

# **SMTPLUS Installation Guide 1.0**

Document Number LX01-0004-01

August 2, 2004

Sine Nomine Associates

43596 Blacksmith Square  
Ashburn, VA 20147

**Release 1.0**



---

**SMTPPLUS**  
**Installation Guide**



---

**SMTPPLUS**  
**Installation Guide**

**First edition, March 2004**

This edition applies to the 1.0 release of SMTPPLUS.

© Copyright Sine Nomine Associates 2004

# Contents

<b>Introduction</b> .....	1
<b>Introduction</b> .....	2
SMTPPLUS Features .....	2
High-Performance Mail Processing .....	2
Automated Virus Scanning .....	2
Automated Spam Detection, Tagging and Rejection .....	3
Automated Removal of Hostile Payloads .....	3
Compatibility with VM SMTP Spool and BSMTP MAILER Interfaces .....	3
Processing Engine .....	3
Application Proxy .....	3
Data Flow Through SMTPPLUS .....	4
Integration Points .....	4
<b>Preparing Your System for SMTPPLUS Installation</b> .....	5
System Requirements .....	5
Hardware .....	5
Software .....	5
Virtual Machine Storage Resources .....	5
Connectivity Resources .....	5
DASD Requirements .....	6
<b>SMTPPLUS Install Planning</b> .....	7
The SMTPPLUS Install Worksheet .....	7
SMTPPLUS Install Type .....	7
Proxy Install .....	7
MX-Style Install .....	7
Mail Processing Policy Decisions .....	8
Host Masquerading .....	8
Host Access Decisions .....	8
IP Address Access Lists .....	9
Host Relaying and Authentication Decisions .....	9
Hosts Permitted to Use SMTPPLUS as a Relay .....	9
SMTP TLS Authentication .....	9
Alias and Delivery Policy .....	10
Use of a MAILER .....	10
Use of LDAP Map .....	10
Use of Local and LDAP-Resolved Aliases .....	10
Virus Scanning Policy Decisions .....	10
Default Action when Virus Detected .....	10
Retention of Infected Files .....	10
Automated Removal of Certain Attachment Types .....	10
Enable Automated Virus Pattern File Updates .....	11
Spam Abatement Policy Decisions .....	11
Spam Tagging and Rejection Levels .....	11
Default Action when Spam Detected .....	11
Retention of Spam Messages .....	11
Connecting the SMTPPLUS Processing Engine to VM TCPIP .....	12

Why Do I Need an Extra IP Address? . . . . .	12
Connectivity Options . . . . .	12
IUCV . . . . .	12
vCTC . . . . .	12
Guest LAN . . . . .	12
IP Addressing . . . . .	12
Connecting the SMTPPLUS Processing Engine to VM TCPIP . . . . .	12
IUCV . . . . .	13
CTC . . . . .	13
Guest LAN . . . . .	13
Using the Application Proxy . . . . .	14
Why Have a Proxy? . . . . .	14
Connecting the Proxy to VM TCPIP . . . . .	14
Directory changes to VM SMTP . . . . .	14

---

## Installing SMTPPLUS . . . . . 15

### Installing SMTPPLUS . . . . . 16

#### Creating the SMTPPLUS user ID . . . . . 17

Mounting the Install Tape . . . . . 17

Loading the Directory Entries . . . . . 17

Attach the tape to the SMTPPLUS Userid . . . . . 19

#### Loading the Installation Tools . . . . . 20

Preparing the 191-disk . . . . . 20

Loading the Installation Tools . . . . . 20

#### Running INSTALL EXEC . . . . . 21

Example run of INSTALL EXEC . . . . . 21

#### Post-Installation Tasks . . . . . 23

IPL the Processing Engine and Run One-time Tasks . . . . . 23

Changing the Root Password . . . . . 23

Halting the Processing Engine . . . . . 23

---

## Configuring SMTPPLUS . . . . . 25

### Configuring the SMTPPLUS Processing Engine . . . . . 26

Editing the SMTPPLUS customization files . . . . . 26

Overview of SMTPPLUS Configuration Files . . . . . 26

### Configuring PROFILE EXEC . . . . . 29

### Configuring BOOT PARAMS . . . . . 31

#### Configuring Mail Processing Parameters in EXIM4 CONF . . . . . 35

Delivering mail in BSMTP format . . . . . 35

Delivering Outbound Mail . . . . . 35

    Delivering in MX Mode . . . . . 35

Delivering in Proxy Mode . . . . . 36

Configuring Host Masquerading . . . . . 36

Configuring Host Access . . . . .	37
Configuring Host Access for SMTP transactions . . . . .	37
Configuring Host Access for Relaying . . . . .	37
Accepting Mail for Multiple Domains . . . . .	38
Configuring SMTP TLS Authentication . . . . .	38
Example: Plaintext SMTP Authentication . . . . .	38
<b>Configuring Alias Resolution in MAPUSER PL . . . . .</b>	<b>39</b>
Default behavior . . . . .	39
MAILER support . . . . .	42
LDAP support . . . . .	42
<b>Configuring Spam and Virus scanning in AMAVISD CONF . . . . .</b>	<b>43</b>
Configuring Virus Scanning in AMAVISD CONF . . . . .	43
Configuring Virus Notification Mail . . . . .	43
Configuring Default Action when virus Detected . . . . .	43
Configuring Retention of Quarantined Files . . . . .	44
Configuring Automated Virus Pattern File Updates . . . . .	44
Configuring Spam Abatement Parameters in AMAVISD CONF . . . . .	44
Spam Scoring . . . . .	44
Configuring Default Action when Spam Detected . . . . .	45
Warning the sender when spam is received . . . . .	45
Configuring Spam Tagging and Rejection Levels . . . . .	45
Configuring Retention of Spam Messages . . . . .	46
Example of Retaining Spam Messages . . . . .	46
Configuring Automated Removal of Certain Attachment Types . . . . .	47
SpamAssassin-specific parameters in LOCAL CF . . . . .	47
<b>Configuring the SMTPPLUS Application Proxy . . . . .</b>	<b>48</b>
Linking to the Application Proxy Code . . . . .	48
Configuring Proxy Settings . . . . .	48
Setting Up Port Proxying . . . . .	49
Static Port Proxying . . . . .	49
Transient Port Proxying . . . . .	49
Restricting Access to the Outbound Proxy . . . . .	50
IPMAPPER Startup . . . . .	50
<hr/>	
<b>Testing SMTPPLUS . . . . .</b>	<b>53</b>
<b>Testing SMTPPLUS . . . . .</b>	<b>54</b>
<b>Testing Mail Delivery to VM Users . . . . .</b>	<b>55</b>
Testing Mail Delivery to VM Users via SMTP . . . . .	55
Making a Telnet Connection to the SMTPPLUS Server . . . . .	55
Connecting in MX Mode. . . . .	55
Connecting in Proxy Mode. . . . .	55
Creating an Example Mail Transaction . . . . .	55
Verifying Delivery via Log Files . . . . .	56
Testing Mail Delivery from VM users via BSMTP Punch Files . . . . .	57
Creating a Sample BSMTP Punch File . . . . .	57
Redirecting the Virtual Punch . . . . .	57
Redirecting the Virtual Punch in MX Mode . . . . .	57
Redirecting the Virtual Punch for Proxy Mode . . . . .	57

Sending the message . . . . .	58
Resetting the virtual punch . . . . .	58
Verifying Delivery via Log Files . . . . .	58
Checking the external account . . . . .	58
<hr/>	
<b>Putting SMTPPLUS Into Production . . . . .</b>	<b>59</b>
<b>Putting SMTPPLUS Into Production . . . . .</b>	<b>60</b>
Common Tasks (MX and Proxy Modes) . . . . .	60
Modifying the Production VM TCPIP Configuration . . . . .	60
Changing the SMTPSERVERID Entry in TCPIP DATA . . . . .	60
<b>MX-Mode Specific Steps . . . . .</b>	<b>61</b>
Add MX records to VM DNS Entry . . . . .	61
Users Change SMTP Server Settings . . . . .	61
Mailer Configuration Reflects New SMTP Servers . . . . .	61
Disabling the VM TCPIP SMTP Server . . . . .	61
<b>Proxy Mode Specific Steps . . . . .</b>	<b>63</b>
Updating SMTP DTCPARMS to Start the Application Proxy . . . . .	63
Editing PROFILE EXEC to Start the Application Proxy . . . . .	63
Restart the SMTP Virtual Machine . . . . .	63
<hr/>	
<b>Maintenance Notes . . . . .</b>	<b>65</b>
<b>Maintenance Notes . . . . .</b>	<b>66</b>
Log Files . . . . .	66
Quarantine Directories . . . . .	66
Checking the Mail Queue . . . . .	66
<hr/>	
<b>Appendices . . . . .</b>	<b>67</b>
<b>Appendix A. Tape Layout . . . . .</b>	<b>68</b>
<b>Appendix B. INSTALL Command Syntax . . . . .</b>	<b>69</b>
Purpose . . . . .	69
Format . . . . .	69
Parameters . . . . .	69
Options . . . . .	69
Usage . . . . .	70

# Figures

1.	Data Flow Through SMTPPLUS	4
2.	Additions to PROFILE TCPIP for IUCV connection	13
3.	Additions to PROFILE TCPIP for CTC connection	13
4.	Additions to PROFILE TCPIP for Guest LAN connection	13
5.	Change to SMTP directory entry to support IPMAPPER	14
6.	Attaching the tape to a Privileged user.	17
7.	Loading the SMTPPLUS Directory Entry template	17
8.	Sample Directory Entry for SMTPPLUS	18
9.	Attaching the tape to SMTPPLUS	19
10.	Formatting the SMTPPLUS 191 disk	20
11.	Loading the SMTPPLUS installer	20
12.	Running INSTALL to install SMTPPLUS	22
13.	IPLing SMTPPLUS	23
14.	Changing the processing engine root password	23
15.	Halting the processing engine	23
16.	PROFILE EXEC for SMTPPLUS	29
17.	BOOT PARAMS for SMTPPLUS	31
18.	Delivering mail in BSMTP format	35
19.	Exim remote_smtp for MX Mode	35
20.	Exim remote_smtp for Proxy Mode	36
21.	Setting the hostname and domain for SMTPPLUS.	37
22.	Setting a host ACL for SMTPPLUS	37
23.	Setting a relay ACL for SMTPPLUS	38
24.	Accepting mail for multiple domains	38
25.	Establishing a plaintext SMTP authenticator	38
26.	MAPUSER PL for SMTPPLUS	40
27.	Warn sender when virus detected	43
28.	Warn recipient when virus detected	43
29.	Warn administrator when virus detected	43
30.	Turning off virus quarantine	44
31.	Automated daily quarantine purging	44
32.	Disabling Virus Database updates	44
33.	Warning a spam sender	45
34.	Default Spam Tagging and Rejection Configuration	45
35.	A more aggressive Spam Abatement Policy	46
36.	Changing spam subject rewriting policy	46
37.	Setting a spam quarantine recipient	46
38.	Adding banned file extensions	47
39.	IPMAPPER Utility Exec	48
40.	IPMAPPER static port mapping	49
41.	IPMAPPER free port set	49
42.	IPMAPPER IP authentication map	50
43.	Modifying IPMAPPER's PROFILE EXEC	50
44.	Starting IPMAPPER within SMTP During Testing	51
45.	Using "tail" in filter mode	54
46.	SMTP Banner	55
47.	SMTP Transaction Example Dialogue	56
48.	Log file for inbound SMTP transaction	56
49.	Amavis-added X-Virus-Scanned: header	57
50.	BSMTP message to test outbound SMTP	57

51.	Directing the virtual punch for MX Mode	57
52.	Directing the virtual punch for Proxy Mode	57
53.	Sending BSMTP via the punch	58
54.	Resetting the virtual punch	58
55.	Log file for outbound SMTP transaction	58
56.	Changes to TCPIP DATA for SMTPPLUS	60
57.	DNS change for MX mode	61
58.	Removing SMTP from TCPIP's control	62
59.	Logging off the SMTP virtual machine	62
60.	SMTP DTCPARMS for IPMAPPER invocation	63

# **Tables**

1. CMS and Linux configuration files . . . . .	27
2. Default Spam Tagging and Rejection Configuration . . . . .	45
3. More Aggressive Spam Tagging and Rejection Configuration . . . . .	46



---

# Introduction

## Introduction

SMTPLUS is a product designed to provide a replacement for the VM SMTP machine included as part of TCP/IP. In addition to performing the functions of VM SMTP, it offers higher performance, as well as integrated spam and virus scanning.

## SMTPLUS Features

SMTPLUS acts as an SMTP transport, listening for SMTP mail on TCP port 25, and delivering that mail either over TCP/IP to other hosts, or directly to the virtual reader of a VM user. This is also the function of the VM SMTP machine, which it replaces. However, SMTPLUS is also capable of performing virus and spam scanning, and performing policy-based routing or rejection of messages determined to be spam or to contain viruses. The SMTPLUS SMTP Mail Transfer Agent is significantly more configurable than VM SMTP. Full IPv6 support is included.

### Performance Note

Performance is significantly better than the base VM SMTP product, when functioning as a pure Mail Transport Agent. Spam and virus scanning add significant CPU and I/O load to the system.

If you need a VM MTA that performs better than your hardware can deliver with scanning enabled, consider turning off spam and virus scanning on SMTPLUS and performing those tasks outboard.

End of Performance Note

## High-Performance Mail Processing

SMTPLUS uses the powerful and popular Exim v.4 as the core of its Mail Transfer Agent. It is extremely customizable and is in wide use at installations of all sizes. Exim is especially popular at sites with very high mail volumes.

## Automated Virus Scanning

SMTPLUS uses the Amavis-NG framework as the infrastructure for both its spam and its virus detection. Mail is passed to Amavis, which in turn feeds it through a set of filters to determine the ultimate fate of the messages. In the default configuration, both incoming and outgoing mail is scanned.

The default configuration uses the Clam Antivirus Open-Source AV scanner; however, Amavis is easily configured to use other scanners, as each site prefers. In the default configuration, Clam AV updates its list of virus signatures nightly. This feature requires configuring the SMTPLUS machine with a globally routable IP address, or positioning it behind an HTTP proxy.

## Automated Spam Detection, Tagging and Rejection

SpamAssassin is configured as the spam scanner. In its default configuration, it takes a fairly conservative view of what it considers spam; only messages with an exceptionally high spam score will actually be discarded (with a negligible false-positive rate; in about a year of using SpamAssassin, the author has never seen a legitimate mail with a spam score above 10.0, which is the default discard threshold). Messages deemed to be spam, but not so offensively so as to be discarded unread, are tagged as spam before delivery, and thus can be easily filtered by the end- user's mail tool.

SpamAssassin is configured to allow Bayesian learning capabilities, but it is the responsibility of the site administrator to sort through the spam and ham folders and recategorize anything in the wrong folder, so that the Bayes algorithms actually learn to discriminate better.

## Automated Removal of Hostile Payloads

The virus scanner, by default, will quarantine any message it recognizes as infected, for further perusal by the system administrator. This can be configured to simply discard the message, to remove the infected content (if possible), and to warn either the sender or the recipient that the message was infected. It will also scan for, and remove, message parts with specified extensions (e.g. .exe and .com).

## Compatibility with VM SMTP Spool and BSMTP MAILER Interfaces

SMTPLUS is fully compatible with the VM SMTP spool and mailer interfaces. Messages may be sent to SMTPLUS's reader in any of three formats: a plaintext (EBCDIC) RFC-2822 mail message, a NETDATA-encoded RFC-2822 encoded message, or a BSMTP transaction. All of these will be decoded into ASCII and fed to the MTA.

Messages for local—i.e. VM punch—delivery are encoded in NETDATA form and punched directly to users' readers. The delivered mail may be in RFC-822 or in BSMTP format, at the site administrator's option. The SMSG interface to the VM SMTP server is not supported although Linux equivalents exist to, for example, monitor queue state.

## Processing Engine

The Processing Engine is the Linux guest (usually SMTPLUS). It runs Exim and Amavis, and Amavis in turn invokes SpamAssassin and Clam Antivirus. These functions are provided by Open Source software.

Inbound and outbound spool integration is provided by Sine Nomine-written custom code. This software is not Open Source, and may not be redistributed.

## Application Proxy

The application proxy is a CMS RSK-based utility known as IPMAPPER. It provides bidirectional proxying from the VM stack to the processing engine. In the shipped configuration, it proxies any traffic to port 25 to the application engine's SMTP port, and allows the applicatin engine to request that a local port be mapped via the proxy to an arbitrary host/port combination.

## Data Flow Through SMTPPLUS

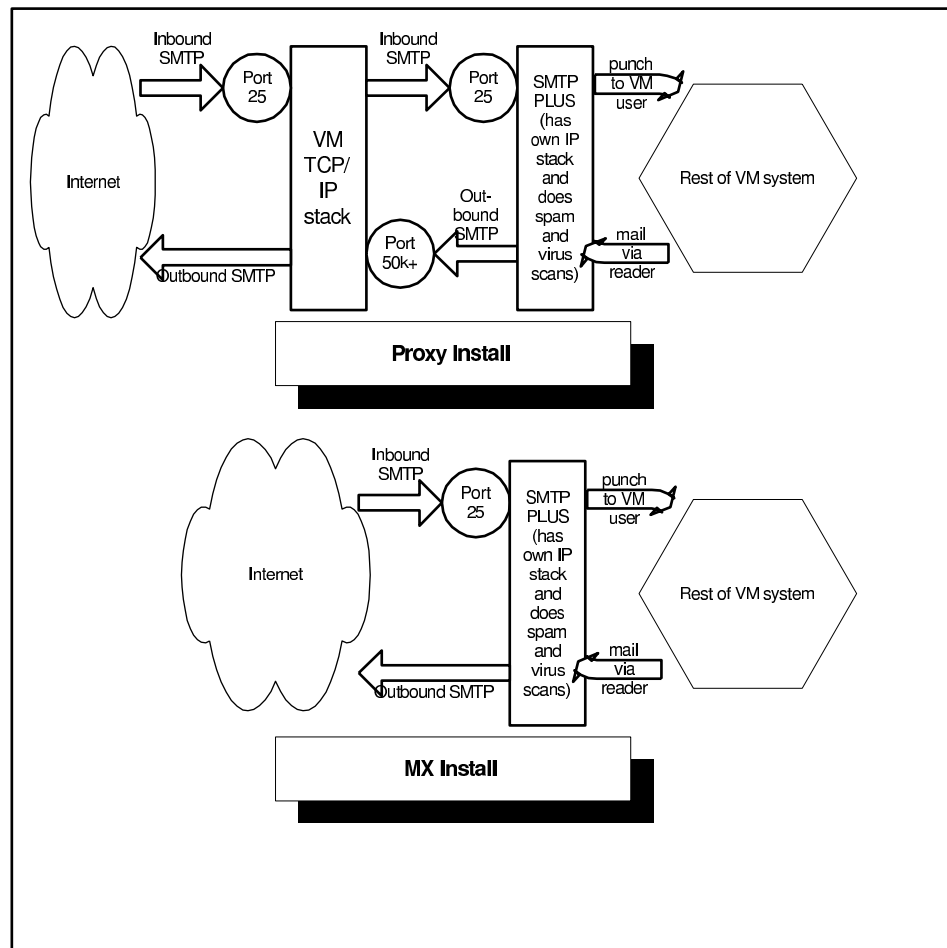


Figure 1. Data Flow Through SMTPPLUS

## Integration Points

There are several opportunities for user modification of SMTPPLUS's functionality.

- Spam and virus mitigation policies are completely user-configurable.
- Mapping of e-mail addresses to local VM users is under the site administrator's control.
- The frequency with which the virus signature database is updated is user-configurable.
- All aspects of mail routing can be set as site policy:
  - Recipient mapping.
  - Inbound and outbound relaying decisions.
  - Host access decisions.
  - User authentication.

---

# Preparing Your System for SMTPPLUS Installation

This section will guide you in determining the requirements for an installation of SMTPPLUS.

---

## System Requirements

### Hardware

At least a G3 System/390 or zSeries processor with the Halfword Immediate instruction set is required; a G5, G6 or zSeries system is strongly recommended. Supported models include:

- P/390E
- R/390
- Integrated Server
- Multiprise 2000 and 3000
- All 9672-G3, -G4, -G5, and -G6 turbo and non-turbo models
- All z/800, z/900, and z/990 models.

### Software

Any currently supported release of VM/ESA or z/VM is supported:

- VM/ESA 2.4
- z/VM 3.x
- z/VM 4.x

### Virtual Machine Storage Resources

The Processing Engine requires a 128 MB virtual machine with 300MB (600000 512-byte blocks) of VDISK space.

If used, the Application Proxy requires a 64MB virtual machine running in XC mode.

### Connectivity Resources

In most configurations using the application proxy or MX-style install, external connectivity to the VM system will still be under VM's control and therefore will not need to change from the current configuration.

Because SMTPPLUS is a Linux virtual machine, communication between it and the VM TCP/IP stack takes place via IP. This requires that the VM TCP/IP network configuration be modified to provide connectivity to and from the SMTPPLUS guest; this connectivity may be vCTC, IUCV, or Guest LAN.

Examples of the necessary stack modifications may be found at Figure 2 on page 13, Figure 3 on page 13, and Figure 4 on page 13.

If external connectivity to the SMTPPLUS guest is required (e.g. in an MX-mode installation) then network routing changes may be necessary to make the SMTPPLUS machine accessible to the rest of the network.

---

## DASD Requirements

SMTPPLUS requires 1300 cylinders of 3390 DASD. See the directory entry Figure 8 on page 18 for the breakdown.

---

# SMTPLUS Install Planning

---

---

## The SMTPLUS Install Worksheet

You should have completed an installation worksheet when placing your order for the SMTPLUS product. The tape set you received is customized with the choices you made there. These choices are reflected in the user-customizable files on the SMTPLUS 191-disk.

Future releases will include an interactive installation worksheet which directly generates the appropriate configuration files.

---

## SMTPLUS Install Type

There are two types of installation: the proxy installation, where the SMTPLUS SMTP mailer appears to reside on the VM TCP/IP stack, and the MX installation, where SMTPLUS has its own distinct IP address seen by the applications that interact with it.

### Proxy Install

The proxy installation is the most common type of installation. In this mode, SMTP mail transactions continue to come to the VM TCP/IP stack, from which the connections are forwarded to the SMTPLUS machine by the application proxy. Outbound mail traffic from SMTPLUS is transmitted by the proxy server through the VM stack as well. Any files in the SMTP userid's spool are periodically transferred to the SMTPLUS virtual reader spool, so the SMTPLUS machine can function as a completely transparent (from the end-user's perspective) replacement for SMTP.

The Processing Engine and the Application Proxy are used in tandem to present the illusion of being operated from the VM TCPIP stack address. Neither end-users on your VM system nor users out in the Internet at large should need to reconfigure anything to use the system in proxy mode.

Note that it is possible to proxy inbound but not outbound connections or vice versa. The two functions are independent. A possible use for this would be the following: you don't need to proxy outbound connections—a connection coming from SMTPLUS is fine—but some users have the VM stack's IP address hardcoded as their outbound SMTP relay. In this case, you might want to forward inbound port 25 to SMTPLUS, but omit outgoing proxying via the VM stack.

### MX-Style Install

The MX-style install omits the proxy features. This requires that the processing engine itself be on a publically-routeable network; mail goes directly to the processing engine without being proxied through the VM stack, and outbound mail comes directly from the processing engine. Spool files arriving at the SMTP userid may still be transferred to the SMTPLUS machine for mail delivery.

This is slightly higher-performance than the proxy install, because there is no need to run an application proxy under VM. However, it will require cooperation from the site's routing and DNS maintenance staff.

SMTPPLUS, when installed in MX mode, does not install the Application Proxy.

In this mode, TCP connections delivering mail from your users will no longer appear to come from the VM stack, and the MX record specifying which machine accepts mail for VM will change.

---

## Mail Processing Policy Decisions

The SMTPPLUS package has been designed to allow all site-specific configuration decisions to be made by editing one of seven files on the SMTPPLUS 191 minidisk. These seven files are PROFILE EXEC, BOOT PARAMS, RESOLV CONF, EXIM4 CONF, AMAVISED CONF, LOCAL CF, and MAPUSER PL.

Each of these files controls a different aspect of the operation of the mail server. These files must reside on a minidisk, rather than in SFS, because Linux cannot directly read files in SFS space, and Linux requires the contents of these files (excepting PROFILE EXEC) at IPL time.

---

### WARNING

---

These files are placed on the A-Disk (191) so that you can edit them from a CMS environment. However, once they're read into Linux at IPL time, they are not written back to the A-disk. What this means is that you should decide whether you will maintain the files in Linux or in CMS, and stick to that decision. The newer file will always win at Linux IPL. If you intend to use both environments to edit the files, it is your responsibility to propagate the changes you make while in Linux back to CMS.

Changes to the mail system parameters take effect upon Linux IPL. Changes to network routing (i.e. changes to BOOT PARAMS) require an additional Linux IPL before they take effect.

---

End of WARNING

---

## Host Masquerading

Host masquerading allows the SMTPPLUS machine to rewrite incoming and outgoing messages to hide or "masquerade" the true origin of the message within your domain. The SMTPPLUS processing engine is capable of rewriting some or all headers and message envelope fields to reflect a site-specific string.

SMTPPLUS uses this capability to allow it to work transparently, rewriting the messages that pass through it to appear it is operating on the VM stack, rather than on the separate SMTPPLUS TCP/IP stack. You may also use this feature to rewrite outgoing mail addresses to reflect a site-wide unqualified domain name (such as foo.com) rather than the specific host name transmitting the message.

## Host Access Decisions

You may want to limit SMTPPLUS to communicating with a specific list of IP hosts. This, for example, is useful for a mailserver that only serves a single department but relies on an external "smarthost" to handle SMTP for destinations outside the organization. This capability can be specified in several ways.

## IP Address Access Lists

You can define IPv4 and IPv6 address ranges for machines which are permitted to use the SMTP server. This is done with an access control list as seen in Figure 22 on page 37.

Proxy Mode Note

This technique does not work in proxy mode, because all hosts connect to the VM stack first, and all connections that SMTPPLUS sees actually come from VM rather than from the originating host. Future releases of the application proxy will include a access list capability to block certain IPv4 address ranges at connect time.

End of Proxy Mode Note

## Host Relaying and Authentication Decisions

SMTPPLUS is initially configured not to allow SMTP relaying at all; it accepts SMTP mail only for itself. Mail coming in via the spool interface is treated as local and is permitted to be sent anywhere.

Proxy Mode Note

If running SMTPINST in proxy mode is necessary, but you also need it to relay for users on trusted networks, you have two options:

- Disable the VM-based proxy, give SMTPPLUS control of the physical network adapter, and use iptables to forward ports on which VM listens to the VM machine, while masquerading outbound connections from VM to come from the SMTPPLUS stack, which will be set up with the VM system's previous IP address.
- Configure TLS authentication and allow only authenticated hosts to relay traffic.

If you require this configuration, please contact Sine Nomine Associates for consulting assistance.

End of Proxy Mode Note

## Hosts Permitted to Use SMTPPLUS as a Relay

By default, no hosts other than the local host can use SMTPPLUS as a relay. This can be changed; see Figure 23 on page 38 if running in MX-mode. Doing this in proxy mode without configuring TLS authentication will, as noted above, turn your system into an open relay.

## SMTP TLS Authentication

Because certificate policies vary so much by site, it is not practical to attempt a "one-size-fits-all" TLS policy. Information on SMTP authentication can be found in chapters 33-37 of the Exim specification at <http://www.exim.org/exim-html-4.30/doc/html/spec.html>. This is the recommended method for configuring relaying, although you should be aware that this approach does require configuration changes on the remote systems attempting to relay mail via your server.

If you require assistance with TLS authentication for SMTPPLUS, please contact Sine Nomine Associates for consulting assistance.

## Alias and Delivery Policy

SMTPPLUS attempts to map the email address of the message envelope onto a VM userid by calling a set of Perl functions to determine the final disposition of the message once it is complete and ready to commit to VM spool space. Once it has done this, it encodes the file that it received as NETDATA, using codepage 1047 ASCII to EBCDIC translation, and sends that file to the appropriate userid's virtual reader. The file is tagged to have come via SMTP so that the CMS RDRLIST command will display it as having come from the SMTP sender identified in TCPIP PROFILE.

Mail may be delivered in either RFC-822 or BSMTP format, at the site administrator's option.

## Use of a MAILER

SMTPPLUS fully supports the use of a mail routing server, or "mailer" virtual machine. Mailer support can be implemented by using the procedure described under "MAILER support" on page 42 for further details.

## Use of LDAP Map

Because the user mapping function is implemented as a Perl script, it is easy to do an LDAP lookup for a particular user and to return the VM user for whom to punch the message.

## Use of Local and LDAP-Resolved Aliases

Again, because user mapping is done with Perl, one can mix and match these approaches, and consult a local alias map first, and if that fails to return a match, try an LDAP query.

---

## Virus Scanning Policy Decisions

The amavisd-new framework is used to support "plugging in" any of several virus scanners. By default, Clam Antivirus is used to scan mail and match its content against a database of virus signatures.

The default amavisd configuration will recognize most common virus scanners when they are installed, so Clam Antivirus can be augmented or replaced with a different scanning product.

## Default Action when Virus Detected

Viruses are, by default, put into a quarantine area within the processing engine where they may be examined and (optionally) purged by a system administrator.

## Retention of Infected Files

Infected files are aged out automatically after a week without being accessed. This retention period, and the decision to retain files at all, are site-configurable policy choices.

## Automated Removal of Certain Attachment Types

Amavis supports discarding files based solely on extension. If site policy states that users are not to use email to transmit executables and scripts, this can be very effective in removing viruses in transit. However, if your users are in the habit of mailing each other executable files, this will cause them to lose mail.

## Enable Automated Virus Pattern File Updates

Freshclam by default checks 5 times a day for new virus patterns. This is a site-configurable policy.

---

## Spam Abatement Policy Decisions

Amavis also controls the spam scanner; just as, in its default configuration, Amavis relies on Clam Antivirus to perform actual virus scanning, Amavis relies on SpamAssassin to perform spam scanning.

SpamAssassin's method of operation is to examine a mail message, and raise or lower its scores based on spam assessment heuristics. For instance, an email whose title includes the word "VIAGRA" in capitals, and whose body consists solely of an HTML link to another site, is likely to receive a high score. After scoring a message, SpamAssassin may rewrite the headers of that message; at that point, control is returned to Amavis, which decides how to route the message based on its spam score.

## Spam Tagging and Rejection Levels

SpamAssassin decides whether a message is spam or not based on a score it assigns from its heuristics. Certain phrases or message attributes (e.g. the presence of the word "Viagra", subjects all in capitals, or a message body consisting solely of a URL) are assigned a positive score; others (e.g. we have seen non-spam mail from the sender before) are assigned a negative score. After the whole message is scanned these values are summed to produce a total score for the message.

Amavis defines three thresholds: at the bottom threshold, tags showing the message's score are added. At the middle one, the message is considered very likely to be spam, and its subject is rewritten. At the top threshold, the message is rejected completely.

## Default Action when Spam Detected

By default, SMTPPLUS tags any message with a spam score greater than zero. Messages receiving a score of 5.0 or higher receive a tag and the subject is rewritten to include "[\*\*\*SPAM\*\*\*]" The original message is repackaged as a RFC-822 message attachment to prevent overenthusiastic mail clients from automatically invoking attached executable content (spam and viruses often go together). Messages scoring 10.0 or above are immediately rejected and are not delivered to the intended user. Legitimate messages with a spam score above 6.0 are vanishingly uncommon.

These thresholds are site-configurable.

## Retention of Spam Messages

Based on a message's spam score, it can be routed into different folders—system-wide or per-user—or isolated in a quarantine directory. These policies are not implemented by default, but the configuration settings to implement them are accessible in the configuration file; see Figure 37 on page 46.

---

## Connecting the SMTPPLUS Processing Engine to VM TCPIP

SMTPPLUS runs inside its own virtual machine, with its own IP stack. To transmit messages from the VM stack to the SMTPPLUS guest requires that the message be delivered to the SMTP port on the Linux guest.

### Why Do I Need an Extra IP Address?

Because SMTPPLUS is actually a Linux guest, it does not use the VM TCP/IP stack, but relies on its own TCP/IP services. Hence it must have an address separate from that of VM.

### Connectivity Options

IUCV, vCTC, and a Guest LAN connection are all supported. For a machine in Proxy mode, the point-to-point solutions are slightly more elegant, but in all cases, Guest LAN is much easier to configure, and we recommend it if you have z/VM 4.2 (with recent service installed) or later available.

#### IUCV

An IUCV point-to-point connection can be used; SMTPPLUS gets its own IP address at one end of the IUCV connection. The other end is the VM TCPIP stack, usually TCPIP. It is not necessary that the VM end of the IUCV connection have its own IP address but it makes debugging easier.

Depending on local network policy, it may be advisable to turn on PROXYARP in the VM stack; in general, it is better to simply advertise a route to SMTPPLUS via the VM stack, but this may not be feasible in all cases.

#### vCTC

The same advice applies to vCTC links; however, the virtual CTC for both the VM TCP/IP stack and for SMTPPLUS must be defined in the directory entry, and the two must be COUPLED together. This is usually done in SMTPPLUS's PROFILE EXEC

#### Guest LAN

The Guest LAN is by far the easiest way to employ SMTPPLUS. Once an appropriate (usually SYSTEM-owned, unrestricted) LAN has been constructed (and VM TCPIP knows how to route for that LAN) then SMTPPLUS can create and couple its NIC devices in PROFILE EXEC (this can be done in the directory entry as well, if preferred).

### IP Addressing

SMTPPLUS needs a single unique IP address; if it is going to communicate with the outside world, this must be a public address or it must be a private address behind a network device that performs the necessary Network Address Translation for packets to and from SMTPPLUS.

## Connecting the SMTPPLUS Processing Engine to VM TCPIP

A route must exist from VM to SMTPPLUS. In the following example, let us assume that the IP address of the interface on the VM TCPIP stack connecting to SMTPPLUS is 10.0.2.1 and that it will be our gateway to the outside world. The SMTPPLUS machine will have an IP address of 10.0.2.2. In the Guest LAN case the netmask will be 255.255.255.224. We assume that the network outside the VM system is configured to route 10.0.2.0/24 via the VM TCPIP stack, and therefore we will not need to configure VM TCPIP to proxy ARP for our guests.

Throughout these examples, we assume that SMTPPLUS is the name of the virtual machine running the SMTPPLUS service.

## IUCV

To enable IUCV connectivity for the SMTPPLUS processing engine:

Modify PROFILE TCPIP for the VM TCPIP stack used to connect SMTPPLUS with the following

---

```

DEVICE IUCV0 IUCV 0 0 SMTPPLUS A
LINK IUCV0 IUCV 0 IUCV0
HOME
  10.0.2.1      IUCV0
GATEWAY
  10.0.2.2      =                IUCV0    1500    HOST
START IUCV0

```

---

Figure 2. Additions to PROFILE TCPIP for IUCV connection

## CTC

In this example, we assume that the CTC pair used to connect to the SMTPPLUS engine is at virtual address 340-341. To enable CTC connectivity:

Modify PROFILE TCPIP for the VM TCPIP stack connecting to SMTPPLUS with the following:

---

```

DEVICE CTC0 CTC 340
LINK CTC0 CTC 0 CTC0
HOME
  10.0.2.1      CTC0
GATEWAY
  10.0.2.2      =                CTC0    1500    HOST
START CTC0

```

---

Figure 3. Additions to PROFILE TCPIP for CTC connection

## Guest LAN

This example demonstrates the use of a virtual OSA adapter coupled to Guest LAN LNXLAN1 at virtual address 0x340. To use guest LAN connectivity:

---

```

DEVICE QDI00 QDIOETHERNET 0340 PORTNAME LNXLAN1
LINK QDI00 QDIOIP QDI00
HOME
  10.0.2.1      QDI00
GATEWAY
  10.0.2.2      =                QDI00    0.0.0.0    0.255.255.224
START QDI00

```

---

Figure 4. Additions to PROFILE TCPIP for Guest LAN connection

This is the recommended method of connecting the SMTPPLUS engine for z/VM 4.2 and higher systems.

---

## Using the Application Proxy

Although the SMTPPLUS engine does not use the VM TCP/IP stack directly, to maintain the illusion that SMTPPLUS is providing the service from the VM TCPIP stack rather than as a separate machine, we use an application proxy to forward connections arriving on the SMTP port of the VM stack to the SMTP port of SMTPPLUS. We also use the proxy in an outbound mode to deliver messages to remote destinations so that the connection looks as if it came directly from the VM stack.

### Why Have a Proxy?

If it is not possible, for technical or political reasons, to modify the DNS entry for your VM system to include a MX record for the SMTPPLUS processing engine IP address or to obtain an additional publically routed IP address for the SMTPPLUS engine at your site, you can pretend that the same address is handling mail as it always has; this also means that, if users connect to the SMTP port of your server, that you can do this without reconfiguring their clients (although please note caveats about SMTP relaying).

### Connecting the Proxy to VM TCPIP

The application proxy runs in the old VM SMTP userid to minimize configuration changes in the VM TCPIP stack. SMTP must therefore run the application proxy's code rather than the native VM SMTP server. To do this in a testing environment, follow the instructions in Figure 44 on page 51, and to do this for production, modify SMTP DTCPARMS as described in Figure 60 on page 63.

### Directory changes to VM SMTP

To allow the application proxy to track port reservations and connection mapping reliably, the application proxy virtual machine (usually SMTP) must have a write link to SMTPPLUS's F191 disk. The default PROFILE EXEC for SMTPPLUS detaches F191 as soon as SMTPPLUS starts; an MR link to SMTPPLUS F191 in the directory entry for the SMTP userid is sufficient. The IUCV message limit is explicitly set to 255, and we allow the application proxy to write application monitor and accounting records as necessary.

It is also necessary to define SMTP as an XC-mode machine with access to multiple data spaces. This is required because the application proxy is an CMS RSK application, and the RSK requires the use of data spaces.

Put the following in SMTP's directory entry:

---

```
IUCV ALLOW PRIORITY MSGLIMIT 255
MACHINE XC
OPTION QUICKDSP SVMSTAT MAXCONN 255 SVM APPLMON ACCT
XCONFIG ADDRSPACE MAXNUMBER 256 TOTSIZE 8192G SHARE
LINK SMTPTST F191 F191 MR
```

---

Figure 5. Change to SMTP directory entry to support IPMAPPER

---

# Installing SMTPPLUS

---

# Installing SMTPPLUS

SMTPPLUS is installed via a four-step process.

- Step 1. As a privileged user, load the SMTPPLUS directory template from tape and create the SMTPPLUS user ID. Attach the tape drive to it.
- Step 2. As SMTPPLUS, load the installation tools.
- Step 3. As SMTPPLUS, run `INSTALL EXEC`.
- Step 4. As SMTPPLUS, perform several post-installation tasks, such as IPLing the processing engine, changing the default root password, and halting the virtual machine for configuration.

---

## Creating the SMTPPLUS user ID

---

---

### Mounting the Install Tape

From a privileged user ID, attach the tape:

---

```
ATTACH real-tape-device-address * 181
```

---

Figure 6. Attaching the tape to a Privileged user.

---

### Loading the Directory Entries

Use your local directory maintenance procedures to create the SMTPPLUS userid. The following template can be pulled from the tape with

---

```
TAPE LOAD SMTPPDIR EXAMPLE A ( 181
```

---

Figure 7. Loading the SMTPPLUS Directory Entry template

Use this template to build the SMTPPLUS user, modifying the disk and network device definitions as necessary. Employ the directory maintenance procedures in use at your site.

---

```

USER SMTPPLUS PASSWORD 128M 256M G
* For IUCV,
* IUCV ANY (or lock it down to TCPIP if you prefer)
*
* For CTC,
* SPECIAL 340 CTCA
* SPECIAL 341 CTCA
* (Couple in PROFILE EXEC)
*
* For Guest LAN, define and couple in PROFILE EXEC, or
* SPECIAL 340 QDIO 3 SYSTEM LNXLAN1
  IPL CMS PARM AUTOOCR
* IOPRIORITY is z/VM 4.4 specific
* IOPRIORITY RELATIVE 100
  SHARE RELATIVE 100
  CONSOLE 0009 3215 T CONLOG
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK MAINT 0190 0190 RR
  LINK MAINT 019E 019E RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019F 019F RR
  CPU 0
  CPU 1
  MACHINE XA 2
* 0191 contains site-specific configuration files
  MDISK 0191 3390 1 5 CKD001
* 0150 is the Linux root filesystem
  MDISK 0150 3390 6 400 CKD001
* 0160-016B are the /var software RAID filesystem
  MDISK 0160 3390 1 50 CKD002
  MDISK 0161 3390 1 50 CKD003
  MDISK 0162 3390 1 50 CKD004
  MDISK 0163 3390 1 50 CKD005
  MDISK 0164 3390 1 50 CKD006
  MDISK 0165 3390 1 50 CKD007
  MDISK 0166 3390 1 50 CKD008
  MDISK 0167 3390 1 50 CKD009
  MDISK 0168 3390 1 50 CKD010
  MDISK 0169 3390 1 50 CKD011
  MDISK 016A 3390 1 50 CKD012
  MDISK 016B 3390 1 50 CKD013

```

---

Figure 8 (Part 1 of 2). Sample Directory Entry for SMTPPLUS

---

```
* 0170-017B are the /tmp software RAID filesystem
  MDISK 0170 3390 51 20 CKD002
  MDISK 0171 3390 51 20 CKD003
  MDISK 0172 3390 51 20 CKD004
  MDISK 0173 3390 51 20 CKD005
  MDISK 0174 3390 51 20 CKD006
  MDISK 0175 3390 51 20 CKD007
  MDISK 0176 3390 51 20 CKD008
  MDISK 0177 3390 51 20 CKD009
  MDISK 0178 3390 51 20 CKD010
  MDISK 0179 3390 51 20 CKD011
  MDISK 017A 3390 51 20 CKD012
  MDISK 017B 3390 51 20 CKD013
* 0192 contains Linux IPL decks, docs, and tools
  MDISK 0192 3390 406 50 CKD001
* F191 contains application proxy code
  MDISK F191 3390 456 5 CKD001
```

---

Figure 8 (Part 2 of 2). Sample Directory Entry for SMTPPLUS

---

## Attach the tape to the SMTPPLUS Userid

From another terminal, log in to the SMTPPLUS user. From the privileged userid, ATTACH the first installation tape to SMTPPLUS:

---

```
ATTACH real-tape-device-address TO SMTPPLUS 181
```

---

Figure 9. Attaching the tape to SMTPPLUS

Now you may log out of the privileged userid.

---

## Loading the Installation Tools

---

### Preparing the 191-disk

From SMTPPLUS, do the following:

---

```

FORMAT 191 A
DMSFOR603R FORMAT will erase all files on disk A(191). Do you wish
to continue?

Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
1x0191
DMSFOR733I Formatting disk A

DMSFOR732I 5 cylinders formatted on A(191)

```

---

Figure 10. Formatting the SMTPPLUS 191 disk

---

## Loading the Installation Tools

Assuming that the tape is attached at device 181 to the SMTPPLUS user, enter the following.

---

```

TAPE REWIND ( 181
TAPE LOAD * * A ( 181

```

---

Figure 11. Loading the SMTPPLUS installer

---

## Running INSTALL EXEC

Next you must run INSTALL to install the SMTPPLUS product onto SMTPPLUS's mini-disks.

---

### Example run of INSTALL EXEC

Here we will run INSTALL with no arguments and allow it to prompt for the address of the tape drive, which is virtual address 181.

Full command syntax for INSTALL is given in Appendix B, "INSTALL Command Syntax" on page 69.

---

**install**

\*\*\*\*\*

Device number of installation tape?

\*\*\*\*\*

**181**

Formatting DASD 0150...

Formatting DASD 0160...

Formatting DASD 0161...

Formatting DASD 0162...

Formatting DASD 0163...

Formatting DASD 0164...

Formatting DASD 0165...

Formatting DASD 0166...

Formatting DASD 0167...

Formatting DASD 0168...

Formatting DASD 0169...

Formatting DASD 016A...

Formatting DASD 016B...

Formatting DASD 0170...

Formatting DASD 0171...

Formatting DASD 0172...

Formatting DASD 0173...

Formatting DASD 0174...

Formatting DASD 0175...

Formatting DASD 0176...

Formatting DASD 0177...

Formatting DASD 0178...

Formatting DASD 0179...

Formatting DASD 017A...

Formatting DASD 017B...

Formatting DASD 0192...

Formatting DASD 0191...

Formatting DASD F191...

Loading CMS files to DASD...

Restoring 150-disk from tape 181...

Installation loaded successfully. Please examine the files on the SMTPTTEST A-disk (191), and customize them if necessary. When satisfied, IPL CMS to start SMTPTTEST.

---

Figure 12. Running INSTALL to install SMTPLUS

---

## Post-Installation Tasks

---



---

### IPL the Processing Engine and Run One-time Tasks

---

Before you do anything else, you should IPL Linux and allow the RAID filesystems to be built. This requires no input from you.

---

```
IPL CMS
```

---

Figure 13. IPLling SMTPPLUS

CMS will IPL and then will pause for thirty seconds before IPLling the Linux system. Wait for the timer to expire and the processing engine IPL to occur. When the processing engine IPLs, it will allow the initial configuration of the RAID volumes to proceed. with an initial password of **rootpass**.

---

### Changing the Root Password

---

The SMTPPLUS processing engine is shipped with a default password for the "root" administrative user of "rootpass" (no quotes).

Change the root password as shown:

---

```
smtplusplus:~#: passwd
Enter new UNIX password: new-password
Retype new UNIX password: new-password
passwd: password updated successfully
```

---

Figure 14. Changing the processing engine root password

---

#### Important Note

---

Please be certain to record your root password in a secure location. Many (if not all) administrative tasks require its use and there is no way to recover a lost root password.

---

End of Important Note

---



---

### Halting the Processing Engine

---

After completing the post-installation tasks shown above, you will need to shut down the processing engine to examine and modify the configuration using the instructions in the next section. To halt the processing engine:

---

```
smtplusplus:~#: halt
```

---

Figure 15. Halting the processing engine

After it shuts down, the processing engine will be logged off. Log back into it, and type a nonblank character and Enter to interrupt the IPL process.

Now you will configure the system by modifying the files on the SMTPPLUS 191 disk.

---

# Configuring SMTPPLUS

---

## Configuring the SMTPPLUS Processing Engine

SMTPPLUS customization is applied by editing files on SMTPPLUS's 191-disk; these files (with the exception of PROFILE EXEC and BOOT PARAMS) are also reflected in the Linux filesystem.

---

### Editing the SMTPPLUS customization files

These files can be edited in CMS via any text editor, such as XEDIT. You may wish to consider putting them under CMS UPDATE control, so that you have an easy way to back out any changes whose effect was harmful.

Note that these files use the Unix convention that a backslash as the last character on the line is a continuation character, and the next line is treated, not as a separate line, but as a continuation of the previous line.

These files may also be edited within Linux if you are more comfortable in that environment. Both vi and emacs are included on the SMTPPLUS image. However, if you choose to edit the files in Linux, you should erase them from the A-disk so that you do not accidentally change the timestamp on the files on the A-disk and then overwrite your (already customized) Linux files with them.

---

WARNING

---

These files are placed on the A-Disk (191) so that you can edit them from a CMS environment. However, once they're read into Linux at IPL time, they are not written back to the A-disk. What this means is that you should decide whether you will maintain the files in Linux or in CMS, and stick to that decision. The newer file will always win at Linux IPL. If you intend to use both environments to edit the files, it is your responsibility to propagate the changes you make while in Linux back to CMS.

---

End of WARNING

---

Changes to the mail system parameters take effect upon Linux IPL. Changes to network routing (i.e. changes to BOOT PARAMS) require an additional Linux IPL before they take effect.

---

### Overview of SMTPPLUS Configuration Files

The initial configuration files (reflecting your responses to the configuration worksheet you filled out when you ordered SMTPPLUS) are supplied on SMTPPLUS 191. These files are:

Table 1. CMS and Linux configuration files	
CMS file name	Linux file name
PROFILE EXEC A	n/a
BOOT PARAMS A	n/a
RESOLV CONF A	/etc/resolv.conf
EXIM4 CONF A	/etc/exim4/exim4.conf
AMAVISD CONF A	/etc/amavis/amavisd.conf
LOCAL CF A	/etc/spamassassin/local.cf
MAPUSER PL A	/usr/local/bin/mapuser.pl

- PROFILE EXEC is responsible for defining virtual NICs and coupling them to a guest LAN, or for coupling the Linux guest's CTC pair to a VM TCPIP machine. It defines what would be the physical layer of network connectivity, if the virtual machine were real hardware.
- BOOT PARAMS specifies the network parameters used to connect the Linux guest to VM (and, possibly, to the rest of the network). This is Layer 2 and 3, IP-level stuff, and requires a properly-configured Layer 1 (set up by PROFILE EXEC or the directory entry) in order to work.
- RESOLV CONF is simply the file defining the Domain Name Service search space and location of the nameservers. You should be able to leave this at its default, 127.0.0.1, to run a local caching-only nameserver in most cases. However, if you have a split-horizon DNS (where the outside world sees different information than hosts inside your network), this will not function; in this case, put your campus or departmental nameserver in this list rather than 127.0.0.1. The adventurous may wish to configure their nameserver as a caching nameserver for outside network addresses and a slave nameserver for internal addresses. This will certainly work, but is not supported in the default installation, and requires fairly intimate knowledge of BIND's operation. Typing "man 5 resolv.conf" on almost any Unix or Linux system will fully define the format of this file. Complete BIND documentation is available at <http://www.nominum.com/content/documend9arm.pdf>.
- EXIM4 CONF is the configuration file governing Exim, the Mail Transfer Agent used on the guest. Full Exim documentation is available online at <http://www.exim.org/exim-html-4.30/doc/html/spec.html>. The options you are most likely to need to modify are discussed below.
- AMAVISD CONF is the file specifying the behavior of the Amavis scanning framework. The SMTPPLUS package uses Amavis to drive both SpamAssassin and its virus scanner (ClamAV, in the default installation). It is extensively commented. The Amavis homepage is <http://www.amavis.org>.

The bulk of the customization for virus and spam scanning you will do is found in AMAVISD CONF. Amavis, and its control file, are written in Perl. While it is possible to adequately tune performance and behavior without knowledge of Perl (most of AMAVISD CONF is simply variable assignments, and assignation of a string to a variable is self-evident), to fully exploit its power requires some Perl knowledge (for instance: to assign an anonymous function to a variable, or a hash reference, is not clear if you don't know Perl.)

- LOCAL CF governs advanced options for controlling SpamAssassin's behavior. This file will probably not need editing, as the common SpamAssassin settings are handled in amavisd.conf. SpamAssassin documentation can be found at <http://www.spamassassin.org/doc.html>.
- MAPUSER PL is a set of Perl functions called when mail is received locally. The file contains documentation in the form of comments inside it; its basic function is to map an incoming email address to a VM user and to spool the punch to that user.

## Configuring PROFILE EXEC

PROFILE EXEC controls the definition and coupling of the network devices. It is here that you would want to define your NIC devices and couple them to a guest LAN, or couple your CTC pair to the VM stack. In this example we define an OSA-type QDIO interface at 340 coupled to guest LAN SYSTEM LNXLAN3.

---

```

/*****
/* FILE:          PROFILE EXEC                               */
/* RELEASE:       1.0.0                                     */
/* PROJECT:       VM SMTP REPLACEMENT                     */
/* DESCRIPTION:   PROFILE FOR LINUX PROCESSING ENGINE     */
/* LICENSE:      COPYRIGHT 2004 SINE NOMINE ASSOCIATES   */
*****/

/* Identify myself */

me = userid();

/* Detach the F191 disk, because SMTP needs a write link */
'CP DET F191'

say me': Creating network devices....'
/* Create/Couple network devices */
/* These must match the definitions you have in BOOT PARAMS */
/* Effectively, you're creating the Layer 1 ("physical") */
/* devices here, and you'll put Layer 2 on top of them in */
/* BOOT PARAMS */

/* No need to do this if devices and LAN are in directory */

'CP DEF NIC 340 TYPE QDIO'
'CP COUPLE 340 SYSTEM LNXLAN1'

/* If you had a CTC here you'd do something like: */
/* 'CP COUPLE 340 TCPIP 341' */
/* 'CP COUPLE 341 TCPIP 340' */

/* IUCV wouldn't need anything special */

/* Generate 3 swap-disks using VDISK */
say me': Creating swap disks in VDISK...'
'SWAPGEN 151 100000'
'SWAPGEN 152 200000'
'SWAPGEN 153 300000'

```

---

Figure 16 (Part 1 of 2). PROFILE EXEC for SMTPPLUS

---

```
/* Boot from DASD unless interrupted */

say me': Enter a non-blank character and ENTER (or two ENTERs)'
say me': within 30 to interrupt 'me' IPL.'

'WAKEUP +00:30 (CONS'

IF rc = 6 THEN DO
  say me ': Interrupt: entering CMS.'
  PULL /* Clear Stack */
  say me': IPL 150 CLEAR to restart 'me'.'
  say me': Type RESCUE to boot rescue system from reader'
  END
ELSE DO
  'CP IPL 150 CLEAR'
END
EXIT
```

---

Figure 16 (Part 2 of 2). PROFILE EXEC for SMTPLUS

## Configuring BOOT PARAMS

BOOT PARAMS contains network parameters used by the processing engine. You must include the type and IP address of your interface. Comments in the file indicate required parameters for each host type.

Note

Device addresses supplied here must match the entries in PROFILE EXEC or in the CP directory entry.

End of Note

```

*****/
** FILE:          BOOT PARAMS          */
** RELEASE:       1.0.0                */
** PROJECT:       VM SMTP REPLACEMENT  */
** DESCRIPTION:   BASIC BOOT PARAMETER FILE */
** LICENSE:       COPYRIGHT 2004 SINE NOMINE ASSOCIATES */
*****/
# Lines beginning with "#" are comments
#
# The fully-qualified hostname is required
# This uniquely identifies the SMTPPLUS processing engine
# It is a global parameter, not per-interface like the rest
#
hostname=smtplus.biz.example
#
#
# Interface definitions
#
# begin with interface.number=interfaceName
# then options are option.interfaceName=value
#
# The SMTPPLUS machine will generally only have one interface,
# therefore its name will generally end in "0".
# Your choices are "eth0", "hsi0", "iucv0", and "ctc0"
# OSA-type qeth devices and LCS ethernet will use "eth0"
# Hipersockets-type qeth devices use "hsi0"
# CTC devices use "ctc0"
# IUCV connections use "iucv0"

# type is mandatory: it may be one of
# qeth (for Hipersocket or OSA connections) (guest LANs use this)
# lcs (for OSA-1, or for the 3088 Ethernet presented by a P/390
#     or Integrated Server)
# ctc (for CTC connections) (pre-guest LAN VM systems may use vCTC)
# iucv (another favorite for pre-guest LAN VM systems)
#
# This must match the networking devices defined in PROFILE EXEC

```

Figure 17 (Part 1 of 4). BOOT PARAMS for SMTPPLUS

---

```
# ip is mandatory.  It is the IPv4 address of the interface.  We do
# not yet support DHCP to automatically set the IP address on
# IPL.

# address is the device starting address of the interface.  This is
# mandatory, except in the case of IUCV, which doesn't have a
# device address.
#
# This must match the definition in PROFILE EXEC as well

# portname is required for an OSA-type qeth device under z/VM
# prior to 4.4.  Usually it will be the same as the guest LAN name
# found in PROFILE EXEC

# peer is mandatory for CTC and IUCV
# It is the IP address of the other end of the point-to-point
# IP connection, and therefore the address of the corresponding
# CTC or IUCV interface on the VM TCPIP stack

# partner is mandatory for IUCV
# It is the VM userid of the peer: the TCP/IP stack you're connecting
# to.  CTC requires this too, but it's done in PROFILE EXEC with
# a COUPLE statement, rather than here.

# protocol is optional for CTC
# "0" means that the CTCs are cross-connected, even-to-odd and odd-
# to-even; "1" means that they're connected odd-to-odd and even-
# to-even.  "0" and a cross-connection is recommended.

# portnum is optional for OSA or LCS
# "0" or "1": which port on the card.  You probably don't need to
# touch this.

# netmask is optional for LCS and qeth, but recommended.  It is
# the network mask for your interface.  Often this is 255.255.255.0
# but it depends on your local network architecture.

# broadcast is optional for LCS and qeth.  It is the IP address of
# the broadcast address for your LAN; sending an IP datagram to this
# address will broadcast it to all devices on the LAN.  It is ignored
# for Hipersocket devices prior to z/VM 4.4, because Hipersockets
# do not support broadcast.  This address is the "all ones" address
# for a given LAN segment--given your address and netmask, it is the
# highest address in that segment.  Thus if your address is a.b.c.d
# where d is between 1 and 254, and your netmask is 255.255.255.0,
# the broadcast address will always be a.b.c.255.

# gateway should be specified for non-point-to-point interface types
# (LCS and qeth).  If you have more than one interface,
# only specify gateway for a single interface.  Any traffic for
# hosts that are not directly-connected (on the same LAN segment as
# SMPPLUS) will be routed through the gateway.
# It is not necessary to specify a gateway for a CTC or IUCV
# connection: because they are point-to-point links, the other end
# of the link is automatically the gateway.
```

---

Figure 17 (Part 2 of 4). BOOT PARAMS for SMPPLUS

---

```
# Example interface definitions
# Here are a few example interfaces to get you started.
# Uncomment the one that's the right type and modify the parameters
# to fit your site.

# This is an OSA-type QDIO adapter at 10.1.1.2. Its netmask is
# 255.255.255.128, and the gateway is at 10.1.1.1. As it happens
# it's a virtual OSA on guest LAN LNXLAN1, which is its portname too.
# The device address is 0x340.

#interface.0=eth0
#type.eth0=qdio
#address.eth0=0x340
#ip.eth0=10.1.1.2
#netmask.eth0=255.255.255.128
#broadcast.eth0=10.1.1.127
#gateway.eth0=10.1.1.1
#portname.eth0=LNXLAN1

# Here we have a Hipersocket interface at 0x344. Its IP address is
# 10.1.5.2; its netmask is 255.255.255.192, and its gateway is
# at 10.1.5.1.

#interface.0=hsi0
#type.hsi0=qdio
#address.hsi0=0x344
#ip.hsi0=10.1.5.2
#netmask.hsi0=255.255.255.192
#gateway.hsi0=10.1.5.1

# This is an LCS Ethernet at 0x348. Its address is 10.1.8.2; its
# gateway is 10.1.8.1, and the netmask is 255.255.255.0. That makes
# the "all ones" broadcast address 10.1.8.255

#interface.0=eth0
#type.eth0=lcs
#address.eth0=0x348
#ip.eth0=10.1.8.2
#netmask.eth0=255.255.255.0
#broadcast.eth0=10.1.8.255

#Next we have a CTC at 0x34b. It's at 10.1.7.2 and its peer is
# at 10.1.7.1

#interface.0=ctc0
#type.ctc0=ctc
#ip.ctc0=10.1.7.2
#peer.ctc0=10.1.7.1
#address.ctc0=0x34b
```

---

Figure 17 (Part 3 of 4). BOOT PARAMS for SMTPLPLUS

---

```
# Finally, we have an IUCV connection.  It's connected to the
# TCPIP virtual machine.  Its address is 10.1.8.2 and the TCPIP
# end of the IUCV connection is at 10.1.8.1
# interface.0=iucv0
# type.iucv0=iucv
# ip.iucv0=10.1.8.2
# peer.iucv0=10.1.8.1
# partner.iucv0=TCPIP
```

---

Figure 17 (Part 4 of 4). BOOT PARAMS for SMTPPLUS

---

## Configuring Mail Processing Parameters in EXIM4 CONF

Mail processing is under the control of EXIM4 CONF. Edit it to modify your mail parameters. Some of the most useful customizations are detailed below.

The structure of the Exim 4 configuration file, and all Exim's recognized parameters, can be found in the Exim specification document at <http://www.exim.org/exim-html-4.30/doc/html/spec.html>.

---

### Delivering mail in BSMTP format

Mail can be delivered to the spool in BSMTP format, rather than as an RFC-822 mail message.

This is done with the following modification to EXIM4 CONF, in the "vmpunch" transport section:

---

```
use_bsmtp
message_prefix = "HELO $primary_hostname\n"
message_suffix = ".\nQUIT\n"
```

---

Figure 18. Delivering mail in BSMTP format

If your site uses a MAILER virtual machine, use of BSMTP delivery is required.

---

### Delivering Outbound Mail

Depending on your installation choices, you will need to configure how SMTPPLUS attempts to deliver outbound mail messages. Modify the EXIM4 CONF file to reflect your choice of delivery method from the selections below.

#### Delivering in MX Mode

In MX mode, the processing engine will deliver mail directly to the remote site's mail exchanger machine. Therefore, in the Exim remote\_smtp transport, you will need a set of rules that look like:

---

```
remote_smtp:
  driver = smtp
  debug_print = "T: remote_smtp for $local_part@$domain"
  user = mail
```

---

Figure 19. Exim remote\_smtp for MX Mode

---

## Delivering in Proxy Mode

If you are running in proxy mode, you will deliver SMTP mail to a local port which drives the outbound application proxy. You need to ensure that the `remote_smtp` configuration looks like:

---

```
remote_smtp:
  driver = smtp
  hosts_avoid_tls = *
  connect_timeout = 30s
  command_timeout = 10s
  hosts_override = true
  hosts = VM-stack-addr
  port = ${run { /usr/local/bin/pickport.pl \
    VM-stack-addr -c 49999 -h $host_address -p 25 \
    -s VM-TCPIP-userid }}
  debug_print = "T: remote_smtp for $local_part@$domain at $host $port"
```

---

Figure 20. Exim `remote_smtp` for Proxy Mode

Note that you must insert the IP address and VM `userid` of the VM TCPIP stack connected to the application proxy virtual machine in the "port" statement shown above for this configuration to function correctly. If you use this configuration, it is highly recommended to block access to the proxy control port (49999 shown above) from the rest of your network to avoid problems with users manipulating the application proxy for nefarious ends.

---

## Configuring Host Masquerading

Host masquerading is a way to configure outgoing mail so that it appears to have come from a different host than the fully-qualified-domain-name of the actual host running the mail transfer agent. This is useful for those sites that want to strip hostname information from outgoing email (that is, to make all mail look as if it came from `user@domain.tld` rather than `user@host.domain.tld`), or who want to make mail coming from SMTPPLUS look as if it came from the VM host.

Host masquerading is controlled in EXIM4 CONF. Set the `primary_hostname` and `qualify_domain` directives to the name of the host with which outgoing email will be stamped.

In a proxy installation, `primary_hostname` should generally be what the VM host defines as its name as an SMTP mail exchanger. In an MX setting, it should probably be the hostname of the Linux host. However, if SMTPPLUS is acting as a mail exchanger for multiple hosts—a domain or subdomain—then this should be set to the domain name rather than its host name spanning those hosts.

For instance, if `vm.biz.example` (a VM SMTP server) currently handles all mail for `biz.example`, and you want mail from your users to come from "user@biz.example" (rather than either "user@smtpplus.biz.example" or "user@vm.biz.example"),

---

```
primary_hostname = vm.biz.example
qualify_domain = biz.example
```

---

Figure 21. Setting the hostname and domain for SMTPPLUS.

---

## Configuring Host Access

### Configuring Host Access for SMTP transactions

SMTPPLUS can be told to refuse to accept connections from particular hosts. In the acl section of EXIM4 CONF, you would create a "deny hosts" or "accept hosts" list. In this example we presume that vm.biz.example is behind a firewall and that only hosts in 10.0.0.0/8 should be able to talk to it; mail coming in from the outside world comes to a different mail server first, and is relayed from there (presuming the relay host is in the 10/8 private network) to VM. This will prevent any host with an IP (v4) address which does not start with 10 from sending mail to or through SMTPPLUS:

---

```
accept hosts = 10.0.0.0/8
```

---

Figure 22. Setting a host ACL for SMTPPLUS

Access lists can contain IPv6 and hostnames (including wildcards) as well as IPv4 addresses. Note that the host list is given in CIDR (Classless Internet Domain Routing) format; an explanation of CIDR can be found at <http://public.pacbell.net/dedicated/cidr.html>.

### Configuring Host Access for Relaying

A host that is a mail relay for a site allows that site to send mail to an arbitrary destination via its SMTP service. This is commonly used so that desktop systems do not need to do their own mail routing, and centralization of outbound SMTP through a relay allows virus and spam scanning to be performed at that outbound relay for mail leaving the site. This setting can be changed in the relay\_from\_hosts configuration setting.

---

#### WARNING

---

If you are running SMTPPLUS in proxy mode. SMTPPLUS will see all connections to it as coming from VM; therefore, opening relaying from VM will allow your system to be used as an open relay. This is almost certainly undesirable.

The best way to fix this is to allow relaying only for authenticated SMTP users. However, this will require changes to your end-users' configuration files and may therefore be unacceptable.

---

End of WARNING

---

Here we allow relaying for any host in the 10.0.0.0/8 network, as well as localhost. We will also include the IPv6 version of the local address, which is all zeroes except for a one in the last octet. In IPv6 notation, leading or trailing zeros can be omitted; nulls between the colons separating the parts of the address are assumed to represent zeroes.

The processing engine supports IPv6 as well as IPv4 natively.

---

```
hostlist relay_from_hosts = 127.0.0.1 : :::::1 : 10.0.0.0/8
```

---

Figure 23. Setting a relay ACL for SMTPPLUS

---

## Accepting Mail for Multiple Domains

You may want to accept mail for multiple domains; in this example, let's assume that `vm.biz.example` accepts mail not only for `biz.example` but also for `newbiz.example`, whom they just acquired. This is changed with the `local_domains` directive:

---

```
domainlist local_domains = @:newbiz.example
```

---

Figure 24. Accepting mail for multiple domains

(The "@" there simply means "my own domain", which is, in this case, "biz.example".)

Currently, alias resolution in `MAPUSER PL` assumes a single domain. This restriction will be relaxed in future releases.

---

## Configuring SMTP TLS Authentication

Exim supports several varieties of SMTP authentication. This topic is covered in detail in Chapters 33-37 of the Exim Specification at <http://www.exim.org/exim-html-4.30/doc/html/spec.html>.

### Example: Plaintext SMTP Authentication

Most systems supporting TLS authentication (and most microcomputer systems) support only the simple plaintext TLS authentication method. Authorized users who are allowed to relay messages through the server are identified in the file `/etc/authpwd` within the processing engine.

To perform plaintext authentication against a user password file in Berkeley DB format held in `/etc/authpwd`, in `EXIM4 CONF` provide the following lines:

---

```
server_condition = ${lookup{$2}dbm{/etc/authpwd}\
  ${if eq{$value}{$3}{yes}{no}}{no}}
```

---

Figure 25. Establishing a plaintext SMTP authenticator

This configuration looks for a entry in `/etc/authpwd` for a user and password, and denies access to the server if correct credentials are not presented.

Exim supports more comprehensive TLS configurations which are too complex to cover in this document. Please contact us if you have questions about advanced TLS authentication methods.

---

## Configuring Alias Resolution in MAPUSER PL

Exim contains many different mechanisms for alias resolution, including a traditional Berkeley DB map for aliases, LDAP resolution, or finding the alias in an RDBMS. SMTPPLUS centralizes these decisions in a single external Perl function, allowing a single point of control, rather than scattering the settings through different files.

Alias resolution is done by the Perl script MAPUSER PL. See below for a couple of examples.

---

### Default behavior

The default behavior of MAPUSER PL is the same as the default behavior of VM IMAP delivering to unenrolled users: the first eight characters

of the local part of the email address on the message envelope are uppercased, and that is used as the VM userid to which to attempt delivery. If that does not resolve to a VM userid, the mail is delivered to a catchall user, by default MAINT.

Here is the default MAPUSER PL as shipped:

---

```
#!/usr/bin/perl

# mapuser.pl provides three functions: mapuser, setpunch, and
# defaultuser.

# mapuser takes a string as input and returns the corresponding user
# to whom to punch the mail

# setpunch attempts to point the punch at the user you give it; you
# must pass $fromuser as well, or all mail will be punched as UNKNOWN
# MAIL. The third argument is a 1 or a 0.
# It specifies whether the file should be of type NOTE (1)
# or type MAIL (0).

# defaultuser is a convenience function that returns the name of the
# account that gets mail if it cannot be determined who it should really
# go to.

sub mapuser {

    # replace as necessary
    # This implementation sends to the user if the user exists
    # Otherwise it goes to defaultuser();

    my $user = shift;

    system("$HCP Q $user >/dev/null");
    my $ret = $? >> 8;
    debug("CP Q $user returned $ret.");
    if (($ret == 0) or ($ret==45)) {
        return uc($user);
    } else {
        return defaultuser();
    }
}

# This version of mapuser is for MAILER support; note that it goes
# with another version of defaultuser()

# sub mapuser {
#     return defaultuser();
# }
```

---

Figure 26 (Part 1 of 2). MAPUSER PL for SMTPPLUS

---

```

# This version is for LDAP support.
# It assumes that your LDAP schema includes an attribute called
# "vmuserid" which holds the equivalent VM userid, to whom you
# punch the message
#
# sub mapuser {
#     use Net::LDAP;
#     my $ldap=Net::LDAP->new('ldap.biz.example');
#     my $r = $ldap->search( base => 'cn=biz,cn=example',
#                           filter => 'uid=$user',
#                           attrs => 'vmuserid');
#     my $s = $r->entry(0); # This should be the only one
#     my $n = $s->get_value("vmuserid");
#     return ($n ? $n : defaultuser());
# }

sub setpunch {
    my $user = shift;
    my $fromuser = shift;
    my $type = shift;
    my $f = uc($fromuser);
    $f = "UNKNOWN" unless ($f);
    if (length($f) > 8) {
        $f = substr($f,0,8);
    }
    $type = ($type?"NOTE":"MAIL");
    my $s = "$HCP SPOOL PUN TO $user CLASS A NAME $f $type>/dev/null";
    debug($s);
    system($s);
    my $ret = $? >> 8;
    die "CP SPOOL PUN TO $user failed with retcode $ret!\n" if ($ret);
    debug("CP SPOOL PUN TO $user succeeded");
    return;
}

sub defaultuser() {
    # Usually MAINT
    return "MAINT";
}

# For MAILER support, assuming MAILER is the name of the mailer machine:
#
# sub defaultuser() {
#     return "MAILER";
# }

# Defined return code so require works.
1;

```

---

Figure 26 (Part 2 of 2). MAPUSER PL for SMTPPLUS

---

## MAILER support

MAILER support can be added very easily by modifying MAPUSER PL so that it returns the name of the MAILER machine regardless of its input.

If the mailer machine is named MAILER, then you would uncomment the `mapuser()` and `defaultuser()` functions described as "for MAILER support" in MAPUSER PL and comment out the default functions.

All incoming mail will be passed to MAILER, and the MAILER machine will be responsible for delivering it. You should also modify `exim4.conf` to deliver in BSMTP format, as shown in Figure 18 on page 35.

---

## LDAP support

Because MAPUSER PL is simply a Perl subroutine, it can include the necessary configuration to bring in a Perl LDAP interface, and construct a mapping based on an LDAP lookup.

Here we will assume that the Base DN is "cn=biz, cn=example" and that the LDAP implementation, which lives on `ldap.biz.example`, allows an anonymous bind. The LDAP schema has been constructed such that "uid" is the username of the incoming email address, and "vmuserid" is the attribute that contains the VM userid corresponding to the uid, and that for any uid there is at most one vmuserid attribute.

To use this method, uncomment the `mapuser()` function labeled "for LDAP support" in MAPUSER PL, and comment out the default `mapuser()`.

---

## Configuring Spam and Virus scanning in AMAVISED CONF

Both spam and virus scanning run under the control of the amavis toolkit. Its behavior is configured in AMAVISED CONF, which is extremely heavily commented Perl code. Some of the more useful options to change are detailed below.

---

### Configuring Virus Scanning in AMAVISED CONF

The virus scanning parameters are defined in AMAVISED CONF.

### Configuring Virus Notification Mail

There are three forms of virus notification:

1. Inform the sender that he or she sent a mail with a virus attached; this is rarely useful to anyone, particularly since most viruses forge the sender.
2. Inform the recipient that his mail was quarantined or discarded because it contained a virus. This is also of limited utility.
3. Inform a system administrator that a virus-laden message was intercepted.

The first setting can be enabled in AMAVISED CONF by setting

---

```
$final_virus_destiny = D_BOUNCE; # D_REJECT probably a better idea in
                                # the general case
$warnvirussender = 1;
```

---

Figure 27. Warn sender when virus detected

The intended recipient can be warned by setting:

---

```
$warnvirusrecip = 1;
```

---

Figure 28. Warn recipient when virus detected

Finally, mail to an administrator can be set. Let's say it should go to MAINT:

---

```
$virus_admin = "maint";
```

---

Figure 29. Warn administrator when virus detected

### Configuring Default Action when virus Detected

By default, email containing a virus is put into quarantine in `/var/lib/amavis/virusmails`. This can be changed (or turned off) by changing the `$virus_quarantine_to` setting in AMAVISED CONF.

This example turns it off:

---

```
$virus_quarantine_to = undef;
```

---

Figure 30. Turning off virus quarantine

## Configuring Retention of Quarantined Files

By default, files kept in quarantine will be deleted if they have not been accessed in a week. This is done with an entry in system crontab file (`/etc/crontab`). If you wanted instead to check at 2:00 each morning, and purge any file in quarantine that has not been accessed in the previous two weeks:

---

```
0 2 * * * root find /var/lib/amavis/virusmails -atime 14 -print | \
    xargs -n 100 rm
```

---

Figure 31. Automated daily quarantine purging

## Configuring Automated Virus Pattern File Updates

The frequency with which freshclam updates its virus pattern database is stored in `/etc/clamav/clamav-freshclam-handledaemon.conf`. The default of `FRESHCLAM_CHECK_FREQUENCY=5`, which updates five times a day, is acceptable for most sites. If your system lacks IP connectivity to the outside world, you will not be able to access the signature database and may wish to disable this feature; to do so, modify `/etc/clamav/clamav-freshclam-handledaemon.conf` and add the following:

---

```
FRESHCLAM_CHECK_FREQUENCY=0
```

---

Figure 32. Disabling Virus Database updates

---

## Configuring Spam Abatement Parameters in AMAVISD CONF

SpamAssassin also runs under the control of Amavis, and, like the virus scanning functionality, most settings can be modified by changing `AMAVISD CONF`.

### Spam Scoring

Spam scanning is conceptually similar to virus scanning; where a virus scanner looks for a pattern of bytes it knows to be a hallmark of a particular virus, the spam scanner looks for patterns that indicate the message is likely to be spam. This can be a straightforward pattern-match (for instance "CHEAP VIAGRA"), or it can be a more complex heuristic (for instance, a large proportion of the user-visible text is broken up by in-line HTML comments).

SpamAssassin runs its checks for each message it receives, sums the results of each check, and assigns that sum as the message's spam score. A score above a (user-configurable) threshold is defined as indicative of a spam message.

## Configuring Default Action when Spam Detected

This is controlled in AMAVISD CONF, and works exactly like virus detection; the only difference is that the string "virus" in each configuration setting becomes "spam".

### Warning the sender when spam is received

Let's say that you want to bounce spam and warn the sender (these are likely not to be what you want to do; most spam does not have a legitimate return address on it):

---

```
$final_spam_destiny = D_BOUNCE;
$warnspamsender = 1;
```

---

Figure 33. Warning a spam sender

This highlights a general feature of AMAVISD CONF and, more generally, SMTPPLUS as a whole: there are a great many places you can set policy, but in most cases, the default policy is the right one.

## Configuring Spam Tagging and Rejection Levels

These settings can also be found in AMAVISD CONF. The default behavior is as follows:

---

Level	Action
0.0	Tag
5.0	Rewrite Subject
10.0	Discard

---

Figure 34. Default Spam Tagging and Rejection Configuration

A more aggressive policy might look like this:

---

Level	Action
0.0	Tag
3.5	Rewrite Subject
6.0	Discard

This would be accomplished with the following directives in AMAVISD CONF:

```
$sa_tag_level_deflt = 0.0;
$sa_tag2_level_deflt = 3.5;
$sa_kill_level_deflt = 6.0;
```

---

Figure 35. A more aggressive Spam Abatement Policy

Now let's assume that you have a bunch of tools in place at your site that look for the string "!!SPAM!!" in the header (rather than the default "\*\*\*SPAM\*\*\*"). Change this:

---

```
$sa_spam_subject_tage = '!!SPAM!! ';
```

---

Figure 36. Changing spam subject rewriting policy

## Configuring Retention of Spam Messages

The `spam_quarantine_to` directive, in AMAVISD CONF, allows you to control the destination—if any—of received spam. The default is to discard it.

This differs from the virus policy for this reason: it is common to get legitimate, but virus-infested mail. If someone whose machine is infected sends mail, it will have a virus attached. That mail may still be important, which is why it is sent to a quarantine directory so the site administrator has the option of manually forwarding it or notifying the intended recipient.

On the other hand, spam, almost by definition, has no redeeming features, and nothing will be lost by discarding it. The only risk is that a legitimate message will have a high enough spam score to trigger deletion. At a discard threshold of 10.0, this risk is negligible.

### Example of Retaining Spam Messages

Let's say you want to capture incoming spam and send it to an alias, "spammy"; you're doing this to build up a training database for your Bayes algorithms (the details of arranging delivery for "spammy" so that it bypasses the VM punch are left as an exercise for the reader).

---

```
$spam_quarantine_to = 'spammy@';
```

---

Figure 37. Setting a spam quarantine recipient

---

## Configuring Automated Removal of Certain Attachment Types

It may prove useful to simply deny certain attachments from being exchanged at all. Certainly, a filter that discards Windows executables will result in a safer system; however, if your users are in the habit of exchanging programs via email, this is obviously not a valid policy choice for you.

The default is to ban all double extensions (a very common virus tactic is to use an innocuous extension followed by the actual one) and to further ban .exe, .vbs, .pif, .scr, .bat, and .com. This can be changed by changing the `$banned_filename_re` directive in `AMAVISD CONF`.

Let's say that we've instituted a sitewide policy that, in addition, bans mailing MP3s and JPGs. Although this can be bypassed simply by changing the filename, it will still help limit the traffic.

---

```
$banned_filename_re = new_RE(
qr'\.[a-zA-Z][a-zA-Z0-9]{0,3}\.\.
(vbs|pif|scr|bat|com|exe|dll|jpg|mp3)$'i,
qr'\.(exe|vbs|pif|scr|bat|com|jpg|mp3)$'i,
);
```

---

Figure 38. Adding banned file extensions

### SpamAssassin-specific parameters in LOCAL CF

LOCAL CF provides the ability to customize some advanced options for SpamAssassin. Please consult the SpamAssassin documentation before enabling these options; they are quite aggressive and if misused, may cause large amounts of mail to be falsely rejected.

In general, you should not need to modify LOCAL CF, which controls SpamAssassin's behavior. The most common configuration options can be modified from `AMAVISD CONF`.

---

## Configuring the SMTPPLUS Application Proxy

The application proxy is designed for the benefit of those sites which cannot easily convince their network administrators to allocate an additional IP address to VM or to allow VM TCPIP to act as a router. It accepts incoming connections on port 25, but rather than routing them to VM SMTP, it reroutes them to the SMTPPLUS server. Likewise, when SMTPPLUS attempts a delivery, rather than delivering directly to the destination host, SMTPPLUS delivers to a local port which is connected to a port on VM, which in turn forwards it to the real destination.

---

### WARNING

This has security implications: most notably, IP-based relay control will not work in this scenario. See the earlier section on access control for details of alternative configuration.

---

End of WARNING

---

## Linking to the Application Proxy Code

The replacement code for the SMTP virtual machine will be loaded onto SMTPPLUS's F191 minidisk. SMTP must have a write link to this minidisk in order to function, so it is imperative that SMTPPLUS detach F191 when it starts up.

All necessary files are put onto the F191 minidisk as part of the installation process.

---

## Configuring Proxy Settings

There are two proxy settings that need to be managed. The first is the exit exec that the IPMAPPER proxy runs once a minute. This should, in general, transfer the contents of its spool to the SMTPPLUS machine; this way, users whose mailers are configured to punch mail to SMTP will not need to change anything. This file is called IPMAPUTL EXEC. A useful configuration would look like this, assuming that the name of the SMTPPLUS machine is passed in as the first argument.

---

```
/* the IPMAPPER Utility Exec */
/* run once a minute when IPMAPPER is running.....*/
/* can be used to do any useful IPMAPPER work */
parse arg name .
'CP TRANSFER RDR ALL 'name
exit
```

---

Figure 39. IPMAPPER Utility Exec

---

## Setting Up Port Proxying

There are two types of port proxying IPMAPPER does; one is the static inbound proxying to map VM port 25 to SMTPPLUS's SMTP port. The other provides transient mappings of outbound ports under the control of SMTPPLUS. These mappings are defined in the file PORT MAPPING on SMTPPLUS F191.

### Static Port Proxying

The static proxy is provided by the first line of PORT MAPPING. The first two characters should be 25 (to indicate port 25); then one or more blanks (EBCDIC 0x40), the IP address of the SMTPPLUS stack, one or more blanks, "25" (the port to which port 25 of the VM stack is redirected), one or more blanks, and the VM userid of the stack (usually TCPIP) that you are forwarding.

This example forwards port 25 to port 25 of the host at 10.1.1.2 via the VM TCPIP virtual machine.

---

```
25          10.1.1.2      25    TCPIP
```

---

Figure 40. IPMAPPER static port mapping

### Transient Port Proxying

The second type of port proxying is a little simpler: IPMAPPER must maintain a list of transient outgoing ports which SMTPPLUS can send mail through. This should not be changed from the default, which is:

---

```
50000      FREE
50001      FREE
50002      FREE
50003      FREE
50004      FREE
50005      FREE
50006      FREE
50007      FREE
50008      FREE
50009      FREE
50010      FREE
50011      FREE
50012      FREE
50013      FREE
50014      FREE
50015      FREE
50016      FREE
50017      FREE
50018      FREE
50019      FREE
```

---

Figure 41. IPMAPPER free port set

All this does is to reserve ports 50000-50019 for transient outgoing connections; where they connect to is driven by SMTPPLUS.

---

## Restricting Access to the Outbound Proxy

IPMAPPER listens on port 49999 and speaks a simple control protocol to assign arbitrary host and port combinations for outbound proxy services. It is undesirable for the control port to be accessible to the network at large, because then anyone, anywhere, can use the IPMAPPER virtual machine as a transparent proxy and could mount attacks that appeared to come from your machine.

However, it is easy to fix this: the file IPAUTH MAPPING on SMTPPLUS F191 contains a list of IP addresses allowed to use the control port to establish an outbound proxy. This should, in general, be limited to the IP address of the processing engine, although for testing purposes you may want to also allow the IP address of the VM stack to drive the control port. Here is an example that allows VM and the processing engine access; VM is at address 10.1.1.1, and the processing engine at 10.1.1.2.

---

```
10.1.1.1
10.1.1.2
```

---

Figure 42. IPMAPPER IP authentication map

---

## IPMAPPER Startup

The PROFILE EXEC on SMTPPLUS F191 may need to be changed if your SMTPPLUS virtual machine is not named SMTPPLUS, or if you are testing on a secondary TCP/IP stack. Simply set the values of the variables **tcpip** and **smtp** to reflect what is in use at your site.

---

```

/*****/
/* File:          PROFILE EXEC (for application proxy)    */
/* Release:       1.0.0                                  */
/* Project:       SMTPPLUS VM SMTP Replacement           */
/* Description:   IPMAPPER startup exec                  */
/* License:       Copyright 2004 Sine Nomine Associates  */
/*****/
/* New Profile Exec to invoke IPMAPPER */
Address Command
tcpip = 'VM-TCPIP-userid'
smtp  = 'SMTPPLUS-userid'
'set language (add bk w user'
'IPMAPPER 'tcpip smtp

```

---

Figure 43. Modifying IPMAPPER's PROFILE EXEC

Start IPMAPPER by logging in to SMTP, re-IPLing CMS, interrupting SMTP's startup procedure by typing a non-blank character followed by Enter, and then doing the following:

---

```
ACCESS F191 A
EXEC PROFILE EXEC A
```

---

Figure 44. Starting IPMAPPER within SMTP During Testing



---

# Testing SMTPPLUS

---

## Testing SMTPPLUS

To test SMTPPLUS, it is necessary to check several things:

- Mail from outside SMTPPLUS, destined for VM users, is delivered properly.
- Mail from VM users, destined for users outside the S/390 or zSeries machine, is delivered properly.
- The mail route travels through the amavis scanning framework, ensuring that virus and spam scanning is occurring.

We will test all of these things and watch the Exim log file to determine that mail processing is happening as it should.

For this step, you will need to be viewing the log file as it develops. Log into SMTPPLUS either at the console or via ssh and run "tail" on the log file in filter mode:

---

```
tail -f /var/log/exim4/mainlog
```

---

Figure 45. Using "tail" in filter mode

For the remainder of this example, we will assume the following:

- The SMTPPLUS machine is named smtpplus.biz.example and is at IP address 10.1.1.2. Its gateway to the VM TCPIP stack is at 10.1.1.1.
- The VM machine is named vm.biz.example.
- Your workstation is named user1.biz.example, and is at IP address 10.1.2.44.
- The VM userid you are testing with is USER1.
- The external address you are testing with is user1@otherbiz.example and you have some way to check its mail.

---

# Testing Mail Delivery to VM Users

---

## Testing Mail Delivery to VM Users via SMTP

First we will test delivery to the VM user USER1. This requires connecting to port 25 of the SMTPPLUS machine with a telnet client. Use whichever desktop telnet client you prefer. However, do not use CMS telnet, as its attempt to use the tn3270 protocol will confuse the SMTP port of SMTPPLUS.

### Making a Telnet Connection to the SMTPPLUS Server

#### Connecting in MX Mode.

If you are testing MX Mode, make the telnet connection to **smtpplus.biz.example:25**.

#### Connecting in Proxy Mode.

If you are testing Proxy Mode, make the telnet connection to **vm.biz.example:25**.

### Creating an Example Mail Transaction

Once you have done this, you should see a banner that looks something like the following:

(We will use here the Unix convention that long lines are continued with a backslash character; in actuality, these will be on a single, long line, and of course times and dates will differ.)

---

```
220 smtpplus.biz.example ESMTP Exim 4.30 Mon, 01 Apr 2004 16:20:00 \
-0500
```

---

Figure 46. SMTP Banner

Now enter the following dialogue; what you are doing is manually creating a message to be delivered to the USER1 user.

---

```

HELO user1.biz.example
250 smtpplus.biz.example Hello user1.biz.example \
    [10.1.2.44]

(If you are in Proxy Mode, the IP address at the end of the line will
show [10.1.1.1])

MAIL FROM: <user1@biz.example>
250 OK
RCPT TO: <user1@vm.biz.example>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
Subject: Test Message
X-Are-You-Paying-Attention: This header is useless and can be omitted

This is a test. This is only a test. Film at 11.
.
250 OK id=1Ac9bF-0005hs-7t
QUIT
221 smtpplus.biz.example closing connection

```

---

Figure 47. SMTP Transaction Example Dialogue

## Verifying Delivery via Log Files

Meanwhile, the log file should have been displaying something like the following:

---

```

2004-03-01 16:20:05 1Ac9bF-0005hs-7t => user1@biz.example \
    R=amavis_inbound T=amavis H=localhost #127.0.0.1
2004-03-01 16:20:07 1Ac9bF-0005hs-7t Completed
2004-03-01 16:20:08 1Ac9dB-0005iM-02 => user1 <user1@vm.biz.example> \
    R=nocheck_local T=vmpunch
2004-03-01 16:20:08 1Ac9dB-0005iM-02 Completed

```

---

Figure 48. Log file for inbound SMTP transaction

There are two important features to notice here: first, the message transaction you entered is tagged with an identifier ("1Ac9bF-0005hs-7t") so you can recognize it in the log file. Second, after the message was accepted, it was routed to Amavis where it was spam and virus scanned, before it was given back to Exim for delivery via the "vmpunch" transport, which delivered it into your spool (possibly by way of a mailer machine, depending on how you configured MAPUSER PL earlier).

Now log on at a different terminal, leaving the view of the log file open, and verify that the mail was indeed delivered to USER1.

When you view the message, you should see an additional header in it:

---

```
X-Virus-Scanned: by amavisd-new-20030616-p5 (Debian) at \
smtpplus.biz.example
```

---

Figure 49. Amavis-added X-Virus-Scanned: header

This header was added by Amavis as it scanned the mail.

---

## Testing Mail Delivery from VM users via BSMTP Punch Files

Now you will test that a VM user can send mail via SMTPPLUS.

### Creating a Sample BSMTP Punch File

The first thing to do is to create a BSMTP transaction as a CMS file. As USER1, create the following file as BSMTP MESSAGE A:

---

```
HELO vm.biz.example
MAIL FROM: <user1@biz.example>
RCPT TO: <user1@otherbiz.example>
DATA
Subject: Test of outbound mail via SMTPPLUS
X-Still-Paying-Attention: Another useless header
```

Test of SMTPPLUS outbound capability, submitted via the punch.

```
Love Always,
User1
.
QUIT
```

---

Figure 50. BSMTP message to test outbound SMTP

### Redirecting the Virtual Punch

Now you must redirect your virtual punch appropriately.

#### Redirecting the Virtual Punch in MX Mode

If you are in MX Mode, execute the following command:

---

```
CP SPOOL PUN TO SMTPPLUS
```

---

Figure 51. Directing the virtual punch for MX Mode

#### Redirecting the Virtual Punch for Proxy Mode

If you are in Proxy Mode, execute the following command:

---

```
CP SPOOL PUN TO SMTP
```

---

Figure 52. Directing the virtual punch for Proxy Mode

## Sending the message

Now you need to actually send the message, by executing the following:

---

```
PUNCH BSMTP MESSAGE A ( NOHEADER
```

---

Figure 53. Sending BSMTP via the punch

## Resetting the virtual punch

Now you should point your virtual punch back at yourself:

---

```
CP SPOOL PUN TO *
```

---

Figure 54. Resetting the virtual punch

## Verifying Delivery via Log Files

Wait up to two minutes for the message to be accepted by SMTPPLUS: if you are in Proxy mode, IPMAPPER transfers its spool files to SMTPPLUS once a minute, and SMTPPLUS checks once a minute to see whether it has any new spool files to send.

In the log, you should see something resembling:

---

```
2004-03-01 16:25:01 SMTP connection from root
2004-03-01 16:25:01 1AcAHW-0005z0-Ls <= user1@biz.example U=root \
    P=local-bsmtp S=483
2004-03-01 16:25:03 1AcAHW-0005z0-Ls => user1@otherbiz.example \
    R=dnslookup T=remote_smtp H=mx.otherbiz.example \
    [10.100.1.20]
```

---

Figure 55. Log file for outbound SMTP transaction

Note that in the default configuration we do not send mail originating from VM users via the spool interface through the spam or virus scanner: VM is not known as a virus-rich environment, and we presume that spammers will, in general run mass-mailing programs that are built for environments other than VM/CMS. Messages relayed through the SMTPPLUS engine via TCP are scanned as normal.

## Checking the external account

Now check the mail at **user1@otherbiz.example** and verify that the mail you sent did indeed arrive.

This testing completes the verification process.

---

# Putting SMTPPLUS Into Production

---

## Putting SMTPPLUS Into Production

The tasks required to put SMTPPLUS into production differ depending on whether you are using MX mode or proxy mode, although there are some common tasks for both scenarios. Both scenarios are discussed below.

---

### Common Tasks (MX and Proxy Modes)

#### Modifying the Production VM TCPIP Configuration

VM's TCPIP configuration (PROFILE TCPIP, or PROFILE *node*) will need to be changed to reflect the network connection to SMTPPLUS; this probably will already have been done while testing the service, as shown in Figure 2 on page 13, Figure 3 on page 13, and Figure 4 on page 13. However, if you were testing the product on an alternate stack, you will need to move it to the main VM TCP/IP stack and modify that stack's configuration.

#### Changing the SMTPSERVERID Entry in TCPIP DATA

TCPIP DATA (usually on TCPIP's 191-disk; however, users may have private copies, and they need to be notified of the change) must be modified so that the SMTPSERVERID setting is the SMTPPLUS machine.

---

```
SMTPSERVERID SMTPPLUS
```

---

Figure 56. Changes to TCPIP DATA for SMTPPLUS

The only effect this has is cosmetic: files arriving in users' readers will appear to have come from *userid AT node* rather from SMTPPLUS AT *localnode*.

This concludes the mode-independent modifications to place the SMTPPLUS engine into production.

---

## MX-Mode Specific Steps

There are four major steps to moving to production in MX mode:

1. The SMTPPLUS machine must be listed as the primary MX for the VM system in all the DNS servers accessible by systems sending mail.
2. Users must change application settings that use a SMTP server to point to SMTPPLUS.
3. Mailer machines, if any, must change the NJE destination for inbound and outbound servers to SMTPPLUS.
4. The VM TCPIP SMTP virtual machine must be disabled.

---

### Add MX records to VM DNS Entry

Your DNS servers must be updated to list the SMTPPLUS machine as the primary mail exchanger (with a smaller priority value than any other) for the VM system. If the processing engine is "smtpplus.biz.example" and the VM system whose SMTP service it is replacing is "vm.biz.example," and if there are no mail exchangers with priority values less than ten, then the appropriate entry in the zone file would be:

---

```
vm.biz.example IN MX 10 smtpplus.biz.example
```

---

Figure 57. DNS change for MX mode

---

### Users Change SMTP Server Settings

Any applications users employ to send mail from VM must be reconfigured to punch outgoing mail to SMTPPLUS, rather than SMTP.

Please consult the documentation for your application to determine how to perform this step.

---

### Mailer Configuration Reflects New SMTP Servers

Any mailer applications must also be reconfigured to accept BSMTMP from SMTPPLUS and send outbound SMTP messages to SMTPPLUS rather than SMTP.

Please consult the documentation for your mailer to determine how to perform this step.

---

### Disabling the VM TCPIP SMTP Server

In order for the MX function to operate correctly, the VM TCPIP SMTP server must be disabled by removing port 25 from the list of ports managed by VM TCPIP, and ceasing to autolog the userid running the VM TCPIP SMTP code. To disable the VM TCPIP SMTP server, comment the following lines in PROFILE TCPIP out as follows:

---

```
AUTOLOG
# SMTP  password

PORT
# 25 TCP SMTP
```

---

Figure 58. Removing SMTP from TCPIP's control

When this has been done, turn off the VM SMTP service:

---

```
FORCE SMTP
```

---

Figure 59. Logging off the SMTP virtual machine

---

## Proxy Mode Specific Steps

There are only three steps required for enabling the proxy mode configuration.

---

### Updating SMTP DTCPARMS to Start the Application Proxy

The userid formerly running the VM TCPIP SMTP server must be modified to run the IPMAPPER code rather than the default SMTP server. On SMTP 191, create SMTP DTCPARMS with the following contents:

---

```
:Nick.SMTP      :Type.server   :Class.smtp
:VMLink.* F191 < F191 A > ( W
```

---

Figure 60. SMTP DTCPARMS for IPMAPPER invocation

---

### Editing PROFILE EXEC to Start the Application Proxy

Then on SMTP F191 (which is also SMTPPLUS F191), edit PROFILE EXEC to refer to the TCPIP stack you will be forwarding to SMTPPLUS and to the correct name for the SMTPPLUS virtual machine. The defaults of "TCPIP" and "SMTPPLUS" are usually fine; however, if you tested SMTPPLUS on a secondary stack, you may now need to move it to the primary TCP/IP stack.

---

### Restart the SMTP Virtual Machine

Now re-IPL SMTP, and it will start up running the IPMAPPER code instead of its own SMTP server.



---

# Maintenance Notes

---

## Maintenance Notes

This section includes some common maintenance items that should be checked periodically during server operation.

---

## Log Files

System messages will accumulate in `/var/log/messages`; mail-system specific messages will be in `/var/log/exim4/mainlog`.

There is a VM implementation of syslog available, thanks to Neale Ferguson, so you could, if you desired, consolidate all your log messages to VM. This, and other of Neale's Open Edition packages, can be found at <http://vm.marist.edu/~neale/vmoe.html>.

---

## Quarantine Directories

The system administrator will want to periodically examine the contents of the quarantine directories under `/var/lib/amavis` to ensure that the disk is not filling up with quarantined viruses or spam. The default installation will purge these directories of any file not accessed in a week, but this may not be frequent enough in very-high-traffic environments.

---

## Checking the Mail Queue

The length of the outbound mail queue can be checked with the command `"mailq"`; messages in the outbound queue reside in `/var/spool/exim4/input`. It is not unusual to have many bounce messages residing in the queue, in a "frozen" state. These will eventually be discarded. These are generally bounce notifications to addresses that sent viruses, and whose viruses were not in the database as forging the return address. Nevertheless, these messages do not have a legitimate return address, so the bounce message fails. You could, of course, suppress this behavior by changing the default virus destiny in `AMAVISD CONF` to `"D_REJECT"`, but this would not generate a warning notice to well-intentioned people sending legitimate, but virus-laden, mail to your users, and their mail would be silently discarded. The "correct" choice is a site policy decision you must make.

---

# Appendices

---

## Appendix A. Tape Layout

There is one 3480 cartridge tape in the installation package. Its contents are as follows:

- Tape 1

**File 1** CMS TAPE DUMP format containing

- INSTALL EXEC
- INSTALL HELPCMS
- MANIFEST LISTING
- SMTPPDIR EXAMPLE
- VMARC MODULE

**File 2** CMS TAPE DUMP format containing

- SMTPDOCS VMARC
- SMTPINST VMARC
- SMTTP192 VMARC
- VMARC MODULE
- SMTTP191 VMARC
- IPMAPPER VMARC

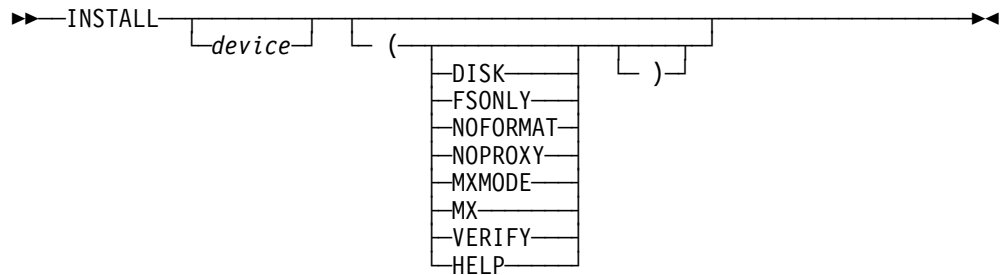
**File 3** DDR DUMP containing cylinders 0-399 of SMTPLUS 150

## Appendix B. INSTALL Command Syntax

### Purpose

- Install the SMTPPLUS product.

### Format



### Parameters

#### device

The address of the device (disk or tape) containing the installation files.

### Options

#### DISK

Specifies that installation is to occur from a minidisk rather than from tape. This option is incompatible with the **TAPE** *n* option.

#### FSONLY

Installation is to skip the resource-checking, formatting, and installation of CMS files, and proceed directly to the Linux DASD restoration. This option is incompatible with the **VERIFY** option.

#### NOFORMAT

Do not verify or format disk resources; install CMS files and then proceed to the DDR restoration. This option is incompatible with the **VERIFY** option.

#### NOPROXY

For an MX-mode installation: do not install the application proxy.

#### MXMODE

Synonym for **NOPROXY**.

#### MX

Synonym for **NOPROXY**.

#### VERIFY

Verify that expected disk resources are available and exit without installing anything. This option is incompatible with the **FSONLY** and **NOFORMAT** options.

**HELP**

Print a usage message and exit.

---

## Usage

INSTALL operates in the following sequence:

1. Verify that disk resources are present, correctly sized, and writeable.

*Exit here if VERIFY is specified.*

2. CMS format minidisks.

*Begin here if NOFORMAT is specified.*

3. Load VMARC MODULE and VMARC archives of CMS files into temporary disk space.

4. Unpack CMS file archives onto destination minidisks. *Begin here if FSONLY is specified.*

5. Restore DDR image of the Linux filesystem onto device 150.

*Execute this step if DISK is not specified.*

- a. Restore SMTPPLUS 150DDR from tape to device 150.

or

*Execute this step if DISK is specified.*

- a. Restore SMTPPLUS 150DDR from DASD to device 150.

6. Exit.



---

Program Number	Feature Number
LX01-0004	SMTP-0001

---

**Release 1.0**