
System Administrator Guide



MX Application Engine “MXAE”

Document Version	Author	Date written	Description of changes
1.0	Ryan	2004/08/02	Initial version created
1.0.1	Ryan	2004/12/13	Updated Section 3.1.4, 3.1.5, 3.1.6, 3.2, 3.3.2, 3.3.4, 3.4, 3.4.1 to 3.4.7, 3.5.1 to 3.5.16, 4.2, Appendix C, D, F, H, J and K.
1.0.2	Ryan	2005/02/03	Updated Section 2 and 3.
1.0.3	Ryan	2005/07/29	Updated Appendix K, L
1.0.3	Megat	2005/08/5	Updated as at MXAE Version 1.0.11
1.0.4	Ryan/Afu	2006/01/23	Upgraded Document to System Admin Guide. Covers installation up to MXAE version 1.0.14
1.0.5	Ivan/Ryan	2008/02/20	Updated document relevant to MXAE version 1.5.2

Table of Contents

1	Introduction.....	3
1.1	Purpose of Document.....	3
1.2	Scope of Document.....	3
1.3	Document Organization.....	3
1.4	Reference Documents.....	4
2	OS Environment Installation.....	4
2.1	Preparing for OS Installation.....	4
2.2	Installing and Configuring OS.....	4
2.3	Post OS Installation.....	8
2.4	Updating The Operating System after Installation.....	13
3	MXAE Installation.....	14
3.1	Running the MXAE Installer for the First Time.....	14
3.2	Upgrading MXAE.....	18
4	General Operation.....	20
4.1	Increasing swap space after adding more RAM.....	20
4.2	General Server Load Information.....	21
4.3	Backing up and Restoring MXAE.....	22
5	Troubleshooting.....	23
5.1	"Allowed memory size of XXXXX bytes exhausted (tried to allocate XXX bytes)" error.....	23
5.2	"500 Ftp server shutdown" error.....	23
5.3	"FTP login failed" error in AMS File Management.....	23
5.4	"Login to host failed" error in AMS File Management.....	24
5.5	"Cannot create shared memory area" error.....	24
5.6	Apache won't start or generates run time errors.....	24
5.7	MXAE is not working because server/s has changed IP addresses.....	25
6	Conclusion.....	25
	Appendix A – Acronyms and Glossary.....	26
	Appendix B - MXAE Environment Installation (Manual).....	27
	Installing and Configuring MySQL.....	27
	Installing and Configuring Apache.....	29
	Installing and Configuring PHP.....	33
	Installing and Configuring Other Applications.....	51
	Appendix C – sshd_config.....	65
	Appendix D – proftpd.conf.....	66
	Appendix E – xinetd.d (proftpd service).....	66
	Appendix F – logrotate.....	67
	Appendix G – reporting.....	67
	Appendix H – virtual host listing in httpd.conf.....	68
	Appendix I – ftponly.....	68
	Appendix J – crontab.....	69
	Appendix K – Links.....	70

1 Introduction

1.1 Purpose of Document

This document serves as a guideline to installing and general administering the MXAE solution, the next generation Content Management System.

1.2 Scope of Document

The scope includes all necessary installation processes and operational procedures for the MXAE solution to function on a day-to-day basis. This includes installing the operating system, the database server, the web server, and all other supporting third parties applications from ground up. Additionally, the document will include operational tasks that needs to be performed to ensure peak performance of MXAE, as well as troubleshooting procedures in the event MXAE is functioning erratically.

1.3 Document Organization

An MXAE solution installation shall be viewed as follows:

Solution	Layer	Major Items	Other Items
MXAE Solution	OS Environment	Fedora Core, Redhat Enterprise Linux (AS or ES)	Third party programs like backup programs, monitoring programs, etc, depending on compatibility with OS.
	MXAE Environment	Apache, MySQL, PHP	Proftpd, ImageMagick, etc as described below.

As such, this document is divided into three major sections, namely the deployment of the OS environment, the deployment of MXAE environment and the operational procedures of both OS and MXAE environments.

The first section describes how the OS environment shall be installed. An OS operating environment shall consist of the OS kernel, libraries and other packages that are related to the monitoring and maintenance of the operating system.

The next section describes the steps for deploying the MXAE application environment. An MXAE environment consists of the web server (e.g. Apache), database server (e.g. MySQL), application server (php), and its supporting modules. This section also explains other third packages that are crucial to the functional execution of MXAE.

The following sections will describe the general operational and administering tasks a System Administrator has to undertake in order to ensure MXAE performs optimally. These also include

steps to be taken when applying updates, patches and troubleshooting procedures.

1.4 Reference Documents

Adaptation from MXAE Installation Guide 1.1, <http://www.redhat.com>

2 OS Environment Installation

This section covers the OS environment installation, including the OS kernel, library files and its related packages that forms the base operating environment. This part shall include all necessary third party tools and packages for maintenance, monitoring and backup of the system. This installation guide defaults to Fedora Core distribution specifics unless otherwise specified.

2.1 Preparing for OS Installation

For the most recent and detailed list of hardware Red Hat Linux supports, check Red Hat's Hardware Compatibility List at <http://hardware.redhat.com/hcl/>.

The hardware requirements shall vary depending on the deployment layout, however the recommended specification for MXAE installs shall be

- Pentium IV processor or higher
- 1GB of RAM or more
- 10GB of free disk space or more
- CD-ROM drive
- Working network card of 10Mbps
- 800x600 monitor resolution
- Mouse or pointing device

For connection to a network, the following information is required:

- Host name of the node.
- Domain name of the network.
- IP address of the machine.
- Netmask of the network.
- Gateway IP address of the machine.
- IP address of name servers (DNS)

2.2 Installing and Configuring OS

This section summarizes the installation process pertaining to Fedora Cores. A detailed walkthrough on instructions how to install the Linux server can be found at <http://www.redhat.com>.

2.2.1 OS installation.

Insert Operating System installation CD #1 into your computer's CD-ROM drive. Reboot your computer. When the installation screen appears, press Enter to begin the installation.

During installation, questions about your computer hardware and the network connections. After you have completed each answer, click Next. The following list describes the information you will need to enter.

Media Check — Optionally check each CD to be sure it is not damaged or corrupted.

Language — Choose the preferred language.

Keyboard — Choose the keyboard type.

Mouse — Identify the type of mouse being used.

Monitor — Identify the monitor model or at least its horizontal and vertical sync.

Upgrade? — If there is an earlier version of Red Hat Linux installed, you can choose Upgrade to upgrade your system without losing data files. Otherwise, you can continue with a new installation.

Install type — Choose a Custom configuration.

Partitions — Either have Red Hat automatically choose your partitions or manually partition yourself (with Disk Druid). Partitioning using Disk Druid is recommended.

Partitioning of the hard disk is required prior to installation. Following is the basic manual partition plan, and partition sizes and configuration can be done according to one's preferences. If there are existing partitions, it is preferable to remove all existing partitions first.

Partition Name	Size	Description
/boot	100MB	Boot partition that stores the booting files such as kernel, LILO or GRUB. Must reside in the first partition.
/swap	At least twice (2x) the size of installed RAM.	Swap partition that stores the entire temporary file that is where the memory writes to. /swap need to be the double size of the memory. Preferably resides in the next partition to the /boot partition.
/	10GB	Root partition that stores all the Operating System's files and future installation programs.
/usr/local/mxae	Up to the remaining disk space	Partition for storing most of the MXAE environment programs and files.

Disk Druid — Whether choosing Automatic or Manual partitioning, Disk Druid appears onscreen to let you review or change the partitions.

Boot Loader — Choose GRUB boot manager to control the boot process. When configuring authentication, the options “Enable MD5 passwords” and “Enable Shadow Passwords” shall be selected.

Network configuration — Set up the LAN connection by manually entering your computer’s IP address, netmask, host name, default gateway, and DNS servers obtained earlier. Choose “activate” the network when Linux boots.

Firewall — Select “No Firewall” if there is a firewall appliance is in place already. There is also an option to enable Selinux. Please ensure that Selinux is DISABLED to ensure compatibility with Oracle client module.

Language support — Choose to install support for additional languages.

Time zone — Select the right time at which the server is installed.

Root password — Add the root user account password.

Packages — Redhat bundles with many applications to choose from. For uses of the MXAE solution, the OS installation shall be customized to its need. As such when prompted, a “Custom Installation” option shall be selected.

Choose groups of software packages to install. Do not choose Everything, or Minimum. Below are some of the crucial packages that can be included during selection from package group:

Package Group	Category	Packages in detail
Desktops	<i>X Window System</i>	Select all package EXCEPT vnc-server, tkinter, desktop-backgrounds-extra, and authconfig-gtk
	<i>Gnome</i>	Select none.
	<i>KDE</i>	Select none.
Applications	<i>Editors</i>	Select vim-enhanced only
	<i>Graphical Internet</i>	Select mozilla only.
	<i>Text-based Internet</i>	Select elinks only.
	<i>Office/Productivity</i>	Select none.
	<i>Sound and Video</i>	Select none.

	<i>Authoring and Publishing</i>	Select none.
	<i>Graphics</i>	Select all EXCEPT kdegraphics
Servers	<i>Server Configuration Tools</i>	Select none.
	<i>Web Server</i>	Select none.
	<i>Mail Server</i>	Select none.
	<i>Windows File Server</i>	Select none.
	<i>DNS Name Server</i>	Select none.
	<i>FTP Server</i>	Select none.
	<i>SQL Database</i>	Select none.
	<i>MySQL Database</i>	Select none.
	<i>News Server</i>	Select none.
	<i>Network Server</i>	Select none.
	<i>Legacy Network Server</i>	Select none.
Development	<i>Development Tools</i>	Select all package EXCEPT Electric Fence, Ant, Ruby
	<i>Kernel Development</i>	Select all packages.

	<i>X Software Development</i>	Select all packages.
	<i>Gnome Software Development</i>	Select all packages.
	<i>KDE Software Development</i>	Select all packages.
	<i>Legacy Software Development</i>	Select all packages.
System	<i>Admin Tools</i>	Select all redhat-config-xxx packages
	<i>System Tools</i>	Select open only.
	<i>Printing Support</i>	Select none.

Installing packages — To this point, install process can be aborted without having written anything to disk. To proceed, select Next, the disk is formatted (if chosen) and selected packages are installed.

When installation is done, remove the Red Hat Linux CD and click Exit to reboot the computer. Linux should boot by default.

After Linux boots for the first time, the Red Hat Setup Agent runs to set system date and time, sign up for software updates, and install additional CDs. On subsequent reboots, a login prompt will appear instead.

2.3 Post OS Installation.

When the installation is done, you need to make sure that required services are running and non essential services are stopped. If linuxconf is installed, then you can use linuxconf to control the services activity.

A list of menus will appear at the console. Then go to Control → Control Panel → Control Server Activity. Stop and set all services to “off”:

- apmd
- arptables_jf
- canna
- hpoj
- mdmonitor
- netfs
- nfslock
- pcmcia
- portmap

- autofs
- cups
- isdn
- kudzu
- nfslock
- cups-lpd
- finger
- imap
- autofs
- iptables
- rpcsvcgssd
- acpid
- rpcidmapd
- rpcgssd
- lisa
- smartd
- nfs
- lm_sensors
- cups-config-daemon
- keytable

The services below are turned “on” by default. Please ensure it is not turned “off”.

- crond
- gpm
- kudzu
- network
- random
- sshd
- syslog
- xinetd

Then select System services and hit enter. Uncheck those unwanted services and check those needed services.

2.3.1 ***Ensuring important packages are installed.***

To check for installed packages, issue the command:

```
shell> rpm -qa | grep <package-name>
```

Below are the important packages that should be installed:

Packages	Description
up2date	The Red Hat Linux update agent.
vim	X Window-based implementation of the vim editor.

wget	A simple HTTP or FTP file retrieval utility.
service	Command to control the other application's status in Redhat, such as to stop, run and etc.
linuxconf	Redhat's control center. With this application, you can create user, control services, change network setting and etc. Redhat 8.0 no longer provide this application. You need to download separately.
openssh	Application that enable a user to login remotely from other server or remote access to another server. Openssh server and openssh client is required.
logrotate	Rotate growing log files
elinks	A text-based Internet Web browser.
anacron	Periodic command scheduler for machines that do not run 24 hours per day
automake	The automake makefile generator.
bash	Version 2.05b or higher of the GNU Bash shell (Bourne Again SHell)
chkconfig	Changes system service run-level information
crontabs	The root crontabs files, used by the cron daemon to schedule tasks.
gzip	A data compression utility.
kudzu	The kudzu hardware autodetection and configuration utilities.
patch	GNU Patch utility for modifying and upgrading files
perl	Perl high-level programming language and components.

rpm	A utility for installing and managing RPM (.rpm) software packages.
SysVinit	Basic Linux boot and startup programs.
syslogd	The syslogd and klogd daemon processes, for logging system messages.
symlinks	Checks and corrects broken symbolic links.
tar	Archival and restoration utility.
unzip	A utility for testing, listing, and extracting files from .zip archives.
XFree86	The basic software needed to run the X Window System.
xinetd	A secure replacement for the inetd super server.
xinitrc	A utility for configuring X Window System startup.
yum	Tools for downloading and installing packages in RPM format.
zip	A compression and decompression utility.

2.3.2 **Configuring hosts file**

Ensure that the hosts file contains the appropriate system hostname. Assuming the server IP is 10.XXX.XXX.XXX, hostname is johndoe and the domain is acme, do the following:

```
# Edit the hosts file
shell> vi /etc/hosts

# Check that the host file contains the correct hostname :

# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      localhost.localdomain  localhost
10.XXX.XXX.XXX    johndoe.acme.com      johndoe
```

2.3.3 **Configuring hosts.allow and hosts.deny file**

A simple host hardening is performed through the use of TCP wrappers. It is implemented by specifying the hosts that are allowed or denied from connecting to the

server, and its related services. Assuming access is allowed from host 10.0.0.2 network and network 192.168.10.0, do the following:

```
# Edit the hosts.allow file
shell> vi /etc/hosts.allow

#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd: 10.XXX.XXX.XXX, 192.168.10.XXX

# Edit the hosts.deny file
shell> vi /etc/hosts.deny

#
# hosts.deny   This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as
#               decided by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow. In
# particular you should know that NFS uses portmap!
ALL: ALL
```

2.3.4 **Creating and Maintaining System User Accounts**

An account other than root is needed to allow you to work as the Site Admin to manage the system either via the console or remote machine. This is to play safe and avoid from hijacker.

```
# Create a group for the user
shell> groupadd netsite

# Create the folder to keep all users' folder
shell> mkdir /home/users

# Create siteadm user
shell> useradd -d /home/users/siteadm -s /bin/bash -g netsite siteadm

# Set the password for siteadm user
shell> passwd siteadm

# Create a folder to keep all the package files
shell> mkdir /home/users/siteadm/package

# Create a folder to keep all package source files
shell> mkdir /home/users/siteadm/src

# Create a folder to keep all mxae binary files
shell> mkdir /usr/local/mxae
```

2.3.5 **Selecting the right runlevels to boot.**

Upon installation, set the system to boot at the right runlevels depending on the functionality of the server installed. Generally, a server serves as a publisher or database should be set to run at runlevel 3. AMS can be set to run as runlevel 5 so as the gui browser is available to login to the AMS at the server console.

For example, to change the runlevel upon booting, modify the file /etc/inittab as follows:

```
shell> vi /etc/inittab

#
# inittab          This file describes how the INIT process should set
up
#                  the system in a certain run-level.
#
# Author:         Miquel van Smoorenburg,
<miquels@drinkel.nl.mugnet.org>
#                  Modified for RHS Linux by Marc Ewing and Donnie
Barnes
#
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have
networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#

id:3:initdefault:
```

2.4 Updating The Operating System after Installation.

In the world of Linux, patches include the updates that can help you keep your systems secure, error-free, and updated with the latest features.

Generally, the most frequently used distributions are Red Hat Enterprise Linux and Fedora Core, as such they will be further explained.

2.4.1 Updating RedHat Enterprise Linux

The current distribution released by Red Hat is known as Red Hat Enterprise Linux (RHEL). RHEL updates are limited to computers with valid subscriptions on the Red Hat Network (<https://rhn.redhat.com>).

It is best to register first with the Red Hat Network. That will help you manage all your RHEL computers with one Red Hat Network account. After that, purchase one or more subscriptions for RHEL and activate the subscription.

RHEL's Update Agent, also known as up2date, provides a complete system for updating the RPM packages installed on RHEL computers. It allows to register with and monitor appropriate Red Hat repositories for updates.

To automatically updates your system with all upgradeable packages, issue the command:

```
# Updatable packages to the installed system.
shell> up2date -u
```

2.4.2 Updating Fedora Core

To update Fedora Core based installations, simply issue the yum command:

```
# Update Fedora Core systems using yum
shell> yum -y update all
```

3 MXAE Installation

The following sections describe the steps to be taken to perform a full MXAE installation via the automated MXAE installer. To fully understand the tasks that is performed by MXAE Installer, it is recommended for you to browse through Appendix B of this document, which outlines the procedures for manual installation of MXAE.

3.1 Running the MXAE Installer for the First Time

There are a few pre-requisites to running the mxae installer script:

1. Installation is on the superuser 'root' account;
2. The directory /usr/local/mxae does not exist (it is advisable to rename any /usr/local/mxae directories to something different. Else, the script will detect this as an upgrade)

```
[root@sashimi MXAE_Releases]# sh mxae-install-1.0.13.i386.sh
Creating directory mxae
Verifying archive integrity... All good.
Uncompressing
MXAEAE.....
..
If the MXAE script does not run, please contact your reseller for
further support.
Copying files from
/root/MXAE_Releases/mxae/./apache2/htdocs/mxlib/resource/bin/../../../../
../../../../* to /usr/local/mxae...

Checking if there is an existing installation of MX Application
Engine...[FOUND]
The existing MX Application Engine version is 1.0.13
Checking prerequisite before proceeding to installation...
Current system information:
    hostname: sashimi.digialliance.com
    sysname: Linux
    nodename: sashimi.digialliance.com
    release: 2.6.9-11.ELsmp
    version: #1 SMP Fri May 20 18:26:27 EDT 2005
    machine: i686
Linux distribution detected: Red Hat Enterprise Linux ES release 4
(Nahant Update 1)
Check if running as 'root' user... [OK]
Checking PHP version (must be 5.0.2 or later)... [OK]
Checking library path in /etc/ld.so.conf...
Welcome to MX Application Engine Setup!
(Press Ctrl-C to quit at any time)
```

The above dialogue is the first set of screens that will welcome you when the mxae install script is running. At this point, the script is detecting if you have fulfilled the necessary pre-requisites, and additional checks on the OS's environment.

```
Please select setup type:
[1] Application Management System (AMS)
[2] Publisher
[3] Database (MySQL)
[4] AMS, Publisher & Database
[5] AMS & Publisher
[6] AMS & Database
[7] Publisher & Database
```

The second set of screen, as shown above, requires a user input with respect to the type of installation you are currently performing.

MXAE's core components, ie. The AMS, Publisher and Database, can either be installed on one server, or spread across a 3-server setup, or a combination of configurations across 2 servers. The dialogue as shown above gives the possible combinations that you wish to install on the current machine. For example, if the current machine is to be the dedicated database on a 3-server configuration of MXAE, please choose option (3). On a single-server configuration, please choose option (4).

```
Path and filename containing MX Application Engine license []:
```

The next dialogue screen will require you to specify the filesystem location of the file that contains mxae license. For example, if you received your license via email, copy the contents of the license to /root/license.txt and specify /root/license.txt in the dialogue screen above.

```
A user 'siteadm.netsite' will be created later. Please assign a password for this user []: asdfjkl
```

The next dialogue screen will inform you a system user 'siteadm' of group 'netsite' will be created, and a password is required. The system user 'siteadm' is the primary owner of various mxae-related files under the directory /usr/local/mxae/apache2/htdocs. As such, it is recommended that this password to be kept confidential.

```
Does your server has a fully qualified domain name (FQDN)?
[hostname.domain.com]: 192.168.1.10
Please specify the email address for Apache administration
notification? [admin@192.168.1.10]:
Do you want to run your AMS in HTTP and/or HTTPS mode
(HTTP/HTTPS/BOTH)? [HTTP]:
What IP do you want the web server to bind to? [192.168.1.10]:
Which port do you want the server to listen to (HTTP)? [80]:

What timezone is this server in? [Asia/Kuala_Lumpur]:

Please specify a default language? [en_us.iso-8859-1]:
```

The next set of dialogue screens will require you to input a few information with regards to the networking information. These information are critical to the correct operation of *Apache Web Server*, and as it is important the correct information are given.

```
What is the Database type? [mysql]:
What is the Database host IP? [192.168.1.10]:
What is the Database host port number? [3306]:
What is the name of the db MX Application Engine is on? [mxsystem]:
What is the username (for AMS) to access the database? [mxae]:
What is the corresponding password to access the database (mxae)? []:
asdfjkl
Assign a MySQL root password []: asdfjkl
```

The next dialogue screens will require you to input the information for the correct operation of the MySQL database server. Again, ensure only accurate information are given. This dialogue screen will only be applicable if the current machine is specified to host a Database in the Setup screen.

```
What is the AMS host IP? [192.168.1.10]:  
What is the URL to access the AMS (eg.  
"http://ams.somedomain.com/index.php")?  
[http://192.168.1.10/index.php]:  
What is the agent host IP? [192.168.1.10]:  
Specify a port for the agent to listen to [7812]:  
  
There should be a MX Application Engine webserver record created for  
this server. What's the ID of this webserver [1]:
```

If the current machine was specified to host AMS (Application Management System) portion of MXAE, the above dialogue screen will be presented. Ensure correct information are given, especially if mxae are to be installed on a multi-server configuration.

```
This part allow you to restrict access to this server by IPs:  
Deny from host [all]:  
Allow from host [all]:
```

The subsequent dialogue screen is the simple access restrictions, if any, to be applied the server. It closely resembles Apache Web Server access restrictions. All hosts/IP are explicitly on Deny list (no access), and explicitly specify the IPs that have access to the server (unless it is 'all'). Syntax of the list of IPs are 192.168..x or 192.168.0.* (multiple entries are separated by commas).

```
Please specify a default backup path [/home/users/siteadm/backup]:
```

The final screen that requires user input is to specify the default location for datafiles backups that is regularly done in-system. These includes Apache, MySQL and AWStats datafiles.

```
.....
Preparing mysql directories' permissions...
Installing mysqld service...
Starting mysqld...
Setting root password for mysqld...
Connecting to mysql DB server (127.0.0.1)...
[OK]
Checking if DB 'mxsystem' exists...[FAILED]
Creating DB 'mxsystem'...[OK]
Populating DB 'mxsystem'...[OK]
Granting access to mxae@192.168.14.87...[OK]
Granting access to root@sashimi.digialliance.com...[OK]
Granting access to root@192.168.14.87...[OK]
Creating a new dbsever record...[OK]
Creating a new websever record...[OK]
Setting up PHP...
There is already an existing php.ini at /usr/local/mxae/lib/php.ini.
Overwrite? (Y/N) [Y]:
Setting up Apache...
Creating necessary web directories...
Updating httpd.conf...
Installing apachectl service...
Starting Apache HTTP Server...
Updating config.ini...
Updating phpMyAdmin config.inc.php...
Setting up ProFTPD...
Updating proftpd.conf...
Starting ProFTPD...
Setting up mxAgent...
Starting mxAgent...
Setting up cron jobs...
Updating MANPATH...

Setup completed!

[root@sashimi MXAE_Releases]#
```

Assuming all information was given correctly, mxae installer will proceed with a long list of diagnostic messages before informing that the setup process has been completed.

Congratulations! You have successfully installed MXAE.

3.2 Upgrading MXAE

The MXAE installer script is a dual-purpose script, it is also used to perform upgrades to MXAE for newer versions. Basically it does this by detecting a current installation on the server, and will ask you to confirm whether an upgrade or fresh installation is to be done. On upgrading, the script will retain all current MXAE settings and datafiles, and binary files will be replaced with the new version. You will have an option to make a backup of the previous version of MXAE.

The following section will attempt to briefly illustrate the procedures in upgrading MXAE via mxae installer.

```
[root@sashimi MXAE_Releases]# sh mxae-install-1.5.2.i386.sh
Creating directory mxae
Verifying archive integrity... All good.
Uncompressing MXAEAE.....

.....
MXAE will be installed to directory /usr/local/mxae...
If the MXAE script does not run, please contact your reseller for
further support.
Checking if there is an existing installation of MX Application
Engine...[FOUND]
The existing MX Application Engine version is 1.0.13
Do you want to UPGRADE or REINSTALL MX Application Engine? (U/R) [U]:
```

MXAE installer has detected that you currently have a working copy of MXAE installed on the machine. At this point you will be asked to confirm whether you wish to Upgrade or Re-Install.

```
Do you want to upgrade/replace the binary files as well MX Application
Engine? (Y/N) [Y]:

Do you want to backup your existing MX Application Engine
installation? (Y/N) [Y]:

Where do you want to backup the existing installation to?
[/tmp/mxaebackup-20060126122822.tgz]:
Backing up your existing installation to /tmp/mxaebackup-
20060126122822.tgz ...
cd /usr/local/mxae/./; /bin/tar -zcvf /tmp/mxaebackup-
20060126122822.tgz mxae
```

The next set of questions will be confirming the actions required prior to the Upgrading processes. The Defaults are highly recommended in all cases ie

1. Binary Files are replaced/upgraded
2. Existing MXAE installation (including datafiles) are backed up to a suitable location

Once confirming the above, the installer will initiate the upgrade processes, and outputting a lot of information to your screen. It is safe to ignore these information.

```
.....  
Checking if there is an existing installation of MX Application  
Engine...[FOUND]  
The existing MX Application Engine version is 1.5.2  
Restarting all related services...  
Restarting mysql service...  
Killing mysqld with pid 2839  
Wait for mysqld to exit. done  
Restarting apache service...  
Restarting proftpd service...  
Restarting mxAgent service...  
Do you want to remove the extracted files at directory  
/root/MXAE_Releases/mxae ? (Y/N) [Y]:  
Removing entire directory...  
  
[root@sashimi MXAE_Releases]#
```

Congratulations! Your MXAE has been upgraded.

4 General Operation

4.1 Increasing swap space after adding more RAM.

The swap partition should be at least twice or more than the installed memory. However, when more memory is added to the system later, instead of increasing the swap partition, the swap space can be added as swapfiles and adding them to the swap pool.

For example, to add 1 GB of swap space, assuming putting the swapfile in /var directory:

```
# To make 1G swap file (do not make 2G in single swap file due to file system  
limitation)  
shell > dd if=/dev/zero of=/var/swapfile1 bs=1024 count=1048576  
shell > mkswap /var/swapfile1 1048576  
shell > sync  
  
# Add the swapfile to the existing swap space  
shell > swapon /var/swapfile1
```

To add more swap space, simply repeat the four commands above, substituting *swapfile1* with *swapfile2* and so on.

To persist against reboots, add the last line to /etc/rc.local:

```
# To make 1G swap file (do not make 2G in single swap file due to file system  
limitation)  
shell > vi /etc/rc.local  
  
# Enable 2GB additional swap space during boot  
swapon /var/swapfile1  
swapon /var/swapfile2
```

4.2 General Server Load Information

To check for currently running server load, issue the command “top” as follows. Check for the “load average” section.

```

shell > top
11:24:29 up 20:35, 1 user, load average: 0.00, 0.00, 0.00
54 processes: 52 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  cpu   user   nice  system  irq  softirq  iowait  idle
              total 0.0%  0.0%  0.0%  0.0%  0.0%  0.0%  100.0%
Mem:   513512k av, 506788k used, 6724k free, 0k shrd, 74456k buff
              318016k actv, 0k in_d, 10772k in_c
Swap: 1052248k av, 0k used, 1052248k free 342004k cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT  %CPU %MEM    TIME CPU  COMMAND
 2993 nobody    15   0 20676  20M  7920 S    0.0  4.0   0:01  0  httpd
 2544 gdm        15   0 17568  17M  7284 S    0.0  3.4   0:00  0  gdmgreeter
 3199 nobody    15   0 15488  15M  7420 S    0.0  3.0   0:00  0  httpd
 2994 nobody    25   0 15444  15M  7376 S    0.0  3.0   0:00  0  httpd
 2995 nobody    25   0 15444  15M  7376 S    0.0  3.0   0:00  0  httpd
 2996 nobody    25   0 15444  15M  7376 S    0.0  3.0   0:00  0  httpd
 2997 nobody    25   0 15444  15M  7376 S    0.0  3.0   0:00  0  httpd
 2367 root       15   0 15412  15M  7356 S    0.0  3.0   0:04  0  httpd
 2455 mysql     24   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2468 mysql     15   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2469 mysql     20   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2470 mysql     24   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2471 mysql     24   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2472 mysql     20   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2478 mysql     15   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2479 mysql     15   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2480 mysql     24   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2481 mysql     20   0 12588  12M  1412 S    0.0  2.4   0:00  0  mysqld
 2536 root       15   0 57928 8284 2300 S    0.0  1.6   0:00  0  X
 2535 root       21   0 3400  3400 2704 S    0.0  0.6   0:00  0  gdm-binary
 2457 xfs        15   0 3320  3320 868 S    0.0  0.6   0:00  0  xfs
 2494 root     22   0 3108  3108 2480 S    0.0  0.6   0:00  0  gdm-binary
 2348 root     15   0 2600  2600 1920 S    0.0  0.5   0:00  0  sendmail
 2357 smmsp    15   0 2272  2264 1720 S    0.0  0.4   0:00  0  sendmail
 3200 siteadm  15   0 2260  2260 1844 R    0.0  0.4   0:00  0  sshd
 3197 root     16   0 2004  2004 1656 S    0.0  0.3   0:00  0  sshd
 2314 root     15   0 1476  1476 1240 S    0.0  0.2   0:00  0  sshd
 3201 siteadm  15   0 1368  1368 1140 S    0.0  0.2   0:00  0  bash

```

Explanation of the statistics is as follows:

Option	Explanation
Us	CPU time spent in user applications.
Sy	CPU time spent in the kernel.
Ni	CPU time spent in "nice"ed processes.
Id	CPU time spent idle.
Wa	CPU time spent waiting for I/O.
Hi	CPU time spent in the irq handlers.
Si	CPU time spent in the softirq handlers.
Load average	The 1-minute, 5-minute, and 15-minute load average.

%CPU	The percentage of CPU that a particular process is consuming.
PRI	The priority value of the process, where a higher value indicates a higher priority. RT indicates that the task has real-time priority, a priority higher than the standard range.
NI	The nice value of the process. The higher the nice value, the less the system has to execute the process. Processes with high nice values tend to have very low priorities.
WCHAN	If a process is waiting on an I/O, this shows which kernel function it is waiting in.
STAT	This is the current status of a process, where the process is either sleeping (S), running (R), zombied (killed but not yet dead) (Z), in an uninterruptable sleep (D), or being traced (T).
TIME	The total amount CPU time (user and system) that this process has used since it started executing.
COMMAND	That command that this process is executing.
LC	The number of the last CPU that this process was executing on.
FLAGS	This toggles whether the load average and uptime information will be updated and displayed.

4.3 Backing up and Restoring MXAE

Generally, all backups are done and taken care of by MXAE during installation (mxae-installer). These tasks are run as crontab entries ("crontab -l" to list or "crontab -e" to make edits). It should not be necessary to do additionally backup tasks apart from the following cases:

1. ad-hoc backups not on the times specified in crontab
2. backing up to external media or to integrate to other backup software.

In most cases, it is necessary to backup only the datafiles of MXAE as the binary files can easily be re-installed from mxae-installer. The datafiles are as follows:

1. Web server and PHP files : /usr/local/mxae/apache2/htdocs/*
2. MySQL data: /usr/local/mxae/mysql/data/*
3. AWstats statistics: /usr/local/mxae/awstats/*

In the event that a complete re-installation of MXAE needs to be done, please ensure that you have the latest copies of the abovementioned files. Do a complete installation of MXAE, and restore the above files into the default locations, overwriting all files. Your MXAE will then be fully restored.

Note: The above assumes a complete restore on a machine that was on the same IP or network location. If you require a change of IP address, please see our FAQ section, as well as contact your reseller for a new working copy of MXAE license.

5 Troubleshooting

5.1 “Allowed memory size of XXXXX bytes exhausted (tried to allocate XXX bytes)” error.

5.1.1 *Problem description*

In apache error log located at /usr/local/mxae/apache2/logs/error.log, an error like “Allowed memory size of XXXX bytes exhausted” appears.

PHP allows each process (i.e. web page) to consume a certain amount of memory (often this is 8M by default). Templates develop on top of MXAE sometimes needs to load a lot of files (e.g. language files, library files) to create one web page, and may start to break the memory limit for some pages.

5.1.2 *Solution*

If you have access to your main php.ini file (located at /usr/local/mxae/lib/php.ini), then find the line that looks like this:

```
memory_limit = OLDVALUE;
```

and change it to:

```
memory_limit = NEWVALUE;
```

then restart Apache with the command “service apacheclt restart”.

5.2 "500 Ftp server shutdown" error.

5.2.1 *Problem description*

The error "500 Ftp server shutdown" appears immediately when starting the proftpd server manually.

5.2.2 *Solution*

Remove the file /etc/shutmsg by issuing “rm -fr /etc/shutmsg” or resume proftpd service with command “/usr/local/mxae/sbin/proftpd resume”.

5.3 “FTP login failed” error in AMS File Management.

5.3.1 *Problem description*

When clicked on “File Management” under AMS, a dialog box with message “FTP login failed” appears.

This could due to the missing file “/bin/ftponly” or the line “/bin/ftponly” is missing from the /etc/shells file.

5.3.2 *Solution*

Check if the ftponly file is in the system by issuing:

```
"ls /etc/ftponly"
```

and

```
`echo "/bin/ftponly" >> /etc/shells"
```

to add the corresponding line into the /etc/shells file.

Also restart the proftpd daemon if it is not running.

```
# Check if proftpd is running
shell > ps -ef | grep proftpd

# Restart proftpd daemon
shell > /usr/local/mxae/sbin/proftpd -c /usr/local/mxae/etc/proftpd.conf
```

5.4 "Login to host failed" error in AMS File Management.

5.4.1 Problem description

When clicking on "File Management" under AMS, a dialog box with message "Login to host failed" appears.

This could be due to the MXAE agentd not running or ftpuser password is incorrect.

5.4.2 Solution

Restart the agentd by issuing the following command or try to login using an ftp client to the user using the correct login and password sets.

```
shell > /usr/local/mxae/bin/php -q
/usr/local/mxae/apache2/htdocs/eacdb/mxlib/agent/agentd.php -c
/usr/local/mxae/apache2/htdocs/eacdb/mxapp/config.ini
```

5.5 "Cannot create shared memory area" error.

5.5.1 Problem description

In apache error_log file, the "cannot create shared memory area" error appears. This is due to default shared memory size for the 2.6 kernel is 32mb only.

5.5.2 Solution

This size can be changed by adding the line kernel.shmmax=VALUE to /etc/sysctl.conf and modifying the option eaccelerator.shm_size="VALUE" in usr/local/mxae/lib/php.ini.

5.6 Apache won't start or generates run time errors.

5.6.1 Problem description

If apache is not starting correctly, there may be mistakes in the httpd.conf file that prevent the Apache process from starting properly.

5.6.2 Solution

Test the Apache configuration file with the -t flag to httpd:

```
# /usr/local/mxae/apache2/bin/httpd -t
```

Check error messages with the command:

```
# "tail -f /var/log/messages" and
```

```
# tail -f /usr/local/mxae/apache2/logs/error_log
```

5.7 MXAE is not working because server/s has changed IP addresses

5.7.1 Problem Description

IF for some reason, the machines that is running the AMS or websites for MXAE changed its IP addresses, AMS and websites cant be accessed.

5.7.2 Solution

A request should be made to your reseller for a new license, either for AMS or websites license. MXAE licenses are tied to network IP addresses, hence this requirement.

Additionally, manual edits to some files are required to reflect these network changes. The edits are merely replacing the new IP addresses, and also to enter the new license strings.

These files are as follows:

1. */usr/local/mxae/apache2/htdocs/mxapp/config.ini* (AMS license and IP addresses)
2. */usr/local/mxae/apache2/htdocs/<website>/config.ini* (Website license and IP addresses)
3. */usr/local/mxae/apache2/conf/httpd.conf* (IP addresses)
4. */usr/local/mxae/etc/proftpd.conf* (IP address in the access restrictions. Note: only localhost IPs are allowed to use the FTP services)

6 Conclusion

This document has covered most of the procedures required to install the MXAE solution and the general day-to-day operations. Please contact your reseller in the event you encounter situations or problems which are not covered in this document.

Appendix A – Acronyms and Glossary

MXAE, or MX Application Engine

This is the core of the MXAE solution whereby most of the application logic and business workflow resides. In short, this is the engine to the whole solution.

AMS, or Application Management System

The front-end application that connects to the MX Application Engine to setup, configure and manage the engine configurations and parameters.

MXAE API, or Application Programming Interface

An external connector, or exposed interface, whereby developer can have their programs interacts with MXAE. Commands can be sent and retrieved from MX Application engine through this connector.

Appendix B - MXAE Environment Installation (Manual)

This section discusses the manual method of mxae installation. This section is included here for reference only. There should not be any need to install MXAE manually as mxae-installer script is robust enough to cover any requirements given the prerequisites are met.

Installing and Configuring MySQL

In this chapter, installation of MySQL is explained and demonstrated. Following are the steps you can find.

- Installation
- MySQL Auto Start and Auto Shut
- MySQL Manual Start and Manual Shutdown
- Setting password for MySQL Admin User

First create a user and group in the linux system prior to installing MySQL.

```
# Create a group for the mysql user
shell> groupadd mysql

# Create mysql user
shell> useradd -d /home/users/mysql -s /bin/false -g mysql mysql
```

Get the binary package for installation. This is the recommended because the latest MySQL binary source is tested and compile for Pentium machine. Download the source file to /home/users/siteadm/packages.

```
shell> tar -zxvf /home/users/siteadm/packages/mysql-<VERSION>-<OS>.tar.gz -C
/usr/local/mxae

shell> ln -s /usr/local/mxae/mysql-<VERSION>-<OS> /usr/local/mxae/mysql

shell> cd /usr/local/mxae/mysql
shell> ./scripts/mysql_install_db

shell> chown -R root /usr/local/mxae/mysql/
shell> chown -R mysql /usr/local/mxae/mysql/data
shell> chgrp -R mysql /usr/local/mxae/mysql
```

You need to choose a database size for MySQL database. There are several database available, choose according to your system needs. The different size configuration is located as the folder below, browse to the folder to choose the configuration file. Then, choose `my-huge.cnf` that suits your machine role, and rename as below.

```
shell> cp /usr/local/mxae/mysql/support-files/my-huge.cnf
/usr/local/mxae/mysql/data/my.cnf
```

After doing so, edit the `mysql.server` file:

```
shell> vi /usr/local/mxae/mysql/support-files/mysql.server
```

Change value of basedir and datadir to:

```
basedir=/usr/local/mxae/mysql  
datadir=/usr/local/mxae/mysql/data
```

Once that's done, continue with the following:

```
shell> cp /usr/local/mxae/mysql/support-files/mysql.server /etc/rc.d/init.d/mysqld  
shell> chmod 700 /etc/rc.d/init.d/mysqld
```

Some additional options can be set in my.cnf to increase the MySQL performance are,

```
set-variable      = thread_concurrency=16  
set-variable      = myisam_sort_buffer_size=100M  
set-variable      = max_connections=400  
set-variable      = max_allowed_packet = 8M  
log-slow-queries
```

MySQL Auto Start and Auto Shut

You can make MySQL to have auto startup and auto shutdown when ever the server reboot or startup. It is recommend to have this service enable for future easy maintainance.

To allow MySQL auto start and auto shut service, issue the following commands,

```
shell> chkconfig --add mysqld  
shell> service mysqld on
```

MySQL Manual Start and Shutdown

With this, you can start, stop or restart the system anytime, anywhere using the following command,

```
shell> service mysqld <stop|start|restart>
```

Setting password for MySQL Admin User Account

After installing the mysql server, start the database and assign administrator user's password for MySQL database account. Administrator user owns the highest privileges of all. You need to be careful and make sure that the person with rights assigned can access this account information.

When starting mysql for the first time, issue the following command:

```
shell> /usr/local/mxae/mysql/bin/mysqladmin -u root password '<new-password>'
```

Otherwise, anybody would be able to access the database as root by just supplying an empty password.

Installing and Configuring Apache

In this chapter, installation of Apache is explained and demonstrated. Download the source file to /home/users/siteadm/packages.

Don't install rpm packages, install the source package.

Apache Pre-Installation

Untar the package:

```
shell> tar -zxvf /home/users/siteadm/packages/httpd-<VERSION>.tar.gz -C  
/home/users/siteadm/src
```

Apache Main Installation

We're ready to build Apache web server.

```
shell> cd /home/users/siteadm/src/apache-VERSION  
  
shell> ./configure --prefix=/usr/local/mxae/apache2 --enable-so --enable-rewrite=shared  
--enable-ssl=shared --enable-deflate=shared --with-ssl=/usr/local/mxae  
  
shell> make  
  
shell> make install
```

If just want to update or upgrade the binary files of apache, replace the 'make install' with the following command:

```
shell> make install-programs
```

Once you have installed apache successfully, you need to modify its' configuration file using 'vi' or your favorite editor.

```
shell> vi /usr/local/mxae/apache2/conf/httpd.conf
```

Following is the changes you need to make in order to run MXAE.

Now you have to edit httpd.conf to ensure PHP extensions are supported.

Look the following lines in the default httpd.conf:

```
#AddType application/x-httpd-php3 .php3
#AddType application/x-httpd-php3-source .phps
```

And change them to look like the below lines:

```
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

Check if the following is set, it should be set automatically

```
LoadModule ssl_module modules/mod_ssl.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule php5_module modules/libphp5.so
```

Include index.php into the DirectiveIndex directive,

```
#
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index. Separate multiple entries with spaces.
#
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html
</IfModule>
```

Make sure directory listing is not turn on in the absence of valid directory index file. Further customizations include:

```
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/usr/local/apache/htdocs/">

#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews". Change symlinks to
# disable directory browsing
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
#Options Indexes FollowSymLinks MultiViews ExecCGI
#Remove Indexes from the Options directive
Options FollowSymLinks MultiViews ExecCGI
#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options",
# "FileInfo", "AuthConfig", and "Limit"
#
    AllowOverride ALL

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all

# set the timezone of the server accordingly (mandatory)
    SetEnv TZ Asia/Kuala_Lumpur
```

```
# set the oracle home directory environment if require oracle support (optional)
  SetEnv ORACLE_HOME /opt/oracle
</Directory>
```

Change directive User and Group in which Apache should run under to httpd and netsite respectively.

```
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# .On SCO (ODT 3) use "User nouser" and "Group nogroup".
# .On HPUNIX you may not be able to use shared memory as nobody, and the
#   suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group nobody on these systems!
#
User nobody
Group #-1
```

Set the ServerSignature to off and ServerTokens, This is for the security reason. ServerTokens is not there by default, you need to add in manually. You can do the same as below.

```
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#
ServerSignature Off
ServerTokens Prod
```

Set your ServerName directive to your host IP and ServerAdmin directive to your webmaster email address. We will take the example that your server IP is <Your-Server-IP>

```
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents.
#
ServerAdmin webmaster@mywebsite.com

#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name YOU
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://10.XXX.XXX.XXX/)
# anyway, and this will make redirections work in a sensible way.
#
ServerName <Your-Server-IP>
```

Add this section to disable browsing for any files with extension .ini

```

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
<Files ~ ".*\.ini">
    Order allow,deny
    Deny from all
</Files>
<Files ~ ".*\.log">
    Order allow,deny
    Deny from all
</Files>
<Files ~ ".*\.conf">
    Order allow,deny
    Deny from all
</Files>

```

Prepare Apache to support virtual hosting by declaring NameVirtualHost directive and also create a default VirtualHost entry. Assuming server IP is 10.0.0.1, edit the following:

```

NameVirtualHost 10.0.0.1:80

# this entry correspond to where MXAE AMS is installed
<VirtualHost 10.0.0.1:80>
    DocumentRoot /usr/local/mxae/apache2/htdocs/mxapp
    ServerName 10.0.0.1
    <Directory "/usr/local/mxae/apache2/htdocs/mxapp/phpMyAdmin/">
        Options None
        AllowOverride None
        Order allow,deny
        # replace with any restricted IPs that u allow access to phpMyAdmin
        Allow from 10.0.0.1, 192.168.88.
    </Directory>
    ErrorLog logs/ams-error_log
    CustomLog logs/ams-access_log combined
    #
    # Directives to allow use of AWStats as a CGI
    #
    Alias /awstatsclasses/ "/usr/local/mxae/awstats/wwwroot/classes/"
    Alias /awstatscss/ "/usr/local/mxae/awstats/wwwroot/css/"
    Alias /awstatsicons/ "/usr/local/mxae/awstats/wwwroot/icon/"
    ScriptAlias /awstats/ "/usr/local/mxae/awstats/wwwroot/cgi-bin/"
    #
    # This is to permit URL access to scripts/files in AWStats directory
    #
    <Directory "/usr/local/mxae/awstats/wwwroot/cgi-bin">
        Options None
        AllowOverride None
        Order allow,deny
        # replace with AMS IP
        Allow from 10.0.0.1
    </Directory>
</VirtualHost>

```

Apache Manual Start and Shut

In order for the modified httpd.conf to take effect, you need to restart the Apache daemon server.

```

shell> /usr/local/mxae/apache2/bin/apachectl start

shell> /usr/local/mxae/apache2/bin/apachectl stop

```

Apache Auto Start and Auto Shut

In order to allow for httpd to be able to start automatically when the system restarts, copy the apachectl file to init.d and append the following lines into /etc/rc.d/init.d/httpd.

```
# Copy startup script to system startup scripts folder
shell> cp /usr/local/mxae/apache2/bin/apachectl /etc/rc.d/init.d/httpd

# Edit the file and add header to enable support by service command
shell> vi /etc/rc.d/init.d/httpd

Line 1 : #!/bin/sh
Line 2 : #
Line 3 : # Startup script for the Apache Web Server
Line 4 : #
Line 5 : # chkconfig: 345 85 15
Line 6 : # description: Apache is a World Wide Web server.  It is used to serve \
Line 7 : #           HTML files and CGI.
Line 8 : # processname: httpd
Line 9 : #
Line 10: # Copyright 2000-2004 The Apache Software Foundation
Line 11: #
Line 12: # Licensed under the Apache License, Version 2.0 (the "License");
Line 13: # you may not use this file except in compliance with the License.
```

Then, add httpd as a service with the following command

```
shell> chkconfig --add apachectl

# Apache can be manually started or stopped with this alternate command after adding
# as service
shell> service apachectl <start|stop|restart>
```

Installing and Configuring PHP

In this chapter, installation of php is explained and demonstrated.

Following are the topic you can find.

- PHP preinstallation.
- PHP Installation
- Editing PHP.INI
- PHP extension modules

PHP Preinstallation

In this section, installation of php dependant packages is explained. These packages are required prior to php compilation.

Following are the topic you can find.

- LDAP
- ORACLE INSTANCE CLIENT
- KERBEROS
- LIBPNG
- LIBTIFF

- LIBJPEG
- LIBXML2
- FREETYPE
- OPENSSSL
- IMAP
- MHASH
- LIBMCRYPT
- FREETDS
- CURL

LDAP

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get first Berkeley DB (http://www.sleepycat.com/products/db.shtml)
shell> tar -zxvf ../packages/db-4.3.27.tar.gz
shell> cd db-4.3.27
shell> cd build_unix
shell> ./dist/configure " '--prefix=/usr/local/mxae'
shell> make && make install

# get openldap from (http://www.openldap.org/)
# download openldap-VERSION.tgz
shell> tar -zxvf ../packages/openldap-VERSION.tgz
shell> cd openldap-VERSION
shell> ./configure --prefix=/usr/local/mxae
shell> make && make install
```

ORACLE INSTANCE CLIENT

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from
(http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html)
# download instantclient-sdk-linux32-10.1.0.3.zip & instantclient-basic-
linux32-10.1.0.3.zip
# unzip both files to /usr/local/mxae

shell> mv /usr/local/mxae/instantclient10_1 /usr/local/mxae/oci8-
instantclient

shell> cd /usr/local/mxae/oci8-instantclient

shell> ln -s libclntsh.so.10.1 libclntsh.so
```

KERBEROS

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://web.mit.edu/kerberos/dist/index.html)

shell> tar -zxvf /home/users/siteadm/packages/krb5-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/krb5-<VERSION>/src

shell> ./configure --prefix=/usr/local/mxae --enable-shared --without-tcl

shell> make && make install
```

LIBPNG

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://www.libpng.org/pub/png/libpng.html)

shell> tar -zxvf /home/users/siteadm/packages/libpng-<VERSION>-
config.tar.gz -C /home/users/siteadm/src

shell> cd /home/users/siteadm/src/libpng-<VERSION>-config

shell> ./configure --prefix=/usr/local/mxae

shell> make && make install
```

LIBTIFF

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://www.libtiff.org/)
shell> tar -zxvf /home/users/siteadm/packages/tiff-<VERSION>.tar.gz -C
/home/users/siteadm/src
shell> cd /home/users/siteadm/src/tiff-<VERSION>
shell> ./configure --prefix=/usr/local/mxae
shell> make && make install
```

LIBJPEG

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://www.ijg.org/)
shell> tar -zxvf /home/users/siteadm/packages/jpegsrc.<VERSION>.tar.gz -C
/home/users/siteadm/src
shell> cd /home/users/siteadm/src/jpeg-<VERSION>
shell> ./configure --prefix=/usr/local/mxae --enable-shared --enable-static
shell> make && make install
```

LIBXML2

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest rpm from (http://xmlsoft.org/sources/)
shell> tar -zxvf ../packages/libxml2-<VERSION>.tar.gz
shell> cd libxml2-<VERSION>
shell> ./configure --prefix=/usr/local/mxae
shell> make && make install
```

FREETYPE

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/freetype-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> ./configure --prefix=/usr/local/mxae

shell> make

shell> make install
```

OPENSSL

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/openssl-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> ./config --prefix=/usr/local/mxae

shell> make

shell> make install
```

IMAP

Download the source file c-client.tar.Z to /home/users/siteadm/packages/ from <ftp://ftp.cac.washington.edu/imap/>. This is the library file required by IMAP.

Installation:

```
shell> cd /home/users/siteadm/src

shell> tar -zxvf ../packages/c-client.tar.Z

shell> cd imap-<VERSION>

shell> make slx SSLDIR=/usr/local/mxae/share/ssl
SSLINCLUDE=/usr/local/mxae/include SSLLIB=/usr/local/mxae/lib

shell> cd ..

shell> cd /usr/local/mxae/imap-<VERSION>

shell> ln -sf c-client include

shell> ln -sf c-client lib
```

MHASH

Download the source file to `/home/users/siteadm/packages/`.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/mhash-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/mhash-<VERSION>

shell> ./configure --prefix=/usr/local/mxae --disable-pthreads

shell> make

shell> make install
```

LIBMCRYPT

Download the source file to `/home/users/siteadm/packages/`.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/libmcrypt-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/libmcrypt-<VERSION>

shell> ./configure --prefix=/usr/local/mxae --disable-posix-threads --enable-
ltdl-install

shell> make

shell> make install
```

FREETDS

FreeTDS enables MSSQL support. Download the source file to `/home/users/siteadm/packages/`.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/freetds-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/freetds-<VERSION>

shell> ./configure --prefix=/usr/local/mxae --with-tdsver=7.0 --enable-
msdblib --enable-dbfix --with-gnu-ld --enable-shared --enable-static

shell> make

shell> make install
```

CURL

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/curl-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/curl-<VERSION>

shell> ./configure --prefix=/usr/local/mxae --with-ssl=/usr/local/mxae --
with-zlib

shell> make

shell> make install
```

PHP Installation

Download the source file to /home/users/siteadm/packages.

```
shell> tar -zxvf /home/users/siteadm/packages/php-<VERSION>.tar.gz -C
/home/users/siteadm/src

shell> cd /home/users/siteadm/src/php-VERSION

shell> ./configure --prefix=/usr/local/mxae --enable-memory-limit --enable-sockets --
enable-pcntl --enable-sigchild --enable-track-vars --with-mime-magic --with-
zlib=/usr/local/mxae --with-bz2 --with-zip --with-mysql=/usr/local/mxae/mysql
--with-mysqli=/usr/local/mxae/mysql/bin/mysql_config --disable-debug --
enable-ftp --enable-bcmath --enable-inline-optimization --with-xml --with-
xmlrpc --with-mhash=/usr/local/mxae --with-mcrypt=/usr/local/mxae --with-
imap=/usr/local/mxae/imap --with-imap-ssl=/usr/local/mxae --enable-trans-sid
--with-gd --with-jpeg-dir=/usr/local/mxae --with-png-dir=/usr/local/mxae --
with-freetype-dir=/usr/local/mxae --enable-soap --with-curl=/usr/local/mxae -
-with-openssl=/usr/local/mxae --with-mssql=/usr/local/mxae --with-config-
file-path=/usr/local/mxae/lib --with-ldap=/usr/local/mxae --with-libxml-
dir=/usr/local/mxae --enable-mbstring=all --enable-mbregex --with-
oci8=instantclient,/usr/local/mxae/oci8-instantclient --with-
apxs2=/usr/local/mxae/apache2/bin/apxs

shell> make

shell> make install

shell> cp /home/users/siteadm/src/php-VERSION/php.ini-dist /usr/local/lib/php.ini
```

Editing PHP.INI

PHP INI is the setting file for the php. You need to change some of the setting as below.

Use 'vi' to open the php.ini,

```
shell> vi /usr/local/mxae/lib/php.ini
```

Change the register_global's values.

```
; You should do your best to write your scripts so that they do not require  
; register_globals to be on; Using form variables as globals can easily lead  
; to possible security problems, if the code is not very well thought of.  
register_globals = On
```

Make sure the following items are set.

```
register_long_arrays = Off  
magic_quotes_gpc = Off  
extension_dir = /usr/local/mxae/lib
```

Change the session.gc_maxlifetime.

```
; After this number of seconds, stored data will be seen as 'garbage' and  
; cleaned up by the garbage collection process.  
session.gc_maxlifetime = 86400
```

Change the session.cache_expire.

```
; Document expires after n minutes.  
session.cache_expire = 1440
```

PHP Extension Modules

Subsequently, installation of php extension modules is explained. These modules are loaded through php's extension support.

Following are the topic you can find.

- APC (Deprecated since MXAE ver. 1.5.0)
- EACCELERATOR (Deprecated since MXAE ver. 1.5.0)
- POP3
- PECL_HTTP
- FILEINFO
- MAILPARSE
- SSH2
- ADODB
- MNOGOSEARCH
- TURCK-MMCACHE (Deprecated since MXAE ver. 1.0.12)
- ZIP
- IMAGEMAGICK
- XCACHE (Added in MXAE ver. 1.5.2)
- JSON (Added in MXAE ver. 1.5.2)

APC (Deprecated)

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# there is two way to install this module but we prefer to get it from
the cvs (method #2)
# apc: (http://pecl.php.net/package/apc)

shell> cd APC-<VERSION>

shell> export PHP_PREFIX="/usr/local/mxae"

shell> $PHP_PREFIX/bin/phpize

shell> ./configure --enable-apc --enable-apc-mmap --with-
apxs=/usr/local/mxae/apache2/bin/apxs --with-php-
config=$PHP_PREFIX/bin/php-config

shell> make

shell> cp modules/apc.so /usr/local/mxae/lib/

shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini

[apc]
apc.enabled="1"
apc.shm_segments="1"
apc.shm_size="32"
apc.optimization="0"
apc.num_files_hint="1000"
apc.ttl="0"
apc.gc_ttl="3600"
apc.cache_by_default="Off"
apc.filters=""
apc.mmap_file_mask=""
apc.slam_defense="0"
apc.file_update_protection="2"
apc.enable_cli="1"
apc.max_file_size="1M"
```

EACCELERATOR (Deprecated)

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# there is two way to install this module but we prefer to get it from
the cvs (method #2)
# get latest from (http://eaccelerator.sourceforge.net/)

shell> cd eaccelerator-<VERSION>

shell> export PHP_PREFIX="/usr/local/mxae"

shell> $PHP_PREFIX/bin/phpize

shell> ./configure --enable-eaccelerator=shared --with-php-
config=$PHP_PREFIX/bin/php-config

shell> make

shell> cp modules/eaccelerator.so /usr/local/mxae/lib/

shell> mkdir /tmp/mmcache

shell> chmod 777 /tmp/mmcache

shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini

[eaccelerator]
zend_extension="/usr/local/mxae/lib/eaccelerator.so"
zend_extension_ts="/usr/local/mxae/lib/eaccelerator.so"
eaccelerator.shm_size="16"
eaccelerator.cache_dir="/tmp/mmcache"
eaccelerator.enable="1"
eaccelerator.optimizer="0"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
;eaccelerator.filter=""
eaccelerator.filter="!* .php !* .inc"
eaccelerator.shm_max="1M"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="0"
eaccelerator.content="none"
eaccelerator.sessions="none"
eaccelerator.keys="shm"
```

POP3

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://pecl.php.net/package/POP3)
shell> cd /home/users/siteadm/src
shell> tar -zxvf ../packages/pop3-<VERSION>.tgz
shell> cd pop3-<VERSION>/
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config --
enable-pop3
shell> make
shell> cp modules/pop3.so /usr/local/mxae/lib/
shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini
extension=pop3.so
```

PECL_HTTP

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://pecl.php.net/package/pecl\_http)
shell> cd /home/users/siteadm/src
shell> tar -zxvf ../packages/pecl_http-<VERSION>.tgz
shell> cd pecl_http-<VERSION>/
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config --
enable-http --with-http-curl-requests=/usr/local/mxae --with-http-
mhash-etags=/usr/local/mxae --with-http-magic-mime=/usr/local/mxae
shell> make
shell> cp modules/http.so /usr/local/mxae/lib/
shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini
extension=http.so
```

FILEINFO

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from (http://pecl.php.net/package/fileinfo), docs =
# http://wiki.cc/php/Fileinfo

shell> cd /home/users/siteadm/src

shell> tar -zxvf ../packages/file-<VERSION>.tar.gz

shell> cd file-<VERSION>/

shell> ./configure --prefix=/usr/local/mxae

shell> make

shell> make install

shell> cd ..

shell> tar -zxvf ../packages/Fileinfo-<VERSION>.tgz

shell> cd Fileinfo-<VERSION>/

shell> export PHP_PREFIX="/usr/local/mxae"

shell> $PHP_PREFIX/bin/phpize
(you need to edit the file 'configure' and add '/usr/local/mxae' into
every occurrence of the string "SEARCH_PATH")

shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config --
with-fileinfo

shell> make

shell> cp modules/fileinfo.so /usr/local/mxae/lib/

shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini
extension=fileinfo.so
```

MAILPARSE

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from http://pecl.php.net/package/mailparse
shell> cd /home/users/siteadm/src
shell> tar -zxvf ../packages/mailparse-<VERSION>.tgz
shell> cd mailparse-<VERSION>/
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config --
enable-mailparse
shell> make
shell> cp modules/mailparse.so /usr/local/mxae/lib/
shell> vi /usr/local/mxae/lib/php.ini
# Add the following line into php.ini
extension=mailparse.so
```

SSH2

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get latest from http://sourceforge.net/projects/libssh2/
shell> cd /home/users/siteadm/src
shell> tar -zxvf ../packages/libssh2-<VERSION>.tar.gz
shell> cd libssh2-<VERSION>/
shell> ./configure --prefix=/usr/local/mxae --with-
openssl=/usr/local/mxae --with-zlib
shell> make
shell> make install
shell> cd ..
shell> tar -zxvf ../packages/ssh2-<VERSION>.tgz
shell> cd ssh2-<VERSION>/
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config --with-
ssh2=/usr/local/mxae
shell> make
shell> cp modules/ssh2.so /usr/local/mxae/lib/
shell> vi /usr/local/mxae/lib/php.ini
#Add the following line into php.ini
extension=ssh2.so
```

ADODB

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> cd /home/users/siteadm/src
shell> mkdir adodb-ext-<VERSION>
shell> cd adodb-ext-<VERSION>
shell> unzip ../../packages/adodb-ext-<VERSION>.zip
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-php-config=$PHP_PREFIX/bin/php-config
shell> make
shell> cp modules/adodb.so /usr/local/mxae/lib/
```

Edit /usr/local/mxae/lib/php.ini and add the following line:

```
shell> vi /usr/local/mxae/lib/php.ini
#PHP.INI
extension=adodb.so
```

MNOGOSEARCH

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> cd /home/users/siteadm/src
shell> tar -zxvf ../../packages/mnogosearch-php-extension-
<VERSION>.tar.gz
shell> cd mnogosearch-php-extension-<VERSION>
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> CPPFLAGS="-I/usr/local/mxae/mysql/include" ./configure --
enable-mnogosearch=shared --with-
mnogosearch=/usr/local/mxae/mnogosearch --with-php-
config=$PHP_PREFIX/bin/php-config
shell> make
shell> cp modules/mnogosearch.so /usr/local/mxae/lib/
```

Edit /usr/local/mxae/lib/php.ini and add the following line:

```
shell> vi /usr/local/mxae/llib/php.ini

#PHP.INI
extension=mnogosearch.so
```

TURCK-MMCACHE (Deprecated)

Turck MMCache version 2.4.6 is available as of this writing, however, you may obtain the latest package from cvs.

Installation:

```
shell> cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/turck-
mmcache login
CVS password: <ENTER>

shell> cvs -z3 -
d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/turck-mmcache co
turck-mmcache
```

Build the package:

```
shell> cd turck-mmcache

shell> export PHP_PREFIX="/usr/local/mxae"

shell> $PHP_PREFIX/bin/phpize

shell> ./configure --enable-mmcache=shared --with-php-
config=$PHP_PREFIX/bin/php-config

shell> make

shell> cp modules/mmcache.so /usr/local/mxae/lib/

shell> mkdir /tmp/mmcache

shell> chmod 777 /tmp/mmcache
```

Edit /usr/local/mxae/lib/php.ini and add the following line:

```
zend_extension="/usr/local/mxae/lib/mmcache.so"
zend_extension_ts="/usr/local/mxae/lib/mmcache.so"
mmcache.shm_size="16"
mmcache.cache_dir="/tmp/mmcache"
mmcache.enable="1"
mmcache.optimizer="0"
mmcache.check_mtime="1"
mmcache.debug="0"
;mmcache.filter=""
mmcache.filter="!*.*php !*.*inc"
mmcache.shm_max="1M"
mmcache.shm_ttl="0"
mmcache.shm_prune_period="0"
mmcache.shm_only="0"
mmcache.compress="0"
mmcache.content="none"
mmcache.sessions="none"
```

```
mmcache.keys="shm"
```

ZIP

Download the source file to /home/users/siteadm/packages from <http://zzip.lib.sourceforge.net/>

Installation:

```
shell> cd /home/users/siteadm/src
shell> tar jxvf ../packages/zzip-lib-<VERSION>.tar.bz2
shell> cd zzip-lib-<VERSION>
shell> ./configure --prefix=/usr/local/mxae
shell> make
shell> make install
shell> cd ..
shell> tar zxvf ../packages/zip-<VERSION>.tgz
shell> cd zip-<VERSION>/
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --enable-zip=shared --with-zip=/usr/local/mxae --
with-php-config=$PHP_PREFIX/bin/php-config
shell> make
shell> cp modules/zip.so /usr/local/mxae/lib/
```

Edit /usr/local/mxae/lib/php.ini and add the following line:

```
shell> vi /usr/local/mxae/lib/php.ini
#PHP.INI
extension=zip.so
```

IMAGEMAGICK

Download the source file to /home/users/siteadm/packages from <http://www.imagemagick.org>

Installation:

```
# get from http://www.imagemagick.org/index.php

shell> tar -zxvf ../packages/ImageMagick-6.1.9-4.tar.gz
shell> cd ImageMagick-6.1.9
shell> ./configure --prefix=/usr/local/mxae
shell> make
shell> make install

# imagick php extension support (http://pecl.php.net/package/imagick)

shell> tar -zxvf ../packages/imagick-0.9.11.tgz
shell> ./configure --with-imagick=/usr/local/mxae --with-php-
config=/usr/local/mxae/bin/php-config
shell> export PHP_PREFIX="/usr/local/mxae"
shell> $PHP_PREFIX/bin/phpize
shell> ./configure --with-imagick=/usr/local/mxae --with-php- shell>
config=$PHP_PREFIX/bin/php-config
shell> make
shell> cp modules/imagick.so /usr/local/mxae/lib/
shell> vi /usr/local/mxae/lib/php.ini:

# add the following into the file
extension=imagick.so
```

XCACHE

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> cd xcache

shell> export PHP_PREFIX="/usr/local/mxae"

shell> $PHP_PREFIX/bin/phpize

shell> ./configure --enable-xcache --with-php-
config=$PHP_PREFIX/bin/php-config

shell> make

shell> cp modules/xcache.so $PHP_PREFIX/lib/

shell> vi /usr/local/mxae/lib/php.ini

# Add the following line into php.ini
[xcache-common]
;; install as zend extension (recommended), normally
"$extension_dir/xcache.so"
zend_extension = /usr/local/mxae/lib/xcache.so

[xcache.admin]
xcache.admin.user = "admin"
; xcache.admin.pass = md5($your_password)
xcache.admin.pass = "bc5fd8097441976f0e1bc087c49d5708"

[xcache]
; ini only settings, all the values here is default unless explained

; select low level shm/allocator scheme implementation
xcache.shm_scheme = "mmap"
; to disable: xcache.size=0
; to enable : xcache.size=64M etc (any size > 0) and your system mmap
allows
xcache.size = 0M
; set to cpu count (cat /proc/cpuinfo |grep -c processor)
xcache.count = 2
; just a hash hints, you can always store count(items) > slots
xcache.slots = 8K
; ttl of the cache item, 0=forever
xcache.ttl = 0
; interval of gc scanning expired items, 0=no scan, other values is in
seconds
xcache.gc_interval = 0

; same as aboves but for variable cache
xcache.var_size = 128M
xcache.var_count = 2
xcache.var_slots = 8K
; default ttl
xcache.var_ttl = 0
xcache.var_maxttl = 0
xcache.var_gc_interval = 300

xcache.test = Off
; N/A for /dev/zero
xcache.readonly_protection = Off
; for *nix, xcache.mmap_path is a file path, not directory.
; Use something like "/tmp/xcache" if you want to turn on
ReadonlyProtection
; 2 group of php won't share the same /tmp/xcache
; for win32, xcache.mmap_path=anonymous map name, not file path
xcache.mmap_path = "/dev/zero"

; leave it blank(disabled) or "/tmp/phpcore/"
; make sure it's writable by php (without checking open_basedir)
xcache.coredump_directory = ""

; per request settings
xcache.cacher = On
xcache.stat = On
xcache.optimizer = Off

[xcache.coverager]
; per request settings
; enable coverage data collecting for xcache.coveragedump_directory
and xcache_coverager_start/stop/get/clean() functions (will hurt
executing performance)
xcache.coverager = Off

; ini only settings
```

JSON

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# json php extension support (http://www.aurore.net/projects/php-  
json/)  
  
shell> tar zxvf ../packages/php-json-ext-1.2.1.tar.bz2  
  
shell> cd php-json-ext-1.2.1/  
  
shell> export PHP_PREFIX="/usr/local/mxae"  
  
shell> $PHP_PREFIX/bin/phpize  
  
shell> ./configure --with-json --with-php-config=$PHP_PREFIX/bin/php-  
config  
shell> make  
  
shell> cp modules/json.so $PHP_PREFIX/lib/  
  
shell> vi /usr/local/mxae/lib/php.ini  
#add the following line  
extension=json.so
```

Installing and Configuring Other Applications

In this chapter, there are still some applications that required. Application's definition is included as well as the installation.

Following are the topic you can find.

- AWSTATS
- GEO-IPFREE
- XDEBUG
- DBI
- DBD-MYSQL
- XPDF
- OPENSSSH
- PROFTPD
- PHPMYADMIN
- TIME SYN

AWSTATS

Download the source file to /home/users/siteadm/packages/.

Installation:

```
# get from (http://awstats.sourceforge.net/)
shell> tar -zxvf ../packages/awstats-<VERSION>.gz -C /usr/local/mxae/
shell> cd /usr/local/mxae
shell> mv awstats-<VERSION> awstats
shell> chmod 755 awstats
shell> chown -R root.root awstats
shell> cd awstats
shell> mkdir data
shell> chmod 755 data
shell> mkdir wwwroot/cgi-bin/etc
```

Modify the script /usr/local/mxae/awstats/wwwroot/cgi-bin/awstats.pl as follows:

```
sub Read_Config {
    ...
    if ($configdir) { @PossibleConfigDir=("$configdir"); }
    else {
    @PossibleConfigDir=("$DIR","$DIR/etc","/etc/awstats","/usr/local/etc/awstats","/etc","/etc/opt/awstats");
    }

    ^-- added this directory to include in the search of config file(s)
```

GEO-IPFREE

Download the source file to /home/users/siteadm/packages/. This is a plugin for awstats.

```
# get from http://www.cpan.org/authors/id/G/GM/GMPASSOS/Geo-IPfree-0.2.tar.gz
shell> tar -zxvf ../packages/Geo-IPfree-<version>.tar.gz
shell> cp -R Geo-IPfree-<version>/lib/Geo /usr/local/mxae/awstats/wwwroot/cgi-bin/plugins/
```

XDEBUG

Version xdebug1.3.2 is available as of this writing, however it is advisable to obtain the latest from cvs.

```
# get from (http://xdebug.org/)
shell> cvs -d :pserver:srmread@cvs.xdebug.org:/repository login
CVS password: srmread
shell> cvs -d :pserver:srmread@cvs.xdebug.org:/repository co xdebug
```

Then proceed to installation

```

shell> cd xdebug/

shell> /usr/local/mxae/bin/phpize

shell> ./configure --enable-xdebug --with-php-
config=/usr/local/mxae/bin/php-config

shell> make

shell> cp modules/xdebug.so /usr/local/mxae/lib/

```

Edit `/usr/local/mxae/lib/php.ini` and add the following line:

```

shell> vi /usr/local/mxae/lib/php.ini

[xdebug]
xdebug.auto_trace = on
xdebug.default_enable = on
xdebug.auto_profiler = on
xdebug.profiler_enable = on
xdebug.output_dir = "/tmp/"
xdebug.auto_profile_mode = XDEBUG_PROFILER_FS_AV
xdebug.profiler_output_dir = "/tmp/"
xdebug.trace_output_dir = "/tmp/"
zend_extension="/usr/local/mxae/lib/xdebug.so"
zend_extension_ts="/usr/local/mxae/lib/xdebug.so"

```

DBI

Download the source file to `/home/users/siteadm/packages/`. The Perl Database Interface. (DBI) requires one or more 'driver' modules to talk to databases after its installation such as `dbd-mysql`, but they are not needed to during the DBI installation.

Installation:

```

shell> tar -zxvf DBI-<version>.tar.gz -C /home/users/siteadm/src
perl Makefile.PL

shell> make

shell> make test && make install

```

DBD-MYSQL

Download the source file to `/home/users/siteadm/packages/`.

Installation:

```

shell> tar -zxvf DBD-mysql-<version>.tar.gz -C /home/users/siteadm/src
perl Makefile.PL

shell> make

shell> make test && make install

```

XPDF

Download the source file to `/home/users/siteadm/packages/`.

Installation:

```
# get from (http://www.foolabs.com/xpdf/)
shell> tar -zxvf /home/users/siteadm/packages/xpdf-<VERSION>-
<OS>.tar.gz -C /home/users/siteadm/src
shell> cd /home/users/siteadm/src/xpdf-<VERSION>-<OS>
shell> ./configure --prefix=/usr/local/mxae
# ignore any warning about X or Motif libraries... don't really need
xpdf anyway, only need pdftotext
shell> make && make install
```

OpenSSH

In this chapter, steps to upgrade ssh is covered. Besides that, security setting also included in order to provide secure connection.

Download openssh rpm packages from www.openssh.org and save into `/home/users/siteadm/packages`.

```
# Install and upgrade the openssh server and client
shell> rpm -Uvh /home/users/siteadm/packages/openssh*
# Edit the config file, please refer to Appendix 1 for guideline.
shell> vi /etc/ssh/sshd_config
```

Change the respective settings for `/etc/ssh/sshd.config` as below,

```
# /etc/ssh/sshd_config
Port 22
Protocol 2
PermitRootLogin no
PasswordAuthentication yes
PermitEmptyPasswords no
```

After completing the changes for the configuration file, then you need to restart the openssh in order for the new configuration to take affect.

```
# Restart ssh after configuration file edit.
shell> /etc/rc.d/init.d/sshd restart
```

PROFTPD

ProFTPD grew out of the desire to have a secure and configurable FTP server, and out of a significant admiration of the Apache web server. Proftpd is a ftp server for unix.

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> tar -zxvf /home/users/siteadm/packages/proftpd-VERSION.tar.gz -
C/home/users/siteadm/src/

shell> cd /home/users/siteadm/src/proftpd-VERSION

shell> ./configure --prefix=/usr/local/mxae

shell> make

shell> make install

# Following might not be applicable if proftpd.conf already in /etc

shell> cp /usr/local/etc/proftpd.conf /etc/

shell> vi /etc/proftpd.conf
```

Make the proftpd.conf file looks like below. Please remove the anonymous section in order to disallow anonymous login. The following config assumes the ftp server running in 'standalone' mode, which is recommended.

```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use. It establishes a single server
# and a single anonymous login. It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName                "Your Server FTP"
ServerType                 standalone
DefaultServer             on

# Port 21 is the standard FTP port.
Port                       21

# Umask 022 is a good standard umask to prevent new dirs and files
# from being group and world writable.
Umask                     022

# To prevent DoS attacks, set the maximum number of child processes
# to 30. If you need to allow more than 30 concurrent connections
# at once, simply increase this value. Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd).
MaxInstances               30
```

```
# Set the user and group under which the server will run.
User          nobody
Group         nobody

# To cause every FTP user to be "jailed" (chrooted) into their home
# directory, uncomment this line.
DefaultRoot ~ !netsite

# Normally, we want files to be overwriteable.
<Directory />
  AllowOverwrite          on
</Directory>

# Bar use of SITE CHMOD by default
<Limit SITE_CHMOD>
  DenyAll
</Limit>
```

After finish the configuration, then you can start the ftp service,

```
shell> /usr/local/mxae/sbin/proftpd -c /usr/local/mxae/etc/proftpd.conf
```

Then you need to make the ftp server to start automatically:

```
shell> vi /etc/rc.d/rc.local

# Append following statement into the file

echo "Starting proftpd server"
/usr/local/mxae/sbin/proftpd -c /usr/local/mxae/etc/proftpd.conf

# save and exit
```

There is another way of having the ftp server to start automatically and it will also become a transparent services that running at the backend. This is to make the ftp server to be one of xinetd services.

To do so, you need to change the ServerType in proftpd.conf.

```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use. It establishes a single server
# and a single anonymous login. It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName          "Your Server FTP"
ServerType          inetd
DefaultServer       on
```

It is not done yet; you need to create a file call proftpd in order to have the services running. The file will locate in /etc/xinetd.d/.

```
shell> vi /etc/xinetd.d/proftpd

# Add following into the file
```

```

service ftp
{
    disable = no
    flags    = REUSE
    socket_type = stream
    instances = 50
    wait     = no
    user     = root
    server   = /usr/local/mxae/sbin/proftpd
    server_args = -c /etc/proftpd.conf
    bind     = 127.0.0.1, <change to your ip>
}

# save and exit
# Then you need to change the owner and the permission
shell> chown root.root /etc/xinetd.d/proftpd

shell> chmod 644 /etc/xinetd.d/proftpd

```

PHPMYADMIN

Installation

You need to know where your web server's DocumentRoot is. For Apache, the default DocumentRoot is /usr/local/apache/htdocs/.

```

shell> tar -zxvf /home/users/siteadm/packages/proftpd-VERSION.tar.gz -C
/usr/local/mxae/apache2/htdocs/mxapp/

shell> mv /usr/local/mxae/apache2/htdocs/mxapp/phpMyAdmin-VERSION
/usr/local/mxae/apache2/htdocs/mxapp/phpMyAdmin

shell> vi /usr/local/mxae/apache2/htdocs/mxapp/phpMyAdmin/config.inc.php

# Make changes to the file as below,
$config['PmaAbsoluteUri_DisableWarning'] = TRUE

# Change the host value
$config['Servers'][$i]['host']          = 'server-ip'; # or
$config['Servers'][$i]['host']          = 'localhost';

# Change the authentication mode to http
$config['Servers'][$i]['auth_type']     = 'http';

```

You can add more `$config['Servers'][$i]['host']` into the config.inc.php. You can see more arrays setting when you scroll down. Add another server-ip and change the authentication. Then from phpMyAdmin, you are able to select which server's MySQL then you want to connect to.

Proceed to modify grant table of mysql server to allow phpmyadmin user with sufficient privileges to modify databases. Login to mysql server to do so.

```

shell> /usr/local/mxae/mysql/bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
mysql> GRANT SELECT ON mysql.* to phpMyAdmin@localhost IDENTIFIED BY '<password>';
Query OK, 0 rows affected (0.00 sec)
mysql> quit

```

TIME SYN

Download the source file to /home/users/siteadm/packages/.

Installation:

```
shell> cd /home/users/siteadm/packages/  
shell> rpm -I ntp-version.rpm
```

NTP will check your server time and will synchronize the server time for you. So, you need to edit the crond to do so.

```
shell> crontab -e
```

Add Following line,

```
0 */6 * * * /usr/sbin/ntpdate time.nrc.ca > /dev/null 2>&1
```

Every 10 minutes, NTP will do the checking.

ZendOptimizer

The Zend Optimizer is a free application that runs the files encoded by the Zend Encoder and Zend SafeGuard Suite, while enhancing the running speed of PHP applications.

Enables users to run files encoded by the Zend Encoder and increases runtime performance up to 40%.

Download the source file to /home/users/siteadm/packages/.

```
# get from http://www.zend.com  
shell> tar -zxvf /home/users/siteadm/packages/ZendOptimizer-VERSION.tar.gz -C  
/home/users/siteadm/src/  
shell> cd /home/users/siteadm/src/ZendOptimizer-VERSION  
# After issued following command, just follow the wizard  
shell> ./install.sh
```

Note: Upon installation of Zend Optimizer, the original php.ini file in /usr/local/mxae/lib/php.ini will be moved to /usr/local/mxae/Zend/etc/php.ini and a symbolic link to /usr/local/mxae/Zend/etc/php.ini will be created for /usr/local/mxae/lib/php.ini. It is always safe to edit /usr/local/mxae/lib/php.ini as before.

MXAE Installation

In this chapter, how to install MXAE and some settings configuration will be explained and demonstrated

- FTPONLY
- MXAE SYSTEM
- FILE PERMISSION
- DATABASE PERMISSION
- LICENSING
- ROUTINE SCRIPTS
- LOGROTATE
- REPORT
- CRONTABS

FTP Only

Ftponly is file based, you need to get the file from MXAE team and save the file as shown in Appendix, then you need to add ftponly to user's shell.

```
shell> export PATH=${PATH}:/usr/local/mxae/bin/
shell> chmod 755 /usr/local/mxae/apache2/htdocs/mxlib/resource/system/ftponly
shell> ln -s /usr/local/mxae/apache2/htdocs/mxlib/resource/system/ftponly
/bin/ftponly
shell> echo "/bin/ftponly" >> /etc/shells
shell> chmod 755 /bin/ftponly
shell> chown root.root /bin/ftponly
```

MXAE System

In MXAE, there are two important directories. They are

- mxapp
- mxlib

Obtain these two directories from MXAE team and perform the following steps.

```

shell> cp -R /path-of-dir/mxapp /usr/local/mxae/apache2/htdocs/
shell> cp -R /path-of-dir/mxlib /usr/local/mxae/apache2/htdocs/

# Create a folder to store development websites (only if publisher and ams is on the
same machine)

shell> mkdir /usr/local/mxae/apache2/htdocs/websites

shell> chmod 755 /usr/local/mxae/apache2/htdocs/websites

shell> ln -s /usr/local/mxae/apache2/htdocs/websites
/usr/local/mxae/apache2/htdocs/mxapp/websites

```

Next, create the necessary database by insert data from sql dump file for the MXAE system.

```

shell> cp path_to/mxdb-system.sql /home/users/siteadm/packages
shell> cp path_to/mxdb-content.sql /home/users/siteadm/packages

shell> /usr/local/mxae/mysql/bin/mysql -u root -p'<password>' -e 'CREATE DATABASE
mxsystem'

shell> /usr/local/mxae/mysql/bin/mysql -u root -p'<password>' -D mxsystem <
/home/users/siteadm/packages/sqldump/mxdb-system.sql

```

M

ake sure to obtain a valid license key for the new server before proceeding. Then, update the MXAE config.ini file /usr/local/mxae/apache2/htdocs/mxapp/config.ini with the following:

```

shell> vi /usr/local/mxae/apache2/htdocs/mxapp/config.ini

# MXAE config.ini
dbHost = 202.XXX.XXX.XXX:XXXX
dbType = mysql
dbUsername = <login>
dbPassword = <password>
dbName = mxsystem
agentip = 202.XXX.XXX.XXX

```

File Permission

File permission plays a very important role in security. You must make sure that only the granted user has the privileges to read, edit or remove the file in MXAE.

```
shell> chown siteadm.netsite /usr/local/mxae/apache2/htdocs
shell> chmod 775 /usr/local/mxae/apache2/htdocs
shell> chown -R siteadm.netsite /usr/local/mxae/apache2/htdocs/mxapp
shell> chown -R siteadm.netsite /usr/local/mxae/apache2/htdocs/mxlib
# to allow logging by AMS
shell> chmod -R 777 /usr/local/mxae/apache2/htdocs/mxapp/logs
# to allow logging by mxAgent
shell> chmod -R 777 /usr/local/mxae/apache2/htdocs/mxlib/agent/logs
# to allow editing by AMS (optional)
shell> chmod 777 /usr/local/mxae/apache2/htdocs/mxapp/config.ini
shell> chmod 777 /usr/local/mxae/apache2/htdocs/mxapp/resource/theme/default.ini
```

Change other directories than htdocs inside apache to httpd user or siteadm user. It is recommended other directories owner is root user.

Database Permission

Same as other applications, MXAE also need to have granted users to have some privileges to database.

Perform following steps to create users and grant privileges for MySQL

Following are the grant statements; change the IP below to the IP of your server.

```
shell> /usr/local/mxae/mysql/bin/mysql -u root -p
# Enter root password
mysql> GRANT SELECT ,
INSERT ,
UPDATE ,
DELETE ,
CREATE ,
DROP ,
RELOAD ,
SHUTDOWN ,
PROCESS ,
INDEX ,
ALTER ,
SHOW DATABASES ,
CREATE TEMPORARY TABLES ,
LOCK TABLES ON * . * TO mxae@<SERVER_IP> IDENTIFIED BY '<password>' WITH GRANT OPTION
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 ;
mysql> flush privileges;
mysql> exit;
```

Licensing

Licensing control the service that provided by MXAE. To activate the licensing, there are two ways of doing.

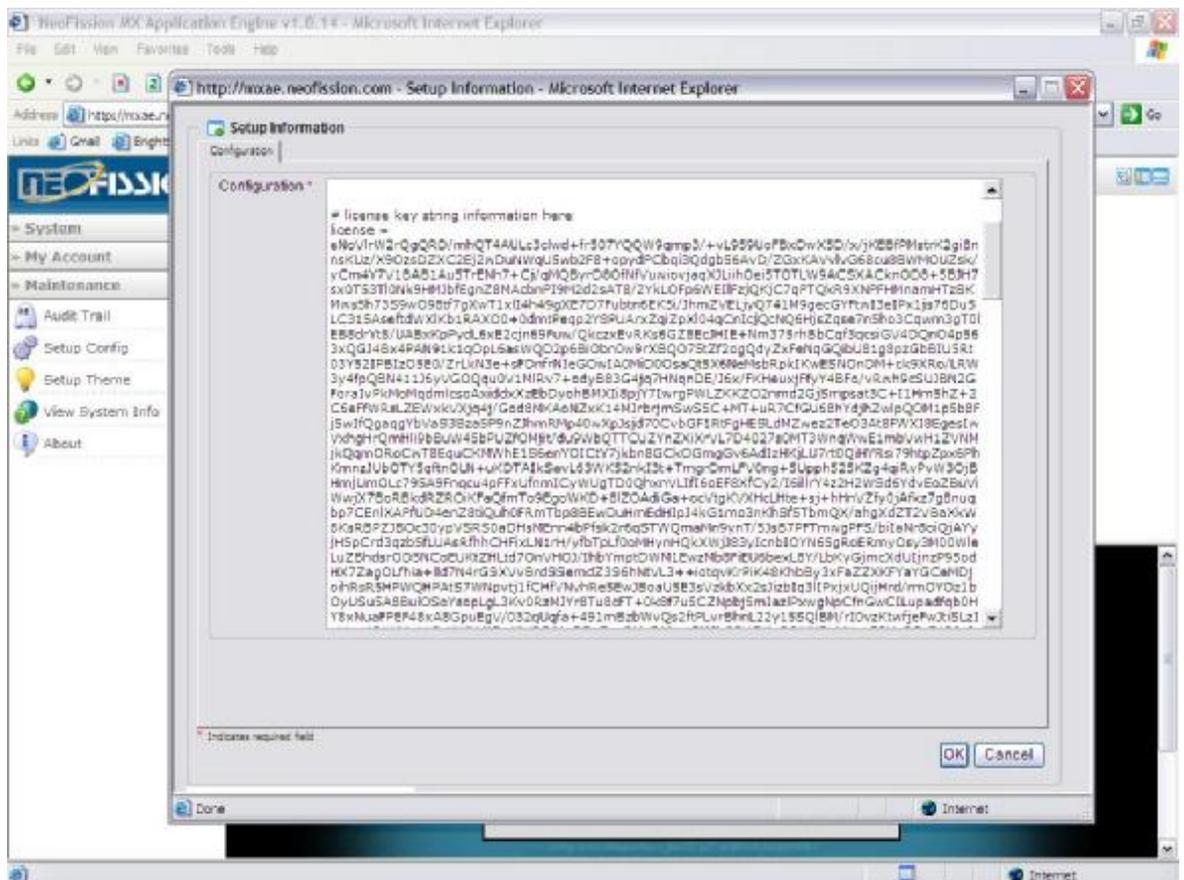
Via Browser Configuration

For the first time, this is the recommended way, MXAE will do the necessarily update for you.

Before you open the browser, you need to change config.ini file permission in order the update to be successful.

```
shell> /usr/local/mxae/apache2/htdocs/mxapp
shell> chmod 666 config.ini
```

Then go to the browser, type in the url and the main page will appear. Login as admin user of MXAE system. Click on 'Maintenance -> Setup Config' at the left panel bar to open a pop up window that allows you to input the license key appropriately.



After being successfully registered then you to change the config.ini file permission again to be read only.

```
shell> cd /usr/local/mxae/apache2/htdocs/mxapp/
shell> chmod 444 config.ini
```

Via System Configuration

Using this method is shown that you already understand all the variables in config.ini. If you are not sure, please do not change anything.

```
shell> cd /usr/local/mxae/apache2/htdocs/mxapp
shell> chmod 777 config.ini
shell> vi config.ini
# Change the value 'license' below the comment # license key string information here
license = <license key>
# Save and exit
shell> chmod 444 config.ini
```

Routine Scripts

Routine scripts are scripts that developed by MXAE team that will help to maintain the server, to provide backup and etc. All routines scripts are developed in php and executed with crond.

Logrotate

Logrotate is to rotate apache log files so that the files won't get too big. It apply logrotate service that provided by Redhat. You need to create a file name httpd in logrotate.d directory in order to activate the server.

Report

This is to install a regular interval server health reporting mechanism.

```
# If bin directory not created
shell> mkdir /usr/local/mxae/apache2/htdocs/mxlib/resource/system

shell> cp /path-of-file/report.sh
/usr/local/mxae/apache2/htdocs/mxlib/resource/system

shell>chmod 755 /usr/local/mxae/apache2/htdocs/mxlib/resource/system

shell> chown -R root.root /usr/local/mxae/apache2/htdocs/mxlib/resource/system
```

report.sh will send report to the person applying emails under RECIPIENT. You need to add or remove the emails accordingly.

```
# Add or remove email to RECIPIENT  
RECIPIENT="sysadmin@yourhost.com"
```

Again, same as routine scripts, report.sh apply crond to active the reporting sending. You will need to edit the crond for it.

```
shell> crontab -e
```

And add following,

```
35 */6 * * * /home/users/siteadm/bin/report.sh > /dev/null 2>&1
```

The report will be sent every 6 six hours.

Same as routines scripts, you need to get from respective person.

Crontabs

Crontabs are regular tasks that must be performed automatically by the system to ensure that the system runs smoothly or to execute routine services such as backup and monitoring.

Appendix C – sshd_config

```
# $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.
#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::
# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 3600
#ServerKeyBits 768
# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO
# Authentication:
#LoginGraceTime 120
PermitRootLogin no
#StrictModes yes
#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile      .ssh/authorized_keys
# rhosts authentication should not be used
#RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
PermitEmptyPasswords no
# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#AFSTokenPassing no
```

```
# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no

# Set this to 'yes' to enable PAM keyboard-interactive authentication
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt no

#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#KeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression yes

#MaxStartups 10
# no default banner path
#Banner /some/path
#VerifyReverseMapping no

# override default of no subsystems
Subsystem      sftp      /usr/libexec/openssh/sftp-server
```

Appendix D – proftpd.conf

```
ServerName          "My FTP Server"
ServerType          standalone
DefaultServer      on
UseReverseDNS      off
IdentLookups       off
Port               21
Umask              022
MaxInstances       30
User               nobody
Group              nobody
DefaultRoot        ~/!netsite
<Directory /*>
    AllowOverwrite  on
</Directory>
# Bar use of SITE CHMOD by default
<Limit SITE_CHMOD>
    DenyAll
</Limit>
```

Appendix E – xinetd.d (proftpd service)

```
service ftp
{
    disable = no
    flags   = REUSE
    socket_type = stream
    instances = 50
    wait    = no
    user    = root
    server  = /usr/local/sbin/proftpd
    server_args = -c /etc/proftpd.conf
    bind    = 127.0.0.1, <change to your ip>
```

```
}

```

Appendix F – logrotate

```
# mxae log rotation (keep log up to 8 weeks)
# create at /etc/logrotate.d/mxae (for ams only)
/usr/local/mxae/apache2/htdocs/mxapp/logs/mxae.log {
    rotate 8
    weekly
    compress
}

# mxae's agent log rotation (keep log up to 8 weeks)
# create at /etc/logrotate.d/mxagentd (for publisher only)
/usr/local/mxae/apache2/htdocs/mxlib/agent/logs/mxac.log {
    rotate 8
    weekly
    compress
    postrotate
        /usr/local/mxae/bin/php -q
        /usr/local/mxae/apache2/htdocs/mxlib/agent/agentd.php -c
        /usr/local/mxae/apache2/htdocs/mxapp/config.ini restart
    endscript
}

# apache log rotation (keep log up to 8 weeks)
# create at /etc/logrotate.d/httpd (for publisher only)
/usr/local/mxae/apache2/logs/*_log {
    rotate 8
    weekly
    compress
    postrotate
        /usr/local/mxae/apache2/bin/apachectl graceful > /dev/null 2>&1
    endscript
}

```

Appendix G – reporting

```
#!/bin/sh

SERVER=`/bin/hostname`
RECIPIENT="sysadmin@yourhost.com"

echo "To: $RECIPIENT" > ~/mail.stat
echo "From: System <root@yourhost.com>" >> ~/mail.stat
echo "Subject: [SRV] $SERVER" >> ~/mail.stat
echo "Content-Type: text/html;" >> ~/mail.stat
echo "" >> ~/mail.stat
echo "<PRE>" >> ~/mail.stat
date >> ~/mail.stat
echo "" >> ~/mail.stat
uptime >> ~/mail.stat
echo "" >> ~/mail.stat
echo "=====  
DISKSPACE REPORT  
=====" >> ~/mail.stat
df -k >> ~/mail.stat
echo "" >> ~/mail.stat
echo "=====  
PROCESS REPORT  
=====" >> ~/mail.stat

```

```

ps aux >> ~/mail.stat
echo "" >> ~/mail.stat
echo "===== TOP (1) REPORT =====" >> ~/mail.stat
top -b -n 1 >> ~/mail.stat
echo "" >> ~/mail.stat
echo "===== TOP (2) REPORT =====" >> ~/mail.stat
top -b -n 1 >> ~/mail.stat
echo "" >> ~/mail.stat
echo "===== VMSTAT REPORT =====" >> ~/mail.stat
vmstat 2 10 >> ~/mail.stat
echo "" >> ~/mail.stat
echo "===== NETSTAT (1) REPORT =====" >> ~/mail.stat
netstat -pa >> ~/mail.stat
echo "" >> ~/mail.stat
echo "===== NETSTAT (2) REPORT =====" >> ~/mail.stat
netstat -pa >> ~/mail.stat
echo "</PRE>" >> ~/mail.stat
/usr/sbin/sendmail -t < ~/mail.stat

```

Appendix H – virtual host listing in httpd.conf

```

NameVirtualHost <SERVER_IP>:80
<VirtualHost <SERVER_IP>:80>
    DocumentRoot /usr/local/mxae/apache2/htdocs/mxapp
    ServerName <SERVER_IP> or <SERVER_NAME>
    <Directory "/usr/local/mxae/apache2/htdocs/mxapp/phpMyAdmin/">
        Options None
        AllowOverride None
        Order allow,deny
        # replace with any restricted IPs that u allow access to phpMyAdmin
        Allow from <IP_ALLOWED>
    </Directory>
    ErrorLog logs/ams-error_log
    CustomLog logs/ams-access_log combined
    #
    # Directives to allow use of AWStats as a CGI
    #
    Alias /awstatsclasses/ "/usr/local/mxae/awstats/wwwroot/classes/"
    Alias /awstatscss/ "/usr/local/mxae/awstats/wwwroot/css/"
    Alias /awstatsicons/ "/usr/local/mxae/awstats/wwwroot/icon/"
    ScriptAlias /awstats/ "/usr/local/mxae/awstats/wwwroot/cgi-bin/"
    #
    # This is to permit URL access to scripts/files in AWStats directory
    #
    <Directory "/usr/local/mxae/awstats/wwwroot/cgi-bin">
        Options None
        AllowOverride None
        Order allow,deny
        # replace with AMS IP
        Allow from <IP_ALLOWED>
    </Directory>
</VirtualHost>

```

Appendix I – ftponly

```

#!/bin/sh
#
# ftponly shell
#
trap "/bin/echo Sorry; exit 0" 1 2 3 4 5 6 7 10 15
#
IFS=""
Admin=root@localhost
System=`/usr/ucb/hostname` `/usr/bin/domainname`

```

```

#
/bin/echo
/bin/echo "*****"
/bin/echo "    You are NOT allowed interactive access to $System."
/bin/echo
/bin/echo "    User accounts are restricted to ftp and web access."
/bin/echo
/bin/echo "    Direct questions concerning this policy to $Admin."
/bin/echo "*****"
/bin/echo
#
# C'ya
#
exit 0

```

Appendix J – crontab

```

put it in root's crontab:
# perform time sync with a time server every 6 hours (make sure this server can
connect to the time server)
0 */6 * * * /usr/sbin/ntpdate time.nrc.ca > /dev/null 2>&1
# restart apache server every midnight (publisher and ams)
30 0 * * * /usr/local/mxae/apache2/bin/apachectl graceful >/dev/null 2>&1
# mxae's worker process to run everyday 4am, 8am, 1pm, 5pm & 10pm hourly (ams only)
#0 4,8,13,17,22 * * * /usr/local/mxae/bin/php -q
/usr/local/mxae/apache2/htdocs/mxlib/agent/worker.php -c
/usr/local/mxae/apache2/htdocs/mxapp/config.ini >/dev/null 2>&1
# mxae's worker process to run every two hourly everyday (ams only)
0 */2 * * * /usr/local/mxae/bin/php -q
/usr/local/mxae/apache2/htdocs/mxlib/agent/worker.php -c
/usr/local/mxae/apache2/htdocs/mxapp/config.ini >/dev/null 2>&1
# run every monday morning 1am immediately after logrotate (to process web log
history file) (ams only)
0 1 * * 1 /usr/local/mxae/bin/php -q
/usr/local/mxae/apache2/htdocs/mxlib/agent/worker.php -c
/usr/local/mxae/apache2/htdocs/mxapp/config.ini -w 1 >/dev/null 2>&1
# restart mxagentd server every midnight (publisher only)
0 0 * * * /usr/local/mxae/bin/php -q
/usr/local/mxae/apache2/htdocs/mxlib/agent/agentd.php -c
/usr/local/mxae/apache2/htdocs/mxapp/config.ini restart >/dev/null 2>&1
# run weblog processing every morning at 4am (publisher only)
# 0 4 * * * /usr/local/mxae/awstats/tools/awstats_updateall.pl now -
configdir=/usr/local/mxae/awstats/wwwroot/cgi-bin/etc -
awstatsprog=/usr/local/mxae/awstats/wwwroot/cgi-bin/awstats.pl >/dev/null 2>&1
# weekly restart of mysql for stability and start new log file purposes (midnight of
every monday - database server only)
0 1 * * 1 /sbin/service mysqld restart >/dev/null 2>&1
# analyze and optimize all dbs and tables. do this during off peak hours. (everyday
3:30am - database server only)
# GRANT SELECT , INSERT ON * . * TO "analyzer"@"localhost" IDENTIFIED BY "*****"
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;
30 3 * * * /usr/local/mxae/mysql/bin/mysqlcheck -Aa -uanalyzer -p<password>

```

Appendix K – Links

Apache
<http://www.apache.org>

PHP
<http://www.php.net>

MySQL
<http://www.mysql.com>

LibXML
<http://xmlsoft.org/sources>

FreeType
<http://www.freetype.org>

OpenSSL
<http://www.openssl.org>

IMAP
<ftp://ftp.cac.washington.edu/imap>

Mhash
<http://mhash.sourceforge.net/>

LibMcrypt
<http://mcrypt.hellug.gr/>

FreeTDS
<http://www.freetds.org/>

CURL
<http://curl.haxx.se/>

ADODB
<http://adodb.sourceforge.net/>

Mnogosearch
<http://search.mnogo.ru/>

Turck-MMcache
<http://turck-mmcache.sourceforge.net/>

ImageMagick
<http://www.imagemagick.org>

Awstats
<http://awstats.sourceforge.net/>

Geo_IPFree
<http://search.cpan.org/~gmpassos/Geo-IPfree-0.2/>

Xdebug
<http://www.xdebug.org>

DBI
<http://dbi.perl.org/>

DBD-MySQL
<http://search.cpan.org/dist/DBD-mysql/>

XPDF
<http://www.xpdf.org>

OpenSSH
<http://www.openssh.org>

Proftpd
<http://www.proftpd.org/>

