Description of the changes for the 2003a "level set" release
Version 1.8e of LISTSERV(R)

26 September 2003
(updated 2 March 2004)

THE 2003a LEVEL SET

The 2003a level set includes all known fixes, patches and between-release enhancements up to 26 September 2003, as follows:

- USABILITY: New SOFTBOUNCE changelog record for nolist changelogs
- USABILITY: New DROP argument for SERVE OFF command
- USABILITY: New ALLOW-BOUNCES parameter for Loopcheck=
- USABILITY: Change to .BB conditional processor
- USABILITY: New &*TOFIELD; and &*NAME; substitutions for mail-merge
- USABILITY: New SPAM_ALERT site configuration variable
- USABILITY, SECURITY: .HH ON/OFF changes
- SECURITY: [Unix] Setting umask/read permissions for web interface
- SECURITY: [Windows] Protecting web index files (IIS) [updated]
- SECURITY: [Windows] Protecting web index files (Apache) [added]
- ANTI-VIRUS: [Windows] F-Secure Anti-Virus 5.41 now supported
- PERFORMANCE: [Windows] SMTP worker load balancing improvements
- PERFORMANCE: Miscellaneous improvements
- Miscellaneous fixes
- Web interface enhancements
- Miscellaneous fixes for 'wa'

Also please note:

- OS support information (important)

****
* USABILITY: 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

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

****
* USABILITY: 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 on the floor 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: 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 1.8e.

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: Change to .BB conditional processor *
****

Prior to this level set, 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'.

For backward compatibility, the original syntax with parenthesis remains valid.

****
* USABILITY: 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 the 2003a level set release. 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).

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. Note that the correct syntax is NAMEFIELD=NAME, not NAMEFIELD=&NAME; or similar. NAMEFIELD=xxx is not ignored for a list

distribution. Any available column name can be specified for NAMEFIELD, at the risk of making a mistake. The design assumption was that in some cases there might be several name columns in the table, for instance with different character sets or one with and one without accents. It was thought best to allow this to override the internal default, even if the default is correct. However, normally one should omit NAMEFIELD=xxx for a list distribution and LISTSERV will provide the correct value.

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 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.

****
* USABILITY: New SPAM_ALERT site configuration variable *
****

A new Boolean site configuration variable, SPAM_ALERT=, has been added beginning with this level set release. 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
            Win:  SPAM_ALERT=0
```

(Under unix, don't forget to export.)

****
* USABILITY, SECURITY: .HH ON/OFF changes *
****

Starting with the 1.8e-2003a level set release:

- .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.

****
* SECURITY: [Unix] Setting umask/read permissions for web interface *
****

In previous versions, LISTSERV has written files to its web interface with whatever default umask was set for the 'listserv' user. This left web index files (listname.ind*) world-readable, and although directory browsing could be disabled in the web server, if one knew the naming conventions for these files, it was not difficult to guess an appropriate URL to be able to read them (usually for the purpose of obtaining addresses of people who post to lists).

From the 1.8e-2003a level set release, LISTSERV now sets a default umask of 0g7 (group being left up to the server administrator) at startup time, and will write these files without world-read permission, closing this potential exposure.

LISTSERV also no longer writes WWWTPL files into the /archives directory with world-read permission.

It may be necessary to delete all files from the /archives directory tree and stop and restart LISTSERV to rebuild them in order for the new permissions scheme to propagate across the entire LISTSERV web interface. Naturally, appropriate permissions can also be applied manually, with the understanding that LISTSERV will subsequently set permissions per the above for all new web interface files it creates.

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

Previously, it has not been possible to protect LISTSERV's web index files under Windows/IIS from being read by general users. Although the files themselves are not linked from anywhere, and it is assumed that security-minded sites have directory browsing disabled, it is not difficult to guess the names of the files if one is determined to read the information they contain. Since they do contain email addresses of people who have posted to the list, leaving them open to read by general users, some of whom may be searching for addresses to add to

spam mailing lists, is considered unacceptable. However, because WA.EXE and the web server as a whole run under the same account (usually IUSR_machinename by default), by definition they cannot be told apart.

From LISTSERV 1.8e-2003a, it is possible to protect the index files from casual viewing. The procedure outlined below assumes the following:

- 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 other words, you haven't given the Everyone user full access).

Step 1: 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: Change the Windows file permissions on the scripts directory to give RX permissions to IUSR_LISTSERV. 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 IUSR_WWW's permissions intact.

Step 3: 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: 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.

Regular web access, on the other hand, will fail unless files have been given some kind of broader access than IUSR_LISTSERV,Read. The current build of LISTSERV sets Everyone,Read on the files that need to have it.

It may be necessary to delete all of the HTML files and images from the 'archives' directory tree in order to let LISTSERV rebuild them.

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

It is much simpler to protect the web index files under Windows if you are using Apache.

You must enable (if it is not already enabled; the installed default is to disable) the use of .htaccess files that may control at least the "Limit" directive, and 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.

****
* ANTI-VIRUS: [Windows] F-Secure Anti-Virus 5.41 now supported *
****

F-Secure Anti-Virus 5.41 is now certified with LISTSERV 1.8e. Sites with current maintenance may obtain a key for FSAV 5.41 from their sales representative (earlier FSAV keys will not work with FSAV 5.41).

The FSAV 5.41 kit may be downloaded from ftp://ftp.lsoft.com/f-secure/541srv9180.zip .

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

****
* 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.

[Please note carefully that 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. L-Soft does not believe that it will be a very significant difference except in low-volume, many-worker scenarios (in a high-volume scenario, workers are not very often in wait state and it does not matter from a resource utilization standpoint which messages they pick and on which criteria).

IMPORTANT: this feature is incompatible with worker pools. Any workers placed in a pool will ignore SMTP_STRICTLY_ROUND_ROBIN=1 and process all messages for the pools they are currently enabled for. Workers not in a pool, if any, will honor the option and this is likely to lead to unprocessed messages. This feature cannot be implemented for pools without major work because the workers do not communicate with each other. This means they cannot maintain a catalogue of all pools currently active and how many workers are currently working on what pools.

It should be made clear that any imbalances in load sharing are outside LISTSERV. 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 pseudo-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 workers by earmarking messages for a particular worker.

```
****************************************
* 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.

```
**********************
* Miscellaneous fixes *
**********************
```

The following problems have been fixed with the release of the LISTSERV 1.8e-2003a level set.

- With "Attachments= No" it was possible for the content filter to be called on LSMTP bounces, and reject them.

- The LISTSERV 1.8e-2002a level set release had a bug that caused the owner-LISTSERV pseudo-mailbox to not be recognized as a valid address.

- "Send= Editor,Hold,Confirm,NOMIME" was not working for confirmation requests sent in response to messages sent from an editor address.

- A variety of problems stemming from the use of .BB statements in which a variable from a DBMS field was used, but does not appear anywhere else other than in .BB statements, were addressed in the DISTRIBUTE optimizer.

- Unquoted semicolons in .BB conditionals now work. Previously one had to code

```
.BB '&X;' = 1
```

in order to force LISTSERV accept the variable with the semicolon. The syntax

```
.BB &X; = 1
```

is now acceptable.

- Sites with an explicit value set for the DEFAULT_SPLIT variable in the site configuration file would receive the error "Error in header data stream" in the web interface when trying to do anything that involved a string function (for instance, when modifying a list header or a template, creating a list, sending a posting, doing a bulk upload, and so forth). Removing the DEFAULT_SPLIT setting and restarting LISTSERV was recommended as a workaround. With the level set release, setting DEFAULT_SPLIT is no longer a problem.

- When a list was 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 was not created.

- Probes returned to LISTSERV in a bad format were being forwarded to list owners without any explanation, rather than being processed correctly by LISTSERV's non-probe bounce code.

- When the primary editor of a moderated list was coded as an access-level (eg, "Editor= Owner" or "Editor= Owners"), LISTSERV would generate approval requests to an invalid address (typically "OWNER@.BITNET"). LISTSERV will now avoid this and default to sending the approval request to the first listed non-quiet list owner. The documented recommendation not to use an access-level as the primary editor of a list continues to stand, as it is not always the case that the list owner should be the primary (or only) editor.

- LISTSERV would log the message "Changelog updated" even when the quantity written was zero.

- It was noted that the LISTSERV-generated RFC822 "Message-ID:" header was not unique if more than 100 outbound messages were generated per second.

- [Non-VM] In previous versions a PUT of a KEYWORDS file did not reset cached list header information and a stop and restart of LISTSERV was required to re-cache the headers after the PUT.

****
* Web interface enhancements *
****

- "Show All Lists" option on Subscriber's Corner is no longer shown to users for whom it is not available.

- "Remove mailing list" link on Admin page provides instructions for removing a mailing list.

****
* Miscellaneous fixes for 'wa' *
****

The "wa" CGI executable is now at version 2.3.27. Since the release of the LISTSERV 1.8e 2002a level set in December 2002, the following fixes have been applied:

- Fixed multiple bugs in archive searches:
  - o match display in multi-list search is improved
  - o multiple bugs in handling of .M from LISTSERV fixed
  - o incorrect sorting of dates should be fixed
- Fixed URL escaping in multiple links
- Fixed header optimizer (List Wizard)

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

****
* OS support information (important) *
****

LISTSERV 1.8e is the last version for several operating systems which have become obsolescent over the life of this product cycle. The operating systems which will no longer be supported after version 1.8e are:

Windows NT 4.0 SP6
Windows 95/98/Me
BSDi (Intel)
IRIX (MIPS)
Solaris-x86 (Intel)

Sites running these operating systems should start planning now for a migration 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 1.8e (in other words, LISTSERV 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) in LISTSERV 1.8e.

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 THE 2003a LEVEL SET *
*******************************
```

Level sets are standard installation kits that replace the previous installation kits on L-Soft's FTP and web servers. They can be used to install a new copy of LISTSERV or upgrade an existing installation. A level set is similar to a Windows NT CD-ROM with the latest service pack pre-applied.

To download the 2003a level set, 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

LICENSE KEY FOR THE 2003a LEVEL SET

The level set is a no-cost upgrade to customers licensed for version 1.8e and will work with your existing 1.8e license key.

The level set will NOT work with a 1.8d or older license key. If you are still running a pre-1.8e LISTSERV installation and would like to upgrade to this level-set release, please contact your sales representative to get a 1.8e LAK BEFORE you attempt to upgrade.

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/fix2003a.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 before applying this level set fix.

2. Make sure to update ALL LISTSERV executables, including wa, lsv_amin, lcmd, etc., and associated files, such as the mail and web default templates. Unix Classic sites need to be sure to download both common.tar.Z and the `uname`.tar.Z for their operating system. For unix we also recommend touching all files in the distribution prior to running 'make update', to ensure that they are "newer" than your existing production files.

3. The 2003a level set is only available 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 the 2003a level set for these systems.

end of file