Description of the changes for Version 14.3 of LISTSERV®

10 Dec 2004
Updated 12 Oct 2005

IMPORTANT: Install your maintenance LAK before upgrading!

A maintenance LAK (License Activation Key) must be installed before upgrading, or LISTSERV will not start after the upgrade. More information

Editorial Note – New Version Numbering

With this release, L-Soft is aligning LISTSERV's version numbering with the rest of the e-mail industry. There have been 48 released versions of LISTSERV since 1986 – 14 major upgrades and 34 minor releases. Version 1.8e in the "traditional" numbering system corresponds to 14.0, and the present update to 14.3. The next major release, 15.0, is planned for 1H2005.

Since LISTSERV 14.2 (formerly 1.8e-2003a) was not formally released other than together with L-Soft's LISTSERV Maestro product, we have included the 14.2 release notes in the present document for your convenience.

HIGHLIGHTS

Skip directly to the table of contents

Product Manager's Message

Version 14.3 is the second most extensive update in LISTSERV's 18-year history. Over 100 of the 174 changes in version 14.3 are a direct result of customer requirements, covering a broad spectrum from internationalization to database store procedures to list moderation. Based on customer feedback, we identified areas of the product requiring improvement, assigned priorities, and got to work. As usual, developers also made a list of things they wanted to include in the new version. To deliver as much customer value as possible by the cut-off date, we have focused on customer-suggested enhancements for 14.3, and postponed many of the developers' own ideas to 15.0. The result is a product with a comfortably familiar interface, containing an impressive collection of new customer-driven enhancements designed to further improve the e-mail list management experience for list subscribers, list owners and LISTSERV site administrators alike.

My personal top 5 feature pick is:

1. **"Send= Public,Confirm,Non-Member".** Say goodbye to spam on your list, while at the same time keeping it open for anyone to post. A favorite among beta-testers!
2. **Messages to the -REQUEST address now require "OK" confirmation.** Instant spam fatality if, like me, you are a list owner for dozens of lists.
3. **The new SPAM_EXIT "hook."** I am spoiled – I was the first to get to test it.
4. **The virus counters had to be increased to 64 bits.** This tiny change would not be worth writing about, but working on spam deflection technology is very frustrating, and sometimes it just feels nice to think about all the trouble you are *not* getting into thanks to

the built-in virus filter. The first person to make a spam filter as fast, precise and reliable as today's leading virus filters will get so much media coverage that he will have to retire to an underground cave to have a shot at living a normal life.

5. **Home page background rebuild.** When you change a site-wide template, either L-Soft's or your own, and list home pages depend on this template (even indirectly), LISTSERV automatically senses it and rebuilds all home pages for you. If you have LISTSERV-HPO, it does it in the background, letting other LISTSERV commands and tasks take precedence.

This is just my personal pick, and I am biased – the first three were my idea. There are well over a hundred improvements in just about every area of the product, and I am confident that you will find an improvement in 14.3 that makes a difference to you.

Headlines

The most notable areas of improvement in version 14.3 are:

- **Spam control.** Version 14.3 contains many improvements to curtail spam, including, in particular, "masking" of all e-mail addresses on the web interface until logged in, and an "exit point" allowing LISTSERV to be hooked into a third-party spam filter, such as *SpamAssassin*. Should LISTSERV's spam quarantine strike a well-meaning innocent, it can be lifted with a simple command. Spam prevention remains one of L-Soft's primary focus areas, since it is a rapidly moving target, and you can expect more spam control features in coming versions.

- **Improved internationalization and customization.** With the introduction of 72 new carefully chosen "message templates," version 14.3 allows a much finer degree of customization and translation of messages sent to list subscribers. This very extensive feature also includes improvements to the template processor and the capability to suppress system error messages completely, since it may be impossible to translate them. There are thousands of messages in LISTSERV and L-Soft expects to add more message templates with every coming version, but the messages selected for version 14.3 should cover at least 80% of scenarios.

- **Database integration: pre-approved messages and stored procedures.** To facilitate database integration, scripts can now use the new DISTRIBUTE POST command to submit pre-approved messages to LISTSERV for distribution, with the proper authorization of course. These pre-approved messages will not need to be confirmed with the "OK" mechanism even if the list normally requires it. DISTRIBUTE also supports stored procedures to the extent that the DBMS and associated drivers permit it. Both SQL Server 2000 and Oracle 10g provide excellent stored procedure support on Windows; other systems may have restrictions.

- **Improved administration.** New features facilitating LISTSERV administration include the SERVE OFF DROP command (invaluable when responding to mail-bombing or other DoS attacks), the SERVE LIST command, the new SITE.MAILTPL, the automatic PERMVARS.FILE cleanup, the automatic background list home page rebuild, and more.

- **Performance enhancement.** Version 14.3 includes significant performance enhancements to the web interface, and in particular the Subscriber's Corner, for sites with thousands of lists, as well as the TUNE_MANY_LISTS configuration option for LISTSERV-HPO, which can dramatically improve performance for sites with tens of thousands of lists.

- **Web interface accessibility.** The Web interface templates and code have been reviewed and modified extensively to follow the accessibility guidelines recommended by

W3C, in an effort to satisfy at least the Priority 1 “checkpoints”. While it is impossible to do exhaustive testing of every possible page generated by WA, over 100 WA-generated web pages – covering over 200 web page templates – passed both automated and manual accessibility checks. Three new “themes” (user-selected display schemes) have been introduced for improved accessibility.

- **F-Secure version certification:** F-Secure Anti-Virus versions up to 5.50 on Windows and up to 4.61 on Linux are now certified for use with LISTSERV 14.3.

CONTENTS

Changes that are new in LISTSERV 14.3:

Changes that are included in LISTSERV 14.3 but were originally introduced in version 14.2:

Also please note:

OS support information (important)
Applying LISTSERV 14.3

SECURITY: Passwords in log file are now suppressed

To prevent unintentional password disclosure when e-mailing logs for troubleshooting purposes, LISTSERV now suppresses all passwords in its logs. Passwords are replaced with the text "[redacted]". This applies not only to the `PW=` keyword, but to the `PW` and `PWC` commands, and the internal `X-PWADD` command used by the web interface.

As it is recognized that sometimes passwords need to be seen for debugging purposes, especially on development servers, a debugging option has been added to override this new behavior. This option, set in the site configuration file, is called DEBUG_LOG_PW and it defaults to 0.

Examples:

VM:	`DEBUG_LOG_PW = 1`
VMS:	`DEBUG_LOG_PW "1"`
unix:	`DEBUG_LOG_PW=1` `export DEBUG_LOG_PW`
Win:	`DEBUG_LOG_PW=1`

ANTI-VIRUS: [Linux x86] F-Secure Anti-Virus 4.61 now supported

F-Secure Anti-Virus is now certified up to and including version 4.61 for use with LISTSERV 14.3. Customers can obtain an installation key for FSAV 4.61 from their sales representative. To ensure optimal virus detection, this update is recommended, but not mandatory. Because of known issues with versions of FSAV older than 4.50, L-Soft strongly recommends an update to at least FSAV 4.50. These issues are:

- Virus database updates for the old 4.1x engine are not released as frequently as for current versions, which can trigger the anti-virus protection warning – on occasion, more than five days can lapse without any updates, even though updates are available for versions 4.50 and higher. This also means that you are not protected as well as you could be.
- Because they do not use a daemon, older versions have to initialize their virus engine for every scan, and it can take up to 10 seconds to scan a message with multiple attachments. FSAV 4.50 and higher have *much* better performance.

The FSAV for Linux Servers 4.61 kit may be downloaded from ftp://ftp.lsoft.com/f-secure/fsav-srv-4.61.3215-lsoft.tar.gz .

Before installing or upgrading, you may want to review L-Soft's Installing F-Secure Anti-Virus document, as well as the LISTSERV/F-Secure FAQ .

ANTI-SPAM: Non-member messages to list can require confirmation

Starting with LISTSERV 14.3, list owners can require that non-subscribers actively confirm their messages to the list, while allowing subscribers to post without confirmation. This can dramatically cut down on spam for lists accepting postings from non-subscribers. To activate this feature, you would set:

```
* Send= Public,Confirm,Non-Member
```

or either

```
* Send= Editor,Confirm,Non-Member
```

or

```
* Send= Editor,Hold,Confirm,Non-Member
```

in the list header.

The intent is to help list owners cut down spam on public discussion lists, without inconveniencing normal list subscribers. For public lists, it works like "Send= Public,Confirm" if you are not a member of the list, otherwise it works as "Send= Public" (no confirmation required from subscribed users). List owners and editors are considered to be members of the list even if they are not subscribed to it, and are thus not subjected to the confirmation requirement.

For edited lists, the behavior is similar -- non-members must confirm their own postings before they are submitted to the editor for approval, whereas members' postings go directly to the editor for approval without the intermediary step. It should be noted that ",Confirm" still activates the anti-spoofing feature that already existed, which requires that the editor must approve his own postings.

You can alternatively request confirmation from all senders to moderated lists, regardless of subscription status, by specifying:

```
* Send= Editor,Confirm,All
```

or

```
* Send= Editor,Hold,Confirm,All
```

IMPORTANT: "Non-Member" or “All” cannot be used without "Confirm". For instance, setting "Send= Public,Non-Member" will not activate the feature.

ANTI-SPAM: Messages to -request addresses require sender confirmation

Starting with LISTSERV 14.3, all messages sent to the LISTSERV-REQUEST, ALL-REQUEST, or *listname*-request addresses require confirmation from the sender before they will be passed on to the appropriate person(s). This change has been made primarily to address the increasing amount of spam and virii being sent to -request addresses by automated processes.

The confirmation required is the usual "OK" confirmation and will contain a web link if the web interface is enabled. The confirmation looks like this:

```
To cut  down on spam,  the TEST list has  been configured to  request positive
confirmation  of messages  posted to  the TEST-request  address. You  must now
confirm that  the enclosed message  did originate from  you. To do  so, simply
reply to the present message and type "OK" (without the quotes) in the text of
your message,  or click on the  link below. If this  does not work, or  if the
message did NOT originate from you, contact the list owner for assistance.

To APPROVE the message:
```
http://listserv.example.com/cgi-bin/wa.exe?OK=31002E37&L=TEST

and is found in the new MSG_POSTING_CONFIRM_SENDER message template.

Note, however, that if "Loopcheck= NoSpam" or "Loopcheck= None" is set at the list level, this feature will be bypassed, as the feature is considered as part of LISTSERV's built-in spam filter.

ANTI-SPAM: Whitelist/blacklist functionality added

LISTSERV 14.3 includes a whitelist/blacklist system that can be used either on its own, or in conjunction with the new spam exit feature. This built-in whitelist/blacklist system is very efficient, which makes it advantageous to duplicate your spam filter's whitelist/blacklist rules so that the spam filter is bypassed for messages whose disposition can be determined simply on the basis of the origin or target address.

There are four separate lists (or in reality, site configuration variables that contain the lists), called:

```
SPAM_BLACKLIST_FROM
SPAM_BLACKLIST_TO
SPAM_WHITELIST_FROM
SPAM_WHITELIST_TO
```

The lists are defined in the site configuration file in the usual space-separated manner. A restart of LISTSERV is required after updating any of the lists, also as usual.

The FROM lists are applied to all the origin fields in the RFC822 header – `From:`, `Sender:`, `Resent-From:` and `Resent-Sender:` (note that the SMTP-level `MAIL FROM:` is not used).

The TO lists are applied to the various recipient fields in the RFC822 header. Please note that LISTSERV does not work at the SMTP transaction level and does not have access to the `RCPT TO:` field.

The listing system is based on a score that LISTSERV maintains as it goes through the four lists in sequence. If the final score is zero, the message is neither white- nor blacklisted, and processing continues normally (to the external spam filter, if one has been configured). If the final score is positive, the message is whitelisted and accepted, bypassing any additional tests, including the external spam filter. If negative, the message is immediately rejected. A negative final score is a conclusive determination that makes any further tests unnecessary.

Being whitelisted normally gives one point, and being blacklisted costs one point. This can be changed by using the following syntax:

```
SPAM_WHITELIST_FROM=*@EXAMPLE.COM *@*.EXAMPLE.COM
SPAM_BLACKLIST_FROM=*@PUBLIC.EXAMPLE.COM >JAMES@EXAMPLE.COM
```

The first entry whitelists all EXAMPLE.COM addresses. The second entry acts as a cancellation of this whitelist for the fictitious machine PUBLIC.EXAMPLE.COM, which is not part of the Intranet. It also blacklists JAMES@EXAMPLE.COM, a notorious source of spam, with a score of 2 (every '>' sign gives another score point). JAMES@EXAMPLE.COM receives 1 point from the whitelist and -2 from the blacklist, so he will be effectively blacklisted. It is possible to load an entry with up to 254 extra score points, although it is not expected that anyone would need to go that far.

Every list gives score points at most once. So if we also had JAMES@EXAMPLE.COM and JAMES@* in the whitelist, he would still get only one point from the whitelist. But, when

applicable, you do get the highest possible number of points that you have matched. If we had JAMES@EXAMPLE.COM and >>JAMES@* in the whitelist, he would get 3 points.

Again, all the whitelists and blacklists scores are added. If you use both FROM and TO, you need to use a point system that gives the desired results. It is much easier if you only use FROM or only TO.

What you would put in TO is defunct addresses (former employees, “honeypots”, *etc.*) that are guaranteed to have come out of spam listings. The message is then bounced for all recipients, not just the defunct address.

In most cases you will not want bounces to be blacklisted, since they are useful to LISTSERV and are processed automatically. So in addition to the new white/blacklists themselves, a new site configuration variable, SPAM_WHITELIST_BOUNCE, has been added. This value is an integer, defaulting to 1, and is the number to be added to the message's whitelist score if it is a bounce. Set to 0 to disable. You can also use a higher value.

Note that bounces may not be run through the spam filter at all. If LISTSERV can immediately determine what the bounce is for and process it, it will not scan it for spam.

SYNTAX: The syntax for all of the white- and blacklists is identical, so we will use just one of them for OS-specific examples.

White/Blacklisting examples:

VM:	`SPAM_BLACKLIST_FROM = '*@PUBLIC.EXAMPLE.COM >JAMES@EXAMPLE.COM'`
VMS:	`SPAM_BLACKLIST_FROM "*@PUBLIC.EXAMPLE.COM >JAMES@EXAMPLE.COM"`
unix:	`SPAM_BLACKLIST_FROM=*@PUBLIC.EXAMPLE.COM >JAMES@EXAMPLE.COM` `export SPAM_BLACKLIST_FROM`
Win:	`SPAM_BLACKLIST_FROM=*@PUBLIC.EXAMPLE.COM >JAMES@EXAMPLE.COM`

SPAM_WHITELIST_BOUNCE examples:

VM:	`SPAM_WHITELIST_BOUNCE = 1`
VMS:	`SPAM_WHITELIST_BOUNCE "1"`
unix:	`SPAM_WHITELIST_BOUNCE=1` `export SPAM_WHITELIST_BOUNCE`
Win:	`SPAM_WHITELIST_BOUNCE=1`

UPDATE, 10/12/2005:

Unfortunately the original 14.3 release notes left out a very important step for enabling this new feature.

In order for LISTSERV to use the blacklists and whitelists, SPAM_EXIT (see the next section) must also be enabled and pointed to an existing, external exit program. This is necessary because the white- and blacklisting feature is part of LISTSERV's overall anti-spam toolbox, which is only activated if SPAM_EXIT is enabled.

While you may of course use the exit program you write to submit inbound mail to an external spam filter for scanning, that is not mandatory. If you are not submitting the inbound mail to an external spam filter, the exit program does not need to do anything other than exit immediately with a return code of zero. For example:

Windows:	site.cfg setting: SPAM_EXIT=SAEXIT

	x:\listserv\main\saexit.cmd : **rem don't do anything, just go back to LISTSERV** **exit 0**
Unix:	go.user setting: SPAM_EXIT="SAEXIT" export SPAM_EXIT ~listserv/home/SAEXIT : **# don't do anything, just go back to LISTSERV** **exit 0**

ANTI-SPAM: New SPAM_EXIT exit point to "hook" into third-party spam filter

Version 14.3 includes a LISTSERV "exit point" that allows you to use a third-party spam filter to scan messages processed by LISTSERV. Although this hook can in principle be used with any spam scanning product, all the examples and step-by-step instructions in this section will relate to *SpamAssassin*, a popular freeware spam filter that can be downloaded from http://spamassassin.apache.org . Please note that L-Soft did not author *SpamAssassin* and is unable to correct problems with the *SpamAssassin* product itself. L-Soft does not make any legal representations or warranties about *SpamAssassin.* Although L-Soft's support department will be glad to answer questions about the integration of *SpamAssassin* and LISTSERV, we cannot answer questions about *SpamAssassin* itself.

Overview

To enable spam filtering in LISTSERV, you must install the third-party spam filter, provide a script that will scan messages using the third-party filter (if using *SpamAssassin*, you can use one of the L-Soft supplied scripts), and activate this script by making changes to the LISTSERV configuration. LISTSERV will then scan every message it processes, with a few exceptions, and reject messages identified as spam.

Optionally, you can also activate LISTSERV 14.3's built-in blacklist/whitelist functionality, which is described in a different section of this document. This feature provides an additional level of spam filtering and can also improve performance significantly, because spam filters like *SpamAssassin* can take up to 5-20 seconds to scan a message.

Step-by-step instructions

This section contains step-by-step instructions for configuring LISTSERV to use SpamAssassin using one of the L-Soft supplied scripts. Throughout this section, we will make the following assumptions:

- *SpamAssassin* has already been installed and configured on a server that we will call `spamd.example.com`. This can, but does not have to be, the machine on which LISTSERV is installed. In particular, you can run LISTSERV on Windows and *SpamAssassin* on unix, and vice-versa.
- `spamd` has been started and is configured to accept incoming requests from the machine on which LISTSERV is installed.
- You have a test message file at your disposal to verify the operation of `spamc/spamd`. We will call this file `testmsg.txt`.

Step 1 of 4: Install and test *SpamAssassin* client.

Unix: Compile `spamc` on the LISTSERV host, then copy it to a directory in LISTSERV's path. To test it, use a command similar to the following:

```
$ spamc -c -d spamd.example.com < testmsg.txt
3.8/5.0
```

The flags you need to use may vary depending on your version of *SpamAssassin* and configuration. The response must be two numbers as shown above, but the numbers can be different than in the example (they are the *SpamAssassin* score of the test message). Any other response indicates an error. Refer to the `spamc` and `spamd` man pages for more information.

Windows: Download and install the spamc.exe executable from L-Soft, and place it in a directory in LISTSERV's path – for instance, the `LISTSERV\MAIN` directory.

To test the client, issue the following command:

```
C:\> spamc -c -d spamd.example.com < testmsg.txt
3.8/5.0
```

The response must be two numbers as shown above, but the numbers can be different than in the example (they are the *SpamAssassin* score of the test message). Any other response indicates an error. In that case, make sure that `spamd` is configured to allow connections from the LISTSERV host.

Step 2 of 4: Install perl or REXX (if not already available).

Unix: Install perl on the LISTSERV host, if not already installed.

Windows: Install a REXX interpreter, such as Regina REXX (http://regina-rexx.sourceforge.net/, Windows kit download available at http://prdownloads.sourceforge.net/regina-rexx/regina33.exe?download). When prompted to register .REXX as a path extension, you should do so. Alternatively, you can simply download REXX.EXE from L-Soft and place it in the same directory where you saved `spamc.exe`.

Step 3 of 4: Install and configure `SAEXIT` script.

- Download the L-Soft supplied sample script at one of the following URLs:

  Unix: ftp://ftp.lsoft.com/LISTSERV/UNIX/CONTRIB/SAEXIT.PL
  Windows: ftp://ftp.lsoft.com/LISTSERV/Windows/CONTRIB/saexit.rexx

- Edit the script to configure, at a minimum, the address of your *SpamAssassin* server. You may also want to change the other parameters.
- Make any other changes that you deem appropriate.
- Save the script in LISTSERV's main directory (on unix, set execute permissions):

  Unix: `~listserv/home`
  Windows: `C:\LISTSERV\MAIN`

- You can call the script anything you want, but in this example we will assume that you have left the name unchanged (`SAEXIT`).
- **Windows:** if you have not registered `.REXX` as a path extension when installing it in step 2 or if you downloaded it from the L-Soft ftp site instead of installing the full kit, you will need to create a script called `saexit.cmd` in the `C:\LISTSERV\MAIN` directory containing the following three lines:

  ```
  @echo off
  ```

```
rexx saexit.rexx %*
exit %ERRORLEVEL%
```

Step 4 of 4: Enable the `saexit` script.

To enable the script, add the following lines to LISTSERV's configuration:

Unix:
```
SPAM_EXIT="saexit"
export SPAM_EXIT
```

Windows:
```
SPAM_EXIT=SAEXIT
```

Restart LISTSERV to make the change take effect, then mail a spam message to a test list to confirm that everything is working as it should.

Restrictions

- The spam exit is a feature of LISTSERV Classic and HPO, and is not available with LISTSERV Lite. Maintenance is required.
- If you are using L-Soft's Anti-Virus Station (AVS) to provide virus protection to a server for which F-Secure Anti-Virus is not available, the spam exit must be installed on the AVS server, not on the primary LISTSERV server. This is because message scanning is bypassed on a server that uses the AVS for this purpose.
- The spam exit is implemented within LISTSERV's message scanner, which is informally known as the "virus scanner," because this was its original purpose. If the message scanner is disabled, for instance by setting the ANTI_VIRUS configuration parameter to 0, or by failing to install the maintenance LAK, both virus scanning and spam scanning are disabled. If ANTI_VIRUS is unset, LISTSERV will enable the message scanner if either virus scanning or spam scanning is configured and available.

VM and VMS

The spam exit is also available on VM and VMS, but there is no version of `spamc` for these systems. Since VM and VMS hosts normally use the AVS for message scanning, spam protection can be provided by simply installing the spam exit on the AVS server.

Advanced configuration

At the list level, "Misc-Options= NO_SPAM_CHECK" can be used to disable spam scans for a particular list and its associated `xxx-request` address.

New statistical counters have been added for spam scans. They work just the same way as the virus counters.

A new configuration parameter, `SPAM_MAXSIZE`, can be used to automatically accept messages larger than a certain size. The rationale is that spam filters can take minutes to process very large messages, whereas spam messages are almost always very small. The following example sets the threshold to 512k:

VM:	`SPAM_MAXSIZE = 512`
VMS:	`SPAM_MAXSIZE "512"`
unix:	`SPAM_MAXSIZE=512` `export SPAM_MAXSIZE`
Win:	`SPAM_MAXSIZE=512`

The default value is 256k. If set to 0, all messages will be scanned, which again could take considerable time. Messages that are not scanned do not count as a spam scan in the LISTSERV statistics.

Formal technical documentation of the SPAM_EXIT exit point

SPAM_EXIT is a standard LISTSERV exit, with all that this entails. The same OS-specific naming requirements used for regular LISTSERV exit points are enforced for the SPAM_EXIT exit point. See chapter 5.1 of the Developer's Guide to LISTSERV for more information.

LISTSERV scans messages in the following sequence:

```
Virus scan -> SPAM_MAXSIZE test -> whitelist/blacklist -> SPAM_EXIT ->
future L-Soft supplied tests
```

The rationale for doing things in this order is that viruses are far more dangerous than spam, so LISTSERV wants to identify them as quickly as possible, and in particular before any whitelist rule has had the opportunity to accept them. Besides, virus scans are much faster than spam scans.

The exit is formally defined as follows:

Name:	**SPAM_EXIT**
Parameters:	`SCAN [listname] | REPORT`
Return code:	0=Accept, 1=Local whitelist, 2=Reject

Its primary input is the file `spam.tmp` in the D directory (typically, unix, `~listserv/tmp` ; Windows, `\LISTSERV\TMP` ; OpenVMS, `LISTSERV_ROOT:[TMP]`). This contains a copy of the whole message, header and body.

When using the SCAN parameter, the exit returns:

- 0: continue normally, per the LISTSERV exit standard. Currently, this means the message is always accepted in practice, but future L-Soft supplied tests would run.
- 1: local whitelist. Accept the message; do NOT run any further tests.
- 2: reject the message. The exit string must then contain an error message to be reported to the sender. LISTSERV will use a standard message if no exit string is supplied, but this standard message is vague since LISTSERV does not know what the exit does.

When using this exit, it is very important to test things carefully, since a mistake could mean that every message is rejected. If for instance the script is not found by the operating system due to a misspelling, and the operating system happens to return 2 in that case, then the message will be automatically rejected even though the message was never scanned. (Because Windows returns 1 for a misspelled exit name, we chose 2=reject instead of the usual 1=reject.)

Once configured, spam scans take place whenever a virus scan takes place and no virus was detected, with two exceptions:

- When downloading binary attachments via WA (virus scan only – spam filters are unlikely to do something meaningful with a .EXE file)
- For the `DISTRIBUTE AV=YES` programming interface.

The minimum implementation for the `REPORT` call is to do the same as `SCAN` does. In addition, it is desirable to create an output file called `spam.report` in the same directory where the input file `spam.tmp` is located. There is no special format for this output file, but it is a good idea to

start with a line saying whether or not the message was identified as spam, and give the score if the spam filter uses a score system. LISTSERV does not process the report, it just ends up being shown to a human. If no `spam.report` file is created by the exit, LISTSERV will use the exit string as a one-line report. If there is no exit string, LISTSERV will generate a hard-coded message.

ANTI-SPAM: Suppress virus and spam bounces

Two new Boolean site configuration variables have been added, BOUNCE_VIRUS and BOUNCE_SPAM, both defaulting to 0. If set to 0, messages containing suspected viruses or spam are just dropped without being returned to sender. If set to 1, they are bounced/rejected as before, except that viruses to -request addresses are always dropped (also as before).

There is a trade-off to be made regarding BOUNCE_SPAM. The default setting of 0 will cause false positives from the spam filter to result in valid messages being dropped without any notification to the sender. Setting it to 1 may cause spoofed e-mail messages to be bounced to innocent third parties; these bounces are perceived by some people as spam and can result in your site being incorrectly blacklisted as a source of spam (particularly if the spoofed From: address happens to be that of an anti-spam “honeypot”).

The trade-off for BOUNCE_VIRUS is much less: false positives are virtually non-existent and most of the recent e-mail borne viruses use spoofed From: addresses, so there is little value in warning the purported sender that they have a virus.

Examples:

VM:	`BOUNCE_SPAM = 1`
VMS:	`BOUNCE_SPAM "1"`
unix:	`BOUNCE_SPAM=1` `export BOUNCE_SPAM`
Win:	`BOUNCE_SPAM=1`

ANTI-SPAM: Suppress Approved-By: header

Starting with 14.3, it is possible to suppress the RFC822 "Approved-By:" headers that would normally be generated by LISTSERV in messages posted through moderated lists by setting "Misc-Options= SUPPRESS_APPROVED_BY" in the list header of the affected list(s). This can only be done at the list level, there is no global setting.

USABILITY: Significant mail template changes

Significant changes have been made both to the mail template processor and to the default mail templates themselves for the version 14.3 release. Many -- *but not all* -- LISTSERV system messages that were previously hard-coded into the executable are now eligible for modification (for instance, to translate into national languages).

It is now possible to define site-wide defaults for all template forms without having to edit the DEFAULT MAILTPL file, which is not upgrade-safe. Any customizations made at the site level to the default templates now go into the site-level SITE MAILTPL file, which will not be disturbed during an upgrade. Defining site-wide defaults in the old $SITE$ MAILTPL file, which in any case did not work for all templates and was never intended for this purpose to begin with, is now

deprecated. In order to give sites that use $SITE$ MAILTPL for this purpose time to migrate to the new system, support for such usage will not be removed until the release of LISTSERV 15.

During startup, LISTSERV 14.3 will migrate existing templates from WWW_ARCHIVE MAILTPL to SITE MAILTPL, unless one of the following conditions arises:

- A conflicting template (same name, different contents) is already in SITE MAILTPL
- The copy of the template in WWW_ARCHIVE matches the first template in the system search order.

If there are no conflicts, WWW_ARCHIVE MAILTPL is then renamed to WWW_ARCHIVE OLDTPL. If a conflict is detected, LISTSERV will not attempt to determine which version of the template is "correct", but rather will log something like the following:

```
7 Oct 2004 10:57:05 Migrating templates from WWW_ARCHIVE to SITE...
- $TEST_TEMPLATE: conflicting version found in SITE
7 Oct 2004 10:57:05 Conflicts detected, finish migration manually
```

It is then incumbent on the site maintainer to harmonize the differing versions of any templates reported to be in conflict.

Because the changes are so extensive, they are not documented in these release notes, but are documented in a completely new comprehensive mail template module which is available on the L-Soft web site.

USABILITY: Mail templates: .else and .quif directives added

An .ELSE directive has been added to LISTSERV's .BB conditional processor for mail templates, making it easier to write complex conditionals. Example:

```
Your message dated &ORGDATE with
.bb &SUBJECT ^= ''
subject "&SUBJECT;"
.else
no subject
.eb
```

There is also a new .QUIF directive:

```
.quif <expression>
```

This is equivalent to:

```
.bb <expression>
.qu
.eb
```

USABILITY: Command to list "served out" users

Starting with LISTSERV 14.3, it is possible to list those users who have been "served out". The four categories of such users -- those who have been served out automatically, those who have been served out manually by the LISTSERV maintainer, those who have been served out manually with the DROP option by the LISTSERV maintainer, and those who have been served out automatically as spammers -- will all be displayed.

The command is simply:

SERVE LIST

and must be issued by a LISTSERV maintainer (a password is also required if the command is sent by email). The output is similar to the following:

```
> serve list
JOE@EXAMPLE.COM                      DROP 2003-08-20 15:51:20 by
nathan@EXAMPLE.COM

FOOBAR@EXAMPLE.EDU                   HARD 2003-04-07 14:55:29 by
NATHAN@EXAMPLE.COM

BLAB@FOO.EXAMPLE.COM                 SOFT 2004-09-14 10:53:18

SPAMMER@SPAMDOMAIN.COM               SPAM 2003-08-20 15:50:55

4 matching entries.
```

Entries are sorted by category (DROP, HARD, SOFT, SPAM) and then by date, from most to least recent.

When a SOFT serve has no date associated with it (as it would if it happened prior to LISTSERV 14.3), today's date is recorded and the record kept internally by LISTSERV is updated. New SOFT serves have their date recorded from the onset.

USABILITY: Command to lift spam quarantine

LISTSERV maintainers may now lift the 48 hour "spam quarantine" for specified users (those who are served out with a severity level of SPAM in the SERVE LIST output) by issuing a SERVE command for that user. If the user is under quarantine, the normal "service has been restored" message is issued, along with "Spam quarantine lifted for <address>."

USABILITY: New MAX_CONSECUTIVE_SUBS site configuration variable

In previous releases, as a preventative against spoofers adding third parties to hundreds of lists without their knowledge, LISTSERV has had a hard-coded maximum for the number of local lists to which a user could subscribe at any one given time. The limit was set at 50, after which LISTSERV would assume that the subscription requests were coming from a spoofer and would cancel the last 50 subscription requests for the user in question. (To clarify, a user could be subscribed to more than 50 lists on the server, but could not issue more than 50 subscription requests in a row.)

The new site configuration variable MAX_CONSECUTIVE_SUBS allows site maintainers full control over the limit. The default remains 50. A setting of 0 disables the anti-spoofing filter (which is not recommended). OS-specific examples are as follows:

Examples:

VM:	`MAX_CONSECUTIVE_SUBS = 25`
VMS:	`MAX_CONSECUTIVE_SUBS "25"`
unix:	`MAX_CONSECUTIVE_SUBS=25` `export MAX_CONSECUTIVE_SUBS`

Win:	MAX_CONSECUTIVE_SUBS=25

USABILITY: DBMS stored procedures now supported

LISTSERV 14.3 supports the use of stored procedures to the extent that the DBMS and its associated drivers permit it.

For SQL Server, it is important to note that procedure calls MUST use the EXECUTE (or EXEC) command verb.

USABILITY: (Non-VM) Ability to send pre-approved messages

It is sometimes desirable to send mail to a LISTSERV list with Send=Editor,Confirm, Send=Owner,Confirm, or Send=*address*,Confirm, and have it go out immediately, rather than wait for the editor to confirm it. Thus, there needs to be a mechanism for password-confirming a posting to a list as a substitute for the e-mail confirmation handshake.

To provide this mechanism, a new DISTRIBUTE POST command has been added. The syntax is based on the existing DISTRIBUTE MAIL command. The differences are:

- The verb is POST instead of MAIL. Data and recipient are supplied as usual. For example, “TO xxx”, “TO DD=xxx”, or implied DD.

- Only one recipient (the address of the list to which you are posting) is allowed in the TO DD; anything else is will generate an error message. Automated scripts should guard against supplying more than one recipient.

- Although an FQDN is required in the TO DD, the hostname of this recipient is assumed to be an alias for the local host, and is ignored. The local-part must be a valid, existing list. An error message will be generated if this is not the case.

- The new PRE-APPROVED=NO|YES option becomes available to DISTRIBUTE POST. The default is NO and it then works like a DISTRIBUTE MAIL job with the list as sole recipient, except that it is more efficient. You must be a list owner to use PRE-APPROVED=YES. PRE-APPROVED=YES cannot be used with DISTRIBUTE MAIL and will generate an error message if so used. If the list is moderated, and the sender address is not an editor, the e-mail sent in this way *will* be submitted for moderation. To clarify: this replaces *confirmation* not moderation approval. Make sure that the poster's address (in the 'From:' line) is authorized to post to the list, or the message may be rejected or forwarded to the moderator. PRE-APPROVED=YES authenticates the origin of the message, but does not bypass the normal authorization steps.

- DISTRIBUTE POST is not available under VM.

- Distribution options are ignored, since DISTRIBUTE does not deliver the message at this time but passes it on to the list posting mechanism.

It is important to understand that DISTRIBUTE POST is just a submission system. The posting may be processed immediately, it may be delayed until the time specified in the list header, it may be sent to the moderator if the owner is not allowed to post to the list (perhaps not very common, but there are such lists), the list could be on hold, a virus could be detected (if you did not use AV=YES), etc.

Following is a list of the DISTRIBUTE options supported or partially supported with DISTRIBUTE POST.

Full support:

- CANON=, AV=
- The new PRE-APPROVED=

Partial support:

- DD=: TO DD must refer to a list.
- DEBUG=YES: the job will be executed with all verifications, but in the end the message will not be posted (this is the primary purpose of DEBUG=YES). The secondary purpose, to receive a report showing the intended message routing, number of recipients and so on, is not supported since DISTRIBUTE POST always sends 1 local message and forwards 0 jobs to other servers.

Other options are ignored as long as the syntax is correct. If there is a syntax error, the job fails.

USABILITY: (Non-VM) New debug setting for logged TCPGUI commands

It is now possible to show fully-logged TCPGUI commands for debugging purposes. By default, LISTSERV truncates overly long command lines to make logs less cumbersome. A new site configuration file variable called DEBUG_LOG_TCPGUI is available, defaulting to 0.

Examples:

VMS:	`DEBUG_LOG_TCPGUI "1"`
unix:	`DEBUG_LOG_TCPGUI=1` `export DEBUG_LOG_TCPGUI`
Win:	`DEBUG_LOG_TCPGUI=1`

When set to 1, all commands starting with 'X-' are logged with no size limit.

Enabling TCPGUI debugging makes logs very large. Therefore L-Soft does not recommend leaving this option enabled for long periods. It should be used for debugging only and disabled after troubleshooting has been completed.

USABILITY: Suppress tracebacks in error messages

It is now possible to suppress tracebacks in error messages by setting the Boolean site configuration variable HIDE_TRACEBACK. When set, the tracebacks are no longer included in error messages, except in certain postmaster-only e-mail messages.

This has no impact on what is written to the LISTSERV log. The traceback is always written to the LISTSERV log.

The default for HIDE_TRACEBACK is 0, or no suppression, that is, the original behavior.

Examples:

VM:	`HIDE_TRACEBACK = 1`
VMS:	`HIDE_TRACEBACK "1"`
unix:	`HIDE_TRACEBACK=1` `export HIDE_TRACEBACK`
Win:	`HIDE_TRACEBACK=1`

USABILITY: Specify maximum digest size in Kilobytes or Megabytes

The Digest= list header keyword has been enhanced in 14.3 to allow list owners to specify a maximum digest size in either kilobytes or megabytes, rather than in lines, which was the only way to do this in earlier versions. For instance:

* Digest= Yes,Same,Daily,Size(100K)	Cut the digest at 100Kbytes
* Digest= Yes,Same,Daily,Size(1M)	Cut the digest at 1Mbyte

As before, the digest will be cut whenever the pending digest grows over the size limit setting. Please remember that the limit refers to the cutoff point, not to the size of the largest digest that LISTSERV will send. Assuming Size(1M), a 1Mbyte message causing the digest to go over the limit will be included in the cut digest in its entirety, so it is to be expected that if the digest already contains nearly that much text, the cut digest will be almost 2MB when it is distributed. A judicious use of the Sizelim= keyword setting in conjunction with the Size() parameter for Digest= is therefore recommended in order to keep digests from either growing too large or being cut too often.

USABILITY: All postmaster commands authenticated with personal password

All LISTSERV postmaster-only commands can be authenticated with the postmaster's personal password. CREATEPW and STOREPW are now considered to be obsolete, and can be completely disabled by setting them to the special value *NOPW*, for instance,

VM:	`CREATEPW = *NOPW*`
VMS:	`CREATEPW "*NOPW*"`
unix:	`CREATEPW="*NOPW"` `export CREATEPW`
Win:	`CREATEPW=*NOPW*`

STOREPW would be disabled in the same way.

PERFORMANCE: Miscellaneous improvements

Where practical, routines have been tightened and performance improved. Specific improvements include:

For HPO only, reverse indexing of attachments has been improved by an enhancement that better recognizes Base64, uuencode, and MIME boundaries. This change is not retroactive; reindexing will be required if reverse indexing is already enabled (ie, DBRINDEX=1). To force reindexing, simply delete *`listname`*`.DBRINDEX` wherever found, and the reverse index will be rebuilt at the next search of the list's archives.

Also for HPO only, performance issues that arise when changelogs get very large have been addressed.

OTHER: New PLAIN_FROMLINE parameter affects mail from LISTSERV address

The site configuration parameter PLAIN_FROMLINE is a Boolean variable (set to 1 or zero) which controls whether or not LISTSERV generates a “plain” From: line when sending administrative mail, eg, setting this variable to "1" would result in

From: LISTSERV@LISTSERV.EXAMPLE.COM

rather than

```
From: "Example Company LISTSERV Server (14.3)"
<LISTSERV@LISTSERV.EXAMPLE.COM>
```

Examples:

VM:	`PLAIN_FROMLINE = 1`
VMS:	`PLAIN_FROMLINE "1"`
Unix:	`PLAIN_FROMLINE=1` `export PLAIN_FROMLINE`
Win:	`PLAIN_FROMLINE=1`

The default is 0, that is, the original behavior.

Enabling PLAIN_FROMLINE overrides any setting made to MYORG=, MYNAME=, and/or MYNAME_HIDE_RELEASE=, since the organization name setting and release number will no longer be displayed.

OTHER: New monthly limits and "booster" licensing for Maestro

Several licensing enhancements have been added that are Maestro-specific. These enhancements require LISTSERV 14.2 or later.

First, monthly limits have been introduced. Daily limits will be phased out.

Second, and for LAKs with monthly Maestro scope only, it is possible to obtain a "booster" LAK to temporarily raise your monthly limit should you hit that limit early in the month. More information on Maestro booster LAKs can be obtained from your sales representative.

Finally, starting with 14.3, if a LISTSERV LAK with a limited Maestro scope is installed, the output of SHOW LICENSE will include information on the number of deliveries left either for the day or for the month (depending on the licensing type). Here is sample output:

```
> show lic
License type:      Permanent
Expiration date:   31 Dec 2004
Maintenance until: 31 Dec 2004 (graduated license)
Capacity:          Unlimited
Maestro limits:    10000/month (5674 used)
Product options:   HPO
Scope:             MAESTRO(0/10000)
Version:           1.8e
```

```
Serial number:      TESTSERVER-1T
Build date:         24 Apr 2004
```

OTHER: Miscellaneous changes and fixes

The following miscellaneous fixes and changes are included in LISTSERV 14.3.

- Digests and indexes are now generated with RFC822 Message-ID: headers.

- On moderated lists, if ",Hold" is not specified in "Send=", it is now forced for multipart messages unless you use the NoMIME option on the Send= keyword.

- The output of the RELEASE command now indicates the build date of the software. (SHOW LICENSE will also continue to indicate the build date.)

- List-level change-logs now record VIRUS records showing each virus caught arriving via email to the list.

- LISTSERV now checks whether or not the sender is served out before it checks to see if the message exceeds Sizelim=.

  Prior to this release, if a user was served out and sent a message to a mailing list that exceeded the mailing list's Sizelim= setting, LISTSERV responded back to the user saying that the size limit had been exceeded. Then, if the user had an autoresponder set up (which is often the case with served out users) the autoresponder would respond back to LISTSERV, which in turn would send the "message received from a user who has been served out" notification to the LISTSERV maintainer. This change eliminates the extra back-and-forth traffic, and (perhaps more importantly) the annoying notifications to the LISTSERV maintainer, who is only being notified because the Sizelim= message comes from LISTSERV and not from the list itself.

- LISTSERV now conforms to RFC3463 and generates bounces in that format. It is possible to revert to the formerly used bounce format with the site configuration variable DEBUG_BOUNCE_OLDFORMAT=1.

- When a subscriber is auto-deleted from a list, the final error that caused the auto-deletion is now recorded on the AUTODEL record found in the list-level change-log for that subscriber. For instance,

  ```
  20040804000038 AUTODEL user@EXAMPLE.NET 5.1.1 No such local user
  ```

- You can now specify IP addresses in SMTP_FORWARD and SMTP_FORWARD_n site configuration variables; for instance, for Windows,

  ```
  SMTP_FORWARD=127.0.0.1
  SMTP_FORWARD_1=2*127.0.0.1
  ```

  While in some cases this has been possible (but not supported) on some operating systems, it is now formally supported.

- The default From: name for mail from the LISTSERV command address has been changed to “*myorg* LISTSERV Server (14.3)”, where *myorg* is the value specified in the site configuration for the MYORG variable if it is present or the NODE variable otherwise.

- Where possible, the default Subject: header for responses to commands sent to the LISTSERV command address is no longer “Output of job…”, but instead “Re: XXXX” where XXXX is the first recognized command in the job. It is also possible to use the new .SJ directive to specify a different subject in message templates defining the response to some commands (see the documentation of .SJ in the new mail template module).

- Job stats are now always suppressed on responses to the *listname*-LEAVE-REQUEST and the other special –REQUEST addresses. Subscriptions and signoffs via these addresses no longer generate *additional* command response e-mails (that is, in addition to the normal SIGNUP1 template and/or WELCOME or FAREWELL files).

- Mail-Merge=Yes can now be used in conjunction with the Sub-Lists= keyword. Previously, this combination was not supported.

- Style-sheets and comments in HTML messages to the LISTSERV command address are now ignored.

- You can now use unquoted semicolons in .BB conditionals. Previously one had to code

  .BB '&X;' = 1

  in order for LISTSERV accept the variable with the semicolon. The syntax

  .BB &X; = 1

  is now acceptable.

- The LISTSERV-generated RFC822 "Message-ID:" header has been extended to prevent any duplication.

- [Non-VM] A PUT of a KEYWORDS file now resets cached list header information (previously a stop and restart of LISTSERV was required to re-cache the headers after the PUT).

- Problems with special character encoding in mail headers have been fixed. For example, Subject: lines with special characters preventing proper processing of subject-tags and topics.

- Various problems with quoted-printable encoding have been fixed, including a problem with &*TICKET_URL(); used in the bottom or top banner of lists with Mail-Merge=Yes.

- A variety of other small defects were fixed. If you are experiencing any difficulties with an older version, please upgrade.

WA: Web interface enhancements and fixes

- Many template and code changes to improve the accessibility of Web pages:
  - <LABEL> tags on form elements.
  - Row and column headers identified for data tables.
  - Three new “themes” (settings that allow each user to choose how the dynamic pages are displayed for them): Theme7 "Big Print" is the same as the default theme (Theme1 "Default Theme") except uses a much larger font size and increases the size of buttons; Theme8 "Default Font" does not specify any font

family or size, allowing the browser default to prevail (the other settings, such as colors and images, are the same as Theme1); Theme9 "No style sheet" does not define any style sheet: no background colors, no fonts, no custom list bullets, only bare HTML (this is the preferred theme for visually impaired users using a screen reader).
    - Ability for visually impaired users employing screen readers to skip repetitive navigation links.

- "Show All Lists" option on Subscriber's Corner has been improved. The site administrator can remove the option from the interface with a simple change to the REPORT-MAIN web template (see the comments at the top of this template). By default, “Show All Lists” will show all lists that are not Confidential=Yes (the same lists that are listed in index.html) and subscribed lists. A site administrator can override this default by creating a template called VISIBLE-LISTS with one list name per line. If this template exists, “Show All Lists” will only show the lists in that template. Note that “Show all lists” will always also include the lists to which the logged in account is subscribed and lists that the logged in account owns (which means that logged in postmaster accounts will always see all lists).

- Subscriber’s Corner no longer requires a login. However a login is still required to subscribe to any lists.

- The template editor now groups templates into categories to make it easier for site administrators and list owners to find templates. Templates that are not usually of use to list owners are hidden in the list template editor. It is now possible to edit the site-wide default mail templates from the Web interface.

- The list owner Wizard has been updated to support the new anti-spam keywords and new templates. The Wizard for the HTML Description includes a “View” option to preview the HTML description. The Send= keyword Wizard now provides an explanation of the current setting. Three new task-oriented Wizards are available for customizing the most important administrative messages and the list’s Web pages, and for quick access to a few predefined reports.

- The server management interface has been expanded to include direct links to certain reports, informational commands, and editing the most useful templates. The tutorials have been expanded to include information on site-wide customization of administrative mail messages and removing a mailing list.

- When viewing public archives, a login is required to see e-mail addresses, both in the mail headers and within the message. When viewed without a login, e-mail addresses are replaced with “[log in to unmask]”. New login and logout functions are available on both the individual message viewing and second-level (weekly/monthly/yearly) archive pages.

- List-based changelog report can now report on viruses sent to the list and neutralized by F-Secure.

- Anti-Virus Statistics and Server Accounting reports now include Anti-Spam statistics: number of spam scan operations performed, number of spam messages detected, and number of spam messages that would have been sent if they had not been neutralized. The counters for numbers of virus and spam messages stopped have been increased to 64 bits (32 bits were not sufficient to hold the number of virus messages stopped in one month at some sites).

- On selected pages, certain functions have been added for browsers with scripting enabled. The moderation page includes buttons for checking or unchecking all messages to assist with spam-heavy moderation loads. When “All Moderators” checkbox is selected, list of messages indicates whether the message was sent to the logged-in user for moderation and/or others (useful in round-robin moderation situations). The list selection pull-down menu automatically issues a “submit” when a list is selected. The template editor selection page automatically issues a submit when a different template category is selected. If scripting is disabled in the browser, users still need to press “submit”/”refresh” or check all moderation boxes one at a time.

- LISTSERV will rebuild all lists' static pages automatically whenever the site-wide WWW_INDEX template form is updated, or whenever a template form that is included (with the .IM directive) in the site-wide WWW_INDEX is updated.

- It is now possible to have a *listname*.html web page even if Notebook=No. This is done automatically when creating a new list in the server management interface when the “Create directory for list archives” checkbox is checked. To add this to existing lists, the site administrator must create a subdirectory for the list in the web archives directory.

- When a list is created from the web interface with digests enabled, but without archives, that is, with something like

  ```
  * Notebook= No
  * Digest= Yes,/home/listserv/lists/mylist-l,Daily
  ```

  the digest directory is now created automatically, when “Create directory for list archives” is selected.
- Performance enhancements on reporting and other pages. Logging out and back in once to change your login cookie is necessary to enable one of them.

- Numerous fixes were made to prevent cross-site scripting (CSS) vulnerabilities.

- A variety of changes were made to the archive search page, including a fix to the “more hits” link.

- A variety of other small defects were fixed. If you are experiencing any difficulties with an older version, please upgrade.

Note that a large number of the Web templates have changed for this version, and so re-customization is recommended in order to get Web-based access to new features, including the accessibility features. However, if the WIZARD_KWSEND_ENTRY and TPLMGR-FORMSEL templates have been modified for your site, they require extra steps to remain backwards compatible. Except for these, your 1.8e templates should work, albeit missing the new functionality.

- If you have modified the WIZARD_KWSEND_ENTRY Web template, you should either re-do your customizations or at a minimum make the following changes to your customizations:

  - Remove all the “(specify)” options from the first pull-down selection menu – they are no longer supported in this menu
  - Replace the references to `&+SEND_1_TEXT_xxx;` with `&+SEND_7_TEXT_xxx;`

- Add the following code preceding the first reference to `&+SEND_7_TEXT_TDEL;`:
  `<input type="hidden" name="0" value="&+SEND_7_TEXT_VAL;">`
- Change the example to include parentheses, for example "(OTHERLIST)".
- Change the WIZARD_KWSEND_HELP2 template to include parentheses.

Note that these minimum changes will only allow the older template to keep working, they will not enable the new functionality, so they should be used as a temporary measure until the new template can be re-customized or re-translated.

- If you have modified the TPLMGR-FORMSEL, the new template categorization feature will cause TPLMGR-FORMSEL to show only the templates that were categorized as "Most Useful Templates". Either re-do your customizations for this template, or add the new code for selecting the template category to the modified template, or change the first line of the TPL-CATEGORIES-WWW and TPL-CATEGORIES templates to say "+SE DEFAULTCATEGORY ALL" (i.e. replace "TOP10" with "ALL"). To open one of these templates in the Web interface template editor (since they are not in the default category), use the wa parameters: "TPLED2=SITE&W=1&a=templatename", for example:
  "http://listserv.example.com/scripts/wa.exe?TPLED2=SITE&W=1&a=TPL-CATEGORIES-WWW".

The "wa" CGI executable compatible with LISTSERV 14.3 is version 2.3.30 or later. To determine what version of "wa" you currently have running, simply invoke "wa" with the parameter `?DEBUG-SHOW-VERSION`. For instance,

Unix:
http://yourserver/path-to-wa/wa?DEBUG-SHOW-VERSION

Windows, OpenVMS:
http://yourserver/path-to-wa/wa.exe?DEBUG-SHOW-VERSION

VERSION 14.2 CHANGES

SECURITY (14.2): [Unix] Setting umask/read permissions for web interface

In previous versions, LISTSERV has written files to its web interface directory with whatever default umask was set by the server administrator for the `listserv` user, under the premise that the server administrator knew best what security policy he or she wished to impose. If the local security policy was to create world-readable files by default, this left LISTSERV's web index files (listname.ind*) world-readable. Although these index files do not contain actual messages, they do contain e-mail addresses and can be used by spammers for address harvesting.

In version 14.2 and following, LISTSERV enforces a default umask of `0g7`, that is: owner – full access, group – as set by the administrator, world – no access. Index files are written without any world access, closing this potential exposure. The same applies to wwwtpl files.

After upgrading LISTSERV, it may be necessary to delete all files from the `archives` directory tree and restart LISTSERV to rebuild with the new permissions. Alternatively, you can set the permissions manually with `chmod -R` after the upgrade, with the understanding that LISTSERV will subsequently set permissions per the above for all new web interface files it creates.

SECURITY (14.2): [Windows] Protecting web index files (IIS)

In previous versions, LISTSERV created web index files (listname.ind*) with the access permissions specified by the directory in which the files were created. If these permissions were set too generously, LISTSERV's web index files could be accessed over the web. Although these index files do not contain actual messages, they do contain e-mail addresses and can be used by spammers for address harvesting.

Starting with LISTSERV 14.2, it is possible to protect the index files from browsing. One fundamental obstacle with IIS is that, by default, basic web server functions (file serving, etc.) and CGI scripts run under the same account, `IUSR_xxxx`. With this setup, it is impossible to protect the index files from browsing without also preventing WA from accessing them, in which case the web interface stops working. The solution is to use two separate accounts.

Let's assume that:

- You are running IIS and the IIS account is called `IUSR_WWW`
- You are starting from a working WA/IIS setup and want to make it more secure
- You have sensible protections on the web directory tree (in particular, you haven't given the Everyone user full access).

Step 1 of 4: Clone the `IUSR_WWW` user into, say, `IUSR_LISTSERV`.

NOTE CAREFULLY: For servers running on domain members which are not domain controllers, you MUST create the `IUSR_LISTSERV` user as a domain user. This does not apply to servers running in a standalone environment or to domain controllers themselves – ONLY to servers running on domain members.

Step 2 of 4: Change the Windows file permissions on the scripts directory to give RX permissions to `IUSR_LISTSERV`, so that it can execute `WA.EXE`. If you do not have other scripts running out of that directory, you can remove the RX permission there for `IUSR_WWW`; otherwise you will probably want to leave the permissions of `IUSR_WWW` intact.

Step 3 of 4: Change the Windows file permissions on the `archives` directory tree and on the various locations of the list archive files such that `IUSR_LISTSERV` has read access. Remove all access to these files from `IUSR_WWW` except for "Traverse directories".

Step 4 of 4: In the Internet Services Manager, right-click on the `scripts` directory under the appropriate web site, and click "Properties". Choose the "Directory Security" tab, then in the box labeled "Anonymous access and authentication control", click "Edit" to bring up a pop-up window entitled "Authentication Methods". In that window, ensure that "Anonymous Access" is checked, and click "Edit" next to "Account used for anonymous access". Change the user from `IUSR_WWW` to `IUSR_LISTSERV`, and do not uncheck the box that says "Allow IIS to control password". Click "OK" until you get back to the main Internet Services Manager window.

At this point, WA is running as `IUSR_LISTSERV`. It has read access to everything in the `archives` directory tree, and to the list archive files, and everything should work as before. From WA's perspective, nothing has changed except its SID. Attempts to read random listname.ind* files from the list directories under `archives` should now require a login and password, which, assuming one is a random user without special access, will fail.

But what about regular HTML pages, like the list home pages? They will be accessed by `IUSR_WWW`, which has no access to the `archives` directory, except Traverse. The answer is a change in LISTSERV 14.2 that makes it add an ACL giving read access to Everyone to files that

need to be available for browsing. This is automatic – you do not need to set anything in the LISTSERV configuration.

After upgrading LISTSERV, it may be necessary to delete all the HTML files and images from the `archives` directory tree in order to let LISTSERV rebuild them and create the special ACL.

SECURITY (14.2): [Windows] Protecting web index files (Apache)

In order to protect web index files under Apache for Windows, you must enable (if it is not already enabled; the installed default is to disable) the use of `.htaccess` files to control at least the "Limit" directive, and manually place an `.htaccess` file in each list-level directory which contains the following:

```
<Limit GET POST OPTIONS PROPFIND>
        Order deny,allow
        Deny from all
</Limit>
```

This will deny access to the files in that directory via http, while still allowing WA.EXE to access them for its own purposes.

Please note carefully that LISTSERV will not change your Apache configuration nor will it write the `.htaccess` files for you. It is your responsibility to make the appropriate Apache configuration change and to secure each new web directory as it is made.

ANTI-VIRUS (14.2): [Windows] F-Secure Anti-Virus 5.50 now certified

F-Secure Anti-Virus is now certified up to and including version 5.50 for use with LISTSERV 14.2 and 14.3. Customers can obtain an installation key for FSAV 5.50 from their sales representative. To ensure optimal virus detection, this update is recommended, but not mandatory, except as listed below.

Mandatory update:

- Sites running Windows XP with an AMD64 processor MUST upgrade to at least FSAV 5.43 prior to installing SP2.
- Other sites running Windows XP MUST upgrade to at least FSAV 5.40. L-Soft no longer supports the special FSAV 5.31 release.
- Reminder: FSAV versions older than 5.30 are not supported.

Strongly recommended update:

- Sites running Windows 2003 SHOULD upgrade to FSAV 5.50.
- All sites SHOULD upgrade to at least FSAV 5.40 in order to ensure detection of “modern” viruses.

The FSAV 5.50 kit may be downloaded from ftp://ftp.lsoft.com/f-secure/550srv10260.zip .

Before installing or upgrading, you may want to review L-Soft's Installing F-Secure Anti-Virus document, as well as the LISTSERV/F-Secure FAQ .

USABILITY (14.2): New DROP argument for SERVE OFF command

It is now possible to serve off a problem user and "drop" further messages coming from that user rather than have them bounced to the postmaster with the message stating that mail has been received from a user who has been served off.

Simply append the argument "DROP" to the end of a SERVE OFF command, for instance:

SERVE userid@host OFF DROP

The QUIET modifier can also be prepended, as before:

QUIET SERVE userid@host OFF DROP

The response from LISTSERV will indicate that further messages from the user in question will be dropped silently:

```
JOE@EXAMPLE.COM has been permanently served out. Access can be restored only
by privileged users. Incoming messages  from JOE@EXAMPLE.COM will be dropped
silently.
```

USABILITY (14.2): New ALLOW-BOUNCES parameter for Loopcheck=

In LISTSERV 1.8d and previous, it was possible to create a list to collect error reports from another list (or lists) by setting "Loopcheck= None" for the list that was to collect the errors. This workaround no longer applies to LISTSERV 14.x (that is, 1.8e) and following.

In LISTSERV 14.2 and following, you may set "Loopcheck= None,Allow-Bounces" in the error-collecting list's header to solve this problem. (Note that setting "Loopcheck= Allow-Bounces" will generate a syntax error. "Allow-Bounces" must be set in conjunction with "None".)

USABILITY: New SPAM_ALERT site configuration variable

A new Boolean site configuration variable, SPAM_ALERT=, has been added beginning with LISTSERV 14.3. SPAM_ALERT defaults to 0, meaning that spam alerts will not be sent to the LISTSERV postmaster (but will still be sent back to the message poster for his/her information).

The previous default behavior, in which the LISTSERV postmaster was cc'd on all spam alerts, can be reverted to by setting SPAM_ALERT=1.

Examples:

VM:	`SPAM_ALERT = 0`
VMS:	`SPAM_ALERT "0"`
unix:	`SPAM_ALERT=0` `export SPAM_ALERT`
Win:	`SPAM_ALERT=0`

USABILITY (14.2): New SOFTBOUNCE changelog record for nolist changelogs

This new event has been added to the nolist changelogs as a counterpart to the BOUNCE record, which logs only permanent bounces (that is, those which are returned with a 5.x.x or 5xx DSN code signifying that they are permanent in nature).

Previously, LISTSERV discarded "nolist" reports that did not refer to "permanent" (5.x.x or 5xx) errors and reported out to the changelog file only those interpreted as being due to a truly "fatal" error (for instance, "no such user").

The SOFTBOUNCE record now provides full reporting for non-fatal bounces, giving the user full control over what to do about temporary errors that may be returned to the owner-nolist address rather than simply discarding them. For example (addresses x'ed out to protect the innocent; they will appear in full in a real changelog):

```
20030528162031 SOFTBOUNCE xxxxxxxxxxxxx@xxxxxxxxxx.COM 4.2.2 Unspecified;
usually "Mailbox full"
20030528162623 SOFTBOUNCE xxxxxxxxxxxxx@xxxxxxxxxx.COM 4.2.2 Disk quota
exceeded
20030529075124 SOFTBOUNCE xxxxx@xxxxx.EDU 5.0.0 550 error - no such recipient
20030529075730 SOFTBOUNCE xxxxx@xxxxx.EDU 5.3.4 552 Requested mail action
aborted: exceeded storage allocation
20030529080506 SOFTBOUNCE xxxxx@xxxxx.EDU 5.0.0 5.4.4 Unable to route
20030529080738 SOFTBOUNCE xxxxx@xxxxx.EDU 4.2.1 4.2.1 Mailbox disabled, not
accepting messages
20030724170230 SOFTBOUNCE XXXXXXX@XXXXXXXX.COM 2.0.0 250 OK
```

Note: A 5.x.x or 5xx error code is normally considered permanent, or "hard". In some cases, however, the standard is ambiguous; a so-called "permanent" error may be due to a transient problem, such as an Internet link being down, or a mail server being wedged. In other cases, the standard is questionable; should a full mailbox (5.3.4) really be considered "permanent", and thus, grounds for deleting the user? LISTSERV therefore counts these types of bounces as "soft", and lets the site administrator make the final decision.

Like the BOUNCE record, SOFTBOUNCE records apply only to nolist changelogs, and will not appear in regular list-based or system changelogs.

USABILITY: Change to .BB conditional processor (precedence)

Prior to LISTSERV 14.3, a complex conditional in a mail-merge job required judicious use of parenthesis. For instance, a conditional evaluating a PARTS variable for three distinct values required the following construction:

```
.BB ((John in &PARTS or Max in &PARTS) or Mary in &PARTS)
-do something interesting-
.EB
```

This construction would be true and something interesting would be done if &PARTS contained one or more of "John", "Max", or "Mary".

The syntax has been simplified, so that it is now possible to specify a .BB conditional line as follows:

.BB John in &PARTS or Max in &PARTS or Mary in &PARTS

without parenthesis to get the same results. There is no operator precedence. By definition, the parser gives precedence from left to right, that is, 'a or b or c' is '(a or b) or c'.

The original syntax with parenthesis remains valid, of course.

USABILITY, SECURITY (14.2): .HH ON/OFF changes

Starting with LISTSERV 14.2:

- .HH commands now nest.

- The .HH ON and .HH OFF dot commands are respected in KEYWORDS files called from list headers with the .IK dot command. Previous builds ignored .HH commands in KEYWORDS files.

The following should be noted:

- In a KEYWORDS file, .HH OFF found in excess of .HH ON will be ignored. This ensures that a KEYWORDS file called from inside of an .HH ON block will not expose the remainder of that block upon return from the call.

- Similarly, LISTSERV will internally generate as many .HH OFF tags as necessary before exiting the KEYWORDS file and returning to the list, if more .HH OFF tags than .HH ON tags exist in the KEYWORDS file.

Both of these precautions ensure that .HH coding errors in a KEYWORDS file will not result in exposure of keyword settings that it is desired to keep hidden.

*USABILITY (14.2): New &*TOFIELD; and &*NAME; substitutions for mail-merge*

Two new built-in mail-merge substitutions and a related (optional) DISTRIBUTE keyword are available starting with version 14.2. These have been added to solve concerns about "(no name available)" appearing in the To: field, and at the same time the flaw in using:

To: "&NAME;" <&*TO;>

which is not correct for all possible values of &NAME.

This feature adds one optional DISTRIBUTE keyword:

NAMEFIELD=xxx

This indicates the name of the XDFN/DBMS field containing the name of the recipient. If absent, the name is unknown (see below). Note that the correct syntax is NAMEFIELD=NAME, not NAMEFIELD=&NAME; or similar. Any available column name can be specified for NAMEFIELD.

In the case of DBMS=LIST, the default value of NAMEFIELD=xxx is set automatically from the "DBMS=" keyword and/or the system defaults found in SITE.CFG. Normally, one should omit NAMEFIELD=xxx for a list distribution and LISTSERV will provide the value in the field identified as the name field in the list header. However, one can specify NAMEFIELD=xxx even for a list distribution. This provides a way for a DISTRIBUTE job to use a different column for the name, for example in cases where there might be several name columns in the table, for instance with different character sets or one with and one without accents.

Two substitution variables are added. &*NAME is replaced with the variable specified in NAMEFIELD=xxx. If unknown, the empty string is substituted as a constant. This is just a convenient way to refer to the name field in examples or generic jobs, regardless of what it is really called.

The second substitution is &*TOFIELD, which is a correct RFC822 To: field (without the "To:") for the supplied name and e-mail address. If the name is unknown or missing, the result is the same

as &*TO. A missing name is NULL, the empty string or '(no name available)'. To clarify, the correct use is:

To: &*TOFIELD;

Note that there is a small performance cost for this option. The RFC822 rules are somewhat time-consuming; additionally, this also requires parsing XDFN lines to extract the name field (when not needed, LISTSERV simply passes them on to LSMTP and adds its own XDFN). Finally, the NAME field is passed to LSMTP even if it is only used for &*TOFIELD.

PERFORMANCE: [Windows] SMTP worker load balancing improvements

An option for strict "round-robin" delivery to the outbound MTA via LISTSERV's SMTP "worker" processes has been added to the Windows version of LISTSERV. This option is not available under unix or OpenVMS at this time.

To activate this feature add the line

SMTP_STRICTLY_ROUND_ROBIN=1

to SITE.CFG. Stop and restart LISTSERV to pick up the change.

Activation of this feature tells the SMTP workers, if plural, to only process messages whose spool ID is congruent to their worker ID modulo the total number of workers. In other words, they only process their fair share of messages. Messages without a numeric spool ID, such as those created manually or by debug scripts, are "free for all" and will be processed by the first worker that sees them. The SMTP worker log files (x:\LISTSERV\LOG\SMTPS#n-yyyymmdd.LOG) will have a message at the top confirming that the feature has been activated.

DRAWBACK: if a worker should die, some messages will be left unprocessed. In practice, we have never ever seen a worker die unless there is some kind of global system problem, like a disk crash or out of swap space condition (not counting workers that abort due to a configuration error). But, in theory, it could happen. Another potential drawback is more cycles spent scheduling workers since a short burst of messages will require the participation of each and every worker.

IMPORTANT: this feature is incompatible with worker pools. Do not use SMTP_STRICTLY_ROUND_ROBIN=1 if worker pools are defined.

SMTP_STRICTLY_ROUND_ROBIN=1 should generally only be used to solve an existing load balancing problem. The default allocation of messages to SMTP workers uses dynamic load balancing, which is in most cases the more efficient method, though SMTP_STRICTLY_ROUND_ROBIN can be helpful in some exceptional situations.

By default (that is, without SMTP_STRICTLY_ROUND_ROBIN=1), the various workers are synchronized using a single, shared event flag. LISTSERV sets the event flag when there is a new message to process. Because there is only one event flag, LISTSERV could not possibly be favoring a particular worker, or ignoring a particular one. All the workers wait on this event flag when they run out of work to do, and Windows decides which of the workers to wake up if several are waiting (only one is awakened). Windows does not know that they are numbered #1, #2, etc. They are just different processes as far as Windows is concerned. All tests done so far have shown that Windows seems to use a round-robin algorithm. Simulations have all shown approximately the same number of wakeups for every process.

However, our simulations were based on workers that did a variety of things at the same speed. If for any reason a worker is slower (typically because it is talking to a slower MTA), it will tend to be busy when the other workers are waiting for new work. That is, there will tend to be a more-than-fair occurrence of the state where all workers are waiting except for the slow worker, which is still busy finishing its last message. At that point it is guaranteed that the next message will be processed by one of the fast, waiting workers. This dynamic load balancing is by design. MTA speed can vary significantly in the space of a few minutes and the workers adjust their respective shares accordingly and without human intervention. The result is more messages to workers (and thus to MTAs) that have more capacity right now, though this can change in only a few minutes.

Thus a slower target server will normally lead to a less-than-fair share of messages. The purpose of SMTP_STRICTLY_ROUND_ROBIN=1 is to enforce a fair share for all MTAs by earmarking messages for a particular worker.

OS support information (important)

LISTSERV version 14.1 (formerly 1.8e-2002a) was the last version for several operating systems which have become obsolescent over the life of this product cycle. **The following operating systems are no longer supported in version 14.3**:

Windows NT 4.0
Windows 95/98/Me
BSDi (Intel)
IRIX (MIPS)
Solaris-x86 (Intel)

Sites running these operating systems should migrate to a different operating system. Please contact your sales representative for further information.

Sites running the Windows 95 shareware should note that their licenses will not activate the product under Windows XP. Please contact your sales representative for alternatives if you are planning to upgrade to Windows XP (optionally you may migrate to the LISTSERV Lite Free Edition). Sites running the Windows 95 Lite Free Edition can simply upgrade to the Windows NT/2000/XP LISTSERV Lite Free Edition. (Naturally you may also elect to continue running LISTSERV under Windows 95/98/Me, but there will be no further new versions or fixes for that platform.)

It should be noted that L-Soft dropped support for the following operating systems with the original release of LISTSERV 14 (in other words, LISTSERV 13 or 1.8d was the last version for these platforms):

Windows NT 3.5, 3.51, 4.0 pre-SP6 (Intel)
Windows NT (Alpha AXP)
SunOS 4.x (SPARC)
Ultrix (MIPS)
OpenVMS (VAX)
VM/SP, VM/HPO

On the plus side, L-Soft now formally supports FreeBSD (Intel) and Linux (S/390) since LISTSERV 14.

A comprehensive list of operating systems (and versions) under which LISTSERV is supported can be found at

http://www.lsoft.com/products/default.asp?item=listserv-ossupport

Applying LISTSERV 14.3

IMPORTANT: Install your maintenance LAK before upgrading! A maintenance LAK (License Activation Key) must be installed before upgrading, or LISTSERV will not start after the upgrade. More information

The installation kits on the L-Soft FTP site can be used to install a new copy of LISTSERV or upgrade an existing installation.

To download LISTSERV 14.3, simply go to L-Soft's web site (or to FTP.LSOFT.COM) and download an evaluation copy of LISTSERV or LISTSERV Lite, then follow the installation instructions for your operating system. The kits can be found at:

http://www.lsoft.com/download/default.asp?item=listserveval

http://www.lsoft.com/products/default.asp?item=listserv_lite

SPECIAL NOTES

1. This document does not include upgrade instructions. Please see the installation guide specific to your OS platform for upgrade instructions.

VMS:	http://www.lsoft.com/manuals/1.8e/vmsinst.html
Unix:	http://www.lsoft.com/manuals/1.8e/unixinst.html
Windows:	http://www.lsoft.com/manuals/1.8e/ntinst.html

VM sites currently at the 1.8e level should download ftp://ftp.lsoft.com/listserv/vm/upd143.hex and install it per the "Fixes and Upgrades" section of ftp://ftp.lsoft.com/listserv/vm/00-read.me . VM sites currently at the 1.8d or earlier level must first upgrade to 1.8e (LISTSERV 14) before applying this level set fix.

2. The Unix kit now only requires a single download. You do not need to also download a common.tar.Z file as in previous versions, as those files are now included in the single OS-specific installation file.

3. LISTSERV version 14.3 is available only for operating systems currently supported by L-Soft. When browsing FTP.LSOFT.COM, you may find installation kits for other operating systems, such as Ultrix or SunOS 4.x, but these kits will be based on older versions and/or code bases. L-Soft no longer has development systems for unsupported operating systems and is not in a position to compile LISTSERV 14.3 for these systems.

end of file