text that is not formatted. -->
<pre>
Host name:          AixHost
  IP Address:       1.2.3.4
Primary use:        AIX central server

Operating System:  AIX 4.3.3
Manufacturer:      IBM
Type/Model:        RS/6000, Mode-Type: 7044-270 (44p)
Serial Number:     XX-YYYY
RAM:               2 GB
Disk 1:            18 GB

Java (JDK):
  version:         java -version
                  1.1.8
  home:           /usr/jdk_base

IBM HTTP Server (Apache)
  version:         1.3.12.2
  home:           /usr/HTTPServer
  docroot:        /usr/HTTPServer/htdocs/en_US/

</pre>
<hr>
</body>
</html>
<!-- end of file -->
```

4) Due to the different web servers that we have, we wanted to have an easier way to get to this file. Thus, we also have a symbolic link at the root level:

Unix:

```
$ cd /
$ ln -s /usr/HTTPServer/htdocs/en_US/whoAmI.html /whoAmI.html
$ ls -dl whoAmI.html
whoAmI.html -> /usr/HTTPServer/htdocs/en_US/whoAmI.html
```

Windows:

Create a “shortcut” of the whoAmI.html file to your desktop or another folder.

Monitoring heterogeneous environments

Debugging suggestions

If the web page with the table does not show up in the web browser, then you can try the following suggestions:

Step 1:

Execute the mhe.pl script on the command line:

- Change to the directory where mhe.pl is installed.
- Execute the following command:

```
perl mhe.pl > mhe-cfg.html
```

- Ignore the warnings displayed on the screen.
- Examine the mhe-cfg.html output file and check the corresponding entry in the table for each host.
- If the result of a host is invalid or undefined, test that host by performing step 2 below.

Step 2:

For each host in hostnames.xml, telnet to the host using the specified username/password.

If the telnet section hangs, the password is invalid, or an unexpected error occurs which will cause the mhe.pl script to fail.

Try to fix the problem with that host, and in the meantime remove the host information from the hostnames.xml file.

Once the problem is resolved, you can add back the host into the xml file.

Monitoring heterogeneous environments

How to get the Perl scripts mentioned in this document

All the Perl scripts used here are available via the public Internet. They might be updated in the future. The scripts are zipped into a single file called **trweb-mhe-1.zip** and it can be downloaded as follows:

FTP site for TeamConnection

You can download the code from our external FTP site for TeamConnection, by doing:

1. ftp ftp.software.ibm.com
2. login as **anonymous** and for password give your email address.
3. cd ps/products/teamconnection/papers
4. binary
5. get trweb-mhe-1.zip
6. quit

Obtaining Info-ZIP

The VisualAge TeamConnection team uses the Info-Zip **zip** and **unzip** tools to package compressed files (in which the files to be packaged are compressed first).

The main advantages of Info-ZIP are:

- Compatibility: these tools are compatible with other ZIP programs.
- Portability: they are available in ALL the platforms that are supported by VisualAge TeamConnection.
- Cross-platform: A zip file prepared in Unix can be unzipped in the correct format in Windows NT and vice versa.

Info-ZIP's software is free and can be obtained for the desired platforms from various anonymous ftp sites, including the URL:

<ftp://ftp.uu.net:/pub/archiving/zip/>

Because of the general value of these tools, it is recommended that you add the unzip and zip tools in a directory in the PATH that is accessible to all the users for the machine (such as C:\WINNT in Windows).

How to unzip files

- To only view the contents of the zip file (without actually unpackaging and uncompressing the files) do: **unzip -l trweb-mhe-1.zip**
- To unpackage and uncompress the zip file do: **unzip trweb-mhe-1.zip**

Monitoring heterogeneous environments

Copyrights, Trademarks and Service marks

The following terms used in this technical report, are trademarks or service marks of the indicated companies:

TRADEMARK, REGISTERED TRADEMARK, OR SERVICE MARK	COMPANY
IBM, VisualAge, TeamConnection, DB2 Universal Database, CMVC	IBM Corporation
Apache	Apache Software Foundation
Unix	The Open Group
Windows, Windows NT	Microsoft Corporation
Info-ZIP	Info-ZIP Group
Solaris	Sun Microsystems, Corp.
HP-UX	Hewlett-Packard Corp.
Tivoli	Tivoli, an IBM Company
Linux	Linus Torvalds
Perl	Larry Wall
GNU	Free Software Foundation, Inc.
WinZip	WinZip Computing, Inc.
PKZip	PKWARE, Inc.
Lotus Domino Go	Lotus Corp, a subsidiary of IBM
Netscape Navigator	Netscape Communications Corp.

*** END OF DOCUMENT ***