

AUTOMON®-VTAM

INSTALLATION SUMMARY

Release 2.1

WORLD HEADQUARTERS
UNICOM Systems, Inc.
1032 Cove Way
Beverly Hills, CA 90210
(818) 838-0326

MISSION HILLS DIVISION
UNICOM Systems, Inc.
15535 San Fernando Mission Blvd.
Mission Hills, CA 91345, U.S.A.
(818) 838-0606 Fax:(818)838-0776

FAREAST DIVISION
UNICOM Systems Korea
Hawangshimnidong 301 102
Sungdong-ku, Seoul, Korea
(02) 296-5476



Unicom Systems, Inc.
15535 San Fernando Mission Blvd.
Mission Hills, California 91345

Sixth Edition – November 13, 1998

The information in this document applies to Release 2.1
Of the AUTOMON®-VTAM™ product for all operating environments.

Published by UNICOM Systems Inc.
15535 San Fernando Mission Blvd. Suite 310
Mission Hills, California 91345

© Copyright UNICOM Systems, Inc.

All rights reserved

AUTOMON®-VTAM

Release 2.1

TABLE OF CONTENTS

PREFACE.....	1
INSTALLATION INTRODUCTION.....	2
MINIMUM READING REQUIREMENTS	2
<i>READING REQUIRED FOR THE INSTALLER</i>	2
<i>READING REQUIRED FOR OPERATION AND MAINTENANCE</i>	2
<i>READING REQUIRED FOR PROBLEM HANDLING</i>	2
COMPRESSION PHILOSOPHY AND TECHNIQUE	3
<i>TYPES OF COMPRESSION TECHNIQUES</i>	4
<i>REPETITIVE CHARACTER ELIMINATION</i>	4
<i>OUTBOUND IMAGE SAVING</i>	4
<i>INBOUND MIRRORING</i>	5
INSTALLATION GENERAL INFORMATION	6
THE INSTALLATION TAPE	6
MVS INSTALLATION	7
<i>MVS INSTALLATION - BASIC PRODUCT INSTALL</i>	7
<i>ORDERING THE LPA</i>	8
<i>CREATION OF A CUSTOMIZATION TABLE</i>	9
<i>MINIMUM TERMINAL INPUT SIZES</i>	13
<i>SNA RU SIZE CONSIDERATIONS</i>	14
DOS/VSE INSTALLATION - BASIC PRODUCT INSTALL	14
<i>CREATION OF A CUSTOMIZATION TABLE</i>	16
<i>LOAD THE SVA MEMBER</i>	19
<i>MINIMUM TERMINAL INPUT SIZES</i>	19
<i>NON-CICS APPLICATIONS</i>	20
KEYWORDS OF THE UNICOM SYSTEMS INC.INSTALLATION PROGRAM	20
<i>FORMAT OF THE STSINST KEYWORDS</i>	20
<i>KEYWORD DESCRIPTIONS</i>	21
INSTALLING THE PRODUCT CONTROL PASSWORD	23
<i>DATA FIELDS OF THE PASSWORD CONTROL TABLE</i>	24
CUSTOMIZATION	26
CODING THE CUSTOMIZATION TABLE	26
<i>CODING FORMAT</i>	26
<i>CUSTOMIZATION STATEMENTS</i>	27
<i>RULES OF CODING THE CUSTOMIZATION TABLE</i>	27
INCLUSION AND EXCLUSION OF APPLICATIONS AND LU'S	48

PREFACE

This preface is designed to provide a quick overview of the contents of the AUTOMON®-VTAM reference guide. Each major section of the manual is briefly described, along with the starting page number for that section.

INTRODUCTION (SECTION 01) Page 2

This section describes AUTOMON®-VTAM in general terms, the types of compression techniques used and methods of employing them. Also included is a discussion of the philosophy behind terminal datastream compression - why it is needed and the techniques used to reduce datastreams.

INSTALLATION (SECTION 02) Page 6

This section describes the process of installing AUTOMON®-VTAM, both for MVS and DOS/VSE. The contents of the installation tape, a description of each program, and the steps required, along with JCL examples, for making it operational are detailed.

CUSTOMIZATION (SECTION 03) Page 34

This section deals with the methods of tailoring AUTOMON®-VTAM to fit the needs of your environment. The process of coding and installing the User Option Table is presented, along with an explanation of each of the customization features.

INDEX (SECTION 04) Page 50

The index arranges key words and phrases in alphabetical order and provides the page number reference to their location.

INSTALLATION INTRODUCTION

MINIMUM READING REQUIREMENTS

If you are approaching AUTOMON®-VTAM for the first time, the following subjects in this manual must be read. These discussions will give you a good working knowledge of the product.

The reading list is sub-divided according to the presumed requirements of the reader.

READING REQUIRED FOR THE INSTALLER

In <i>INSTALLATION</i> , section number 02:	Page
<i>GENERAL INFORMATION</i>	09
<i>THE INSTALLATION TAPE</i>	10
OR	
<i>MVS INSTALLATION</i>	11
<i>DOS/VSE INSTALLATION</i>	21

In *CUSTOMIZATION*, section number 03:

<i>CODING FORMAT</i>	34
<i>VSTART - ESTABLISH INITIAL OPTIONS</i>	37
<i>VAPPL - IDENTIFY APPLICATIONS FOR COMPRESSION</i>	40
<i>VLU - IDENTIFY LOGICAL UNITS FOR COMPRESSION</i>	52
<i>VFTRAN - IDENTIFY FILE TRANSFER BEGINNING AND END</i>	61
All of <i>OPERATION</i> - <i>MVS SYSTEMS</i> , section	04
	65
or <i>OPERATION</i> - <i>DOS/VSE SYSTEMS</i> , section	05
	105

READING REQUIRED FOR OPERATION AND MAINTENANCE

In <i>INTRODUCTION</i> , section number 02:	Page
<i>COMPRESSION PHILOSOPHY AND TECHNIQUE</i>	4
All of <i>CUSTOMIZATION</i> , section number	03
	34
All of <i>OPERATION</i> - <i>MVS SYSTEMS</i> , section	04
	65
or <i>OPERATION</i> - <i>DOS/VSE SYSTEMS</i> , section	05
	105

READING REQUIRED FOR PROBLEM HANDLING

All of <i>USING THE VTAM TRACE FEATURE</i> , section	07
	131
and <i>PROBLEM ANALYSIS AND DIAGNOSIS</i> , section	09
	147
and <i>APPENDIX C - PROBLEM REPORTING</i>	204

COMPRESSION PHILOSOPHY AND TECHNIQUE

In today's large-system, multi-terminal networks, the single most critical factor of day-to-day operation has become terminal response time. Excessive response time is responsible for many of the problems in company operations. It is often the major factor behind project overruns, worker frustration and apathy, and careless errors.

There are many factors that determine how fast a terminal can respond, but basically they fall into three major categories:

- 1). Overall system tuning.
- 2). Individual application program efficiency.
- 3). The amount of data transmitted.

Today's network system programmer, for the most part, only has control over the first of these variables, overall system tuning. He can distribute the load over the available resources to maximize efficiency, and adjust the various application controls to eliminate bottlenecks. He can do very little, in most shops, to control application program efficiency or data transmission amounts, since much of today's application software is made up of vendor-written packages.

In the local environment, the amount of data transmitted is not quite the noticeable factor that it is in the remote environment, since the transmission rate is considerably higher for local terminals than it is for remotes. For remote terminals running at 4800 or 9600 BAUD, it is often the single most important factor.

Terminal datastreams produced by most of today's application software is largely inefficient. This simply means that much more data is transmitted on every transaction than is necessary. There are three major reasons for this:

- 1). Inefficient mapping techniques. Most application software uses some sort of mapping to build their terminal datastreams. Rows of asterisks or dashes, fields of spaces, etc. are assembled into the map as character constants and transmitted intact, with no attempt made to reduce the number of characters by use of the native 3270 repeat-to-address order.
- 2). Inefficient outbound programming techniques. Many application programs are designed to produce the same display, over and over, to be sent to the terminal. The field titles present in the display are sent, again and again, even though they are already displayed on the terminal. It is easier, from a programmer's standpoint, to erase the screen and re-send the entire display than to remember what was sent last and only send the variable fields.
- 3). Inefficient inbound programming techniques. Many application programs also overuse the modified-data-tag on screen fields. Again, it is easier for the programmer to turn the modified-data-tag on for all variable fields, so that they are all re-transmitted when the operator presses the ENTER key, whether the fields were changed or not.

As always, the best solution from a system efficiency standpoint would be to rewrite all application software to eliminate these inefficient techniques. However, since this is almost never a cost-effective approach, the answer lies in a software package which will examine the outgoing datastreams and dynamically correct the problems.

There are several datastream compression packages on the market today which do a good job of optimizing terminal datastreams. However, they all have a common drawback. Each of these packages operates for a single application system. That is, they work for CICS, or IMS, or TSO, for instance but not all three. If you want datastream compression for more than one type of application system, you must buy more than one type of compression package.

AUTOMON@-VTAM addresses all three of the problem areas discussed above, and, in addition, removes the restriction of only working for one application system at a time. It will work on all terminal datastreams in the system, as long as the application is controlled by VTAM.

TYPES OF COMPRESSION TECHNIQUES

There are three types of compression techniques used by AUTOMON@-VTAM. These are:

- 1). Repetitive character elimination.
- 2). Outbound image-saving.
- 3). Inbound mirroring.

Each of these techniques are discussed below.

REPETITIVE CHARACTER ELIMINATION

The average outbound terminal datastream contains from 20 to 25 percent repetitive characters. Strings of spaces, nulls, asterisks, dashes, etc. are common in almost all output displays.

Most application programs and map generators handle a string of repetitive characters as a single field, which may or may not be preceded by a START-BUFFER-ADDRESS (SBA) sequence.

Thus, a row of 30 asterisks starting somewhere on the screen would appear in hexadecimal form as:

```
5C 5C
```

for a total field length of 30 bytes.

This same field of 30 asterisks can be sent to the terminal and displayed properly in only four bytes by taking advantage of the buffer display logic that is present in all 3270-mode devices. To do this, we can use the REPEAT-TO-ADDRESS (RA) sequence instead of the SBA sequence. With an RA sequence, you tell the device what character you want to repeat, and the ending buffer address of the field. Thus, the following sequence would display exactly the same on the terminal, but only needs to transmit four bytes:

```
3C xx xx 5C (The xx xx is the buffer address of the end of the stream).
```

AUTOMON@-VTAM examines the outgoing datastreams and looks for fields of repetitive characters that exceed four bytes. When one is found, the ending address of the field is calculated and an RA sequence is inserted into the datastream. The extra characters in the original field are no longer needed and are discarded.

OUTBOUND IMAGE SAVING

It is very often the case in day-to-day terminal operation that an operator will use the same transaction continuously over a period of time. This usually involves the same basic display with only the data fields changing from one display to the next. That is, the screen titles and text fields remain the same.

Terminal datastream programming provides a technique to minimize data transmission by using a terminal write of data only for all but the first display. In other words, the first time a particular display is sent to the terminal, the ERASE command is included to clear the screen, then all of the data, titles, text and data fields are sent. For the next display, the ERASE command is not sent and only the data fields are output. As long as the operator continues to use the same display, there is no need to continue to clear the screen and re-send all of the data with every output transmission.

The problem with this technique is that it requires more sophisticated programming effort. The application program needs to know that the display in question is the same display (except for the data fields) that was sent last time. This can involve quite a bit more program logic, and most application programs just don't go to that much trouble. It's far easier to erase and re-send all data.

AUTOMON®-VTAM imposes this programming technique on all output datastreams without placing the burden on the application programs.

An image of each terminal buffer is maintained in memory. This image is always an exact replica of what is displayed on the terminal at any given time. Now, when an output datastream is sent to the terminal, **AUTOMON®-VTAM** compares the outgoing image with the image of the buffer.

If it determines that the outgoing datastream is an entirely different screen display, the repetitive character elimination logic is performed on this datastream, then it is sent to the terminal, leaving the ERASE command undisturbed.

If, however, the outgoing datastream contains many of the same title and text fields as currently in the image of the terminal buffer, **AUTOMON®-VTAM** will remove the ERASE command, making it a data-only write, and also remove all fields in the outgoing datastream that are already present on the terminal screen. Even the data fields are examined, and removed if the same data is being sent again.

The result is a tremendous reduction, system-wide, in the amount of data that is transmitted to the terminals.

INBOUND MIRRORING

On the input side, that is, the data being transmitted from the terminal to the host system, there is nothing that can be done to reduce the amount of data that is keyed by the operator. There is, however, something that can be done to reduce the total amount of data transmitted from the terminal in many cases.

When an operator enters data into any field on the screen, the logic contained in the terminal causes a flag to be turned on in the attribute byte of that field, indicating that the field was modified. This flag is called the modified-data-tag, or MDT.

The MDT can also be turned on by an application program when the datastream is transmitted to the terminal. If this is done, the field will be transmitted back from the terminal, whether the operator enters anything in that field or not. It is a very common programming technique, to turn on the MDT for many (in some cases all) of the fields prior to sending the datastream to the terminal. This eliminates the necessity in the application program of determining which fields the operator modified.

AUTOMON®-VTAM can reduce the amount of data transmitted from the terminal by turning off all modified-data-tags as the datastream is going out to the terminal. Now, only the data that is actually modified by the operator is transmitted back in.

Since **AUTOMON®-VTAM** keeps an image in memory of the current state of the terminal screen, it knows which fields had MDT on, and therefore would normally have been re-transmitted. If these fields do not come back in, they are inserted into the datastream prior to sending it on to the application system. Thus, when the application program receives the input data, it looks just as it would normally appear. The application program never knows the difference, but the actual amount of data transmitted is significantly reduced.

INSTALLATION GENERAL INFORMATION

Installation of AUTOMON®-VTAM involves three separate processes:

- 1). Installation of the basic AUTOMON®-VTAM product. This involves linking the installation program from the tape, running the installation program to extract and deblock the AUTOMON®-VTAM program modules and create the installation JCL, tailoring the installation JCL and running it to load the program modules.
- 2). Creation of a customization table. This involves editing the distributed skeleton table to supply your unique application names, as well as any other customization desired.
- 3). Changes to be made to some application systems terminal definitions. CICS, for instance, requires that the RAMAX value be set to at least 1536 and the minimum TIOA length of each terminal be set to an equal value. TSO requires no changes but IMS may require changes similar to CICS, as may any other VTAM application which is to be compressed.

For your initial evaluation of the product, we suggest that you select one VTAM application for testing; then tailor the customization table to use only that application. You may also want to limit the number of terminals that can participate, in order to more easily control and evaluate the results at first.

Once you have experimented with the product, you could open it up to more application systems and/or terminals.

THE INSTALLATION TAPE

The installation tape consists of two files.

File 1. The first file is the object records for four program modules which are used both to finish the installation process and to provide the temporary expiration password. A utility memory display/alter program is also included in this file. These four programs are:

- 1). STSINST- The installation program.
- 2). STS0100 - The password processor program.
- 3). STSPASS - The password table. This module is pre-loaded with a 30-day password.
- 4). STSCORE - This is a utility program which runs in CICS. It is a memory display and alter program which can be used to access and optionally alter various control blocks in CICS as well as in AUTOMON®-VTAM.

The STSPASS module contains a temporary password to keep the product from expiring for approximately 30 days. If you need to change the password, refer to *INSTALLING THE PRODUCT CONTROL PASSWORD*, later in this section.

File 2. The second file contains the records for seven of the products available from Unicom Systems Inc. Systems. You must use the STSINST program to extract and deblock the product to be installed. The products present in file 2 are:

AUTOMON®-VTAM	VTAM-WINDOWS
AUTOMON®-VTAM	HELP-WINDOWS
CICS-JUGGLER	CICS-FILESERV
CICS-JUGGLER/SVT	

Feel free to install any of the other products on the tape. Each product has a documentation member with the installation procedure and a brief overview of the product. Appendix C of this manual contains a general overview of each product. A technical reference guide for the product can be obtained from your sales representative at Unicom Systems Inc. Systems.

MVS INSTALLATION

MVS INSTALLATION - BASIC PRODUCT INSTALL

Installation consists of the following steps:

- 1). Link-edit the four program modules from file 1 of the tape.
- 2). Run the STSINST program to print the full installation instructions and create the installation JCL.
- 3). Tailor the installation JCL for your library names.
- 4). Run the installation JCL to load all programs and files.
- 5). Define a type-3 SVC to MVS.
- 6). Define a sub-system control table entry to MVS
- 7). Define the required CICS table entries.
- 8). Make any required changes to your application terminal tables.

For MVS, you must prepare initial JCL for two steps:

Step 1. Load the installation programs and password table

The following JCL can be used to install the first file of the tape for MVS users:

```
//LOADSTS    JOB    1,'ACCOUNT-ID',MSGCLASS=X
//STEP1      EXEC   PGM=IEWL,PARM='LIST,LET,XREF'
//SYSPRINT   DD     SYSOUT=*
//SYSLIB     DD     DSN=CICS???.LOADLIB,           [Note 1]
//           DD     DISP=SHR
//SYSLIN     DD     DSN=MASTER.TAPE,
//           DD     UNIT=TAPE,
//           DD     VOL=(,RETAIN,,,SER=INPUT),
//           DD     LABEL=(1,NL),
//           DD     DCB=(RECFM=FB,LRECL=80,
//           DD     BLKSIZE=80),
//           DD     DISP=(SHR,PASS)
//SYSUT1     DD     UNIT=SYSDA,
//           DD     SPACE=(1024,(20,20))
//SYSLMOD    DD     DSN=SYS1.CMDLIB,             [Note 2]
//           DD     DISP=SHR
```

[Note]:

1. This must be a CICS load library where the command-level Assembler stub program (DFHEAI) resides. This is required for the STSCORE program. You should move STSCORE to a CICS load library when you install STSCORE for CICS.
2. This can be any load library, however, the STS0100 and STSPASS modules must either reside in the same library as the VTEXPRES program or be in a library which is available to TSO. The VTEXPRES program functions as a TSO command processor and is normally link-edited to SYS1.CMDLIB. For further considerations, please refer to *ORDERING THE LPA*, later in this section.

Step 2. Print installation instructions and create installation JCL

The following JCL executes the STSINST program to print the installation documentation and create the install JCL:

```
//STEP2      EXEC  PGM=STSINST
//SYSPRINT   DD    SYSOUT=*
//STEPLIB    DD    DSN=SYS1.CMDLIB,           [Note 1]
//           DD    DISP=SHR
//MPRDIN     DD    DSN=MASTER.TAPE,UNIT=TAPE,
//           DD    LABEL=(2,NL),VOL=SER=INPUT,
//           DD    DCB=BLKSIZE=32000,
//           DD    DISP=(SHR,PASS)
//SYSPCH     DD    DSN=????.(STSJCL),         [Note 2]
//           DD    DCB=BLKSIZE=????,DISP=SHR
//SYSIN      DD    *
PRODUCT=EXPRESS [Note 3]
MODE=PRINT      [Note 3]
OPSYS=MVS       [Note 3]
LINES=56        [Note 3]
JCL=INSTALL     [Note 3]
/*
```

[Notes]:

1. The STEPLIB must identify the library where STSINST was link-edited in step 1.
2. This is any PDS where you want to load the installation JCL. You must specify a member name for your JCL and you must specify the blocksize of the PDS on the DCB parameter.
3. See *KEYWORDS OF THE UNICOM SYSTEMS INC.INSTALLATION PROGRAM*, later in this section, for the meaning of these keywords.

At this point, the full installation instructions will be printed. The installation JCL will be loaded into the member name of the PDS specified by SYSPCH. Retrieve or view the printed installation documentation and follow the instructions provided there to complete the installation of the basic product.

ORDERING THE LPA

It is sometimes necessary to provide an IEAPAKxx member in SYS1.PARMLIB in order to ensure that the VTEXMAIN module loads properly with respect to one other VTAM module. In order for VTEXMAIN to operate properly, it must be loaded into the LPA at an address which is lower than the VTAM module ISTAICIR. In most cases, the natural load order of the LPA modules works out so that this is the case, however, if module VTEXMAIN loads at a higher address than ISTAICIR, all VTAM applications will not accept logons when AUTOMON®-VTAM is active.

You can determine whether you need to order the LPA by listing the LPA directory and checking the addresses of these two modules. The JCL to list the LPA directory is as follows:

```
//STEP1 EXEC  PGM=AMBLIST
//SYSPRINT   DD    SYSOUT=*
//SYSIN      DD    *
LISTLPA
/*
```

Now, look at the alphabetic portion of the module name listing and locate the two modules, VTEXMAIN and ISTAICIR. Look at the EP ADDR (entrypoint address) of the two modules. The address for VTEXMAIN must be a smaller hexadecimal number than the address for ISTAICIR. (When comparing the addresses, ignore bit zero, the x'80' bit in the first byte).

If VTEXMAIN loads at a higher address than ISTAICIR, you must order the LPA. Do this as follows:

Edit member IEASYS00 in SYS1.PARMLIB. There must be a keyword in the form PAK=xx in this member. If there is not one, add it.

The xx is a two-character suffix code which corresponds to the last two characters of the active IEAPAK member. The IEASYS00 member contains many keywords such as:

REAL=xx,CLOCK=xx,CON=xx,MLPA=xx,RDE=xx.

Coding PAK=00, for instance, would mean that there is another member in SYS1.PARMLIB with a member name of IEAPAK00 which describes the load order for certain LPA modules.

The IEAPAKxx member contains one or more member names or sets of member names in parentheses. The standard member supplied with the IPO contains a single entry as follows:

(IEFBR14)

Add a comma after the last entry, then add the following:

(VTEXMAIN,ISTAICIR).

The standard IPO member, after modification, would look like:

(IEFBR14),(VTEXMAIN,ISTAICIR)

Now, save and exit the IEAPAKxx member. When you next IPL, the VTEXMAIN module will load into the LPA at a lower address than the ISTAICIR module.

[Note]: The procedure for ordering the LPA with IEAPAKxx is documented in the MVS Initialization and Tuning manual.

CREATION OF A CUSTOMIZATION TABLE

The customization table for AUTOMON@-VTAM is used to limit the applications and/or logical units which can participate in compression as well as to set the various operational options that determine the type and scope of compression that will be done. This is also where you define the destination for the compression statistics that AUTOMON@-VTAM will capture if you want to do statistics logging in addition to the on-line statistics displays. In addition, there is a VTAM trace feature which can be activated through the customization table, plus a facility for defining the datastream tokens that begin and end a PC file transfer operation.

While we don't anticipate that there will be problems, it is always a good idea to move slowly when installing a product which has system-wide ramifications, so we recommend that you set up a customization table now which limits AUTOMON@-VTAM to only one application and possibly even to just a few terminals.

The skeleton customization table that was loaded as part of the basic product install can be modified at this point and used for your test, or you may want to copy it into another member and use that one. The

member name is not important, since you specify the location and name of the customization table at execution time, however, to conform to the conventions used with PARMLIB statements, we suggest you use the first six characters, VEXPRS, as the member name and the last two characters as a suffix code.

Refer to section 03 - *CUSTOMIZATION* of this manual for an explanation of the various parameters of the customization table. Then edit the VTEXSKEL member (or a copy of it) and make the desired changes.

The VTEXSKEL member appears as follows:

```

*****
*
*           AUTOMON@-VTAM SAMPLE CUSTOMIZATION TABLE
*
*****
*
*****      System Options
*
          VSTART   LU=INCL                Include all LUs
          VSTART   MAXBUF=1024K           Max static storage = 1 Megabyte
          VSTART   LPADSN=SYS1.LPALIB     DSN where VTEXMAIN resides
          VSTART   SVC=xyy                number of IGC00xyy from install
          VSTART   STATMON=N              No CANDLE status monitor
*
*****      TSO Application, APPLID should be TSO
*
TSO       VAPPL   INCL                    Include this application
TSO       VAPPL   APTYPE=TSO              Application type
TSO       VAPPL   CPRIMG=Y                Compressed terminal images
TSO       VAPPL   DELAY=0                 No log-on delay
TSO       VAPPL   EWIMG=B                 Byte-level compression
TSO       VAPPL   EXTATTR=CMPRS          Compress extended attributes
TSO       VAPPL   INBND=Y                 Inbound compression
TSO       VAPPL   PSIZE=1920              Primary screen size
TSO       VAPPL   ASIZE=1920              Alternate screen size
TSO       VAPPL   RBGEN=Y                 Read-buffer generation
TSO       VAPPL   STATS=N                 No statistics logging
*
*****      CICS Application, provide your own APPLID
*
CICS     VAPPL   INCL                    Include this application
CICS     VAPPL   APTYPE=CICS              Application type
CICS     VAPPL   CPRIMG=Y                Compressed terminal images
CICS     VAPPL   DELAY=02                 Log-on delay = 2 sends
CICS     VAPPL   EWIMG=B                 Byte-level compression
CICS     VAPPL   EXTATTR=CMPRS          Compress extended attributes
CICS     VAPPL   INBND=Y                 Inbound compression
CICS     VAPPL   KEY=8                    Storage protect key
CICS     VAPPL   SCRFSIZE=CICS           Get screen sizes from CICS
CICS     VAPPL   RBGEN=Y                 Read-buffer generation
CICS     VAPPL   STATS=N                 No statistics logging
*
*****      IMS Application, provide your APPLID
*
IMS      VAPPL   INCL                    Include this application
IMS      VAPPL   APTYPE=IMS              Application type
IMS      VAPPL   CPRIMG=Y                Compressed terminal images

```

IMS	VAPPL	DELAY=0	No log-on delay
IMS	VAPPL	EWIMG=B	Byte-level compression
IMS	VAPPL	EXTATTR=CMPRS	Compress extended attributes
IMS	VAPPL	INBND=Y	Inbound compression
IMS	VAPPL	KEY=7	Storage Protect Key
IMS	VAPPL	PSIZE=1920	Primary screen size
IMS	VAPPL	ASIZE=1920	Alternate screen size
IMS	VAPPL	RBGEN=Y	Read-buffer generation
IMS	VAPPL	STATS=N	No statistics logging
*			
*****		OTHER Application, provide your APPLID	
*			
OTHER	VAPPL	INCL	Include this application
OTHER	VAPPL	APTYPE=OTHER	Application type
OTHER	VAPPL	CPRIMG=Y	Compressed terminal images
OTHER	VAPPL	DELAY=0	No log-on delay
OTHER	VAPPL	EWIMG=B	Byte-level compression
OTHER	VAPPL	EXTATTR=CMPRS	Compress extended attributes
OTHER	VAPPL	INBND=Y	Inbound compression
OTHER	VAPPL	PSIZE=1920	Primary screen size
OTHER	VAPPL	ASIZE=1920	Alternate screen size
OTHER	VAPPL	RBGEN=Y	Read-buffer generation
OTHER	VAPPL	STATS=N	No statistics logging
*			
*****		Default for all non-specified applications	
*			
\$DEFAULT	VAPPL	EXCL	Exclude all other applications
*			
*****		Individual or generic LUs to be included or excluded. If only these LUs are	
*		to be compressed, change the EXCL on the VLU to 'INCL' and change	
*		VSTART to LU=EXCL	
*			
*		Note, these examples illustrate wild-card coding, do not activate using	
*		these examples.	
*			
LUnicode01	VLU	EXCL	Fully qualified LUnicode
L???XX	VLU	EXCL	Exclude LUs with 'L' in pos.1,
*			'XX' in 5-6 & anything in 7-8
????++++	VLU	EXCL	Exclude all LUs with numerics in
*			pos. 5-8
? X	VLU	EXCL	Exclude all LUs with alpha in 2-3
*			'X' in 4 and anything in 1,5-8
LU??????	VLU	EXCL=TSO	For all LUs beginning with 'LU',
*			exclude when logged-on to TSO,
*			otherwise take action as coded
*			on VSTART LU= keyword.
LUnicode02	VLU	EXCL=(TSO,OTHER)	For this LU, exclude from TSO and
LUnicode02	VLU	INCL=(IMS,CICS)	OTHER, include in CICS and IMS.
*			
*****		Start/Stop token table for standard IBM file transfer	
*			
	VFTRAN	START=(C'IND\$',3)	IND\$ inbound at pos. 4
	VFTRAN	STOP=(C'INW0',2)	INW0 outbound at pos. 3
	VFTRAN	START=(C'IND\$',6)	IND\$ inbound at pos. 7
	VFTRAN	STOP=(C'ICH70001I',12)	ICH70001I outbound at 13
	VFTRAN	STOP=(C'INW0',9)	INW0 outbound at 10

```

*
*****      Sample VSKIP statements - Remove for production
*
      VSKIP      ROWCOL=(3,4),TEXT=C'ABCD'Skip if row3 col4=ABCD
      VSKIP      OFS=163,TEXT=C1C2C3C4      Skip if pos. 164=ABCD
      VSKIP      TYPE=IN,TEXT=C'ABCD'      Skip next output if first
                                                    input text = ABCD
*
*
*****      Sample VTRACE statements - Remove for production
*
LUNAME1      VTRACE                               Diagnostic trace for LUNAME1
LUNAME2      VTRACE      I/O                       I/O trace for LUNAME2

```

The distributed skeleton customization table provides a good working model for most installations. You should, however, read section 03 - *CUSTOMIZATION* carefully to be sure that you understand all of the options and that they will be correct for your installation.

If you want to use this same table structure, change the APPLIDs in position 1 of each VAPPL statement to your APPLIDs and tailor the VAPPL options, if desired. Note that in almost every case, the correct APPLID for TSO is 'TSO'. The name is taken from the first three characters of the ACBNAME of the APPL definitions in VTAM for TSO, which will be 'TSO' unless it has been changed.

If you are certain that the VTAM DLOGMOD designates the correct screen size for every terminal you can remove the PSIZE and ASIZE keywords. If you are not sure, go ahead and leave them as a default since the DLOGMOD specification will override PSIZE and ASIZE if it is present.

If you want to include only certain terminals, change the LU=INCL on the VSTART statement to LU=EXCL, then change LUNAME1, LUNAME2 and LUNAME3 to your logical unit names and add any others desired. Specify INCL on the VLU statements. Once you have completed evaluation and want to open it up for all users, change the VSTART back to LU=INCL. If you want to include all applications you can either insert a VAPPL statement for each application or change the \$DEFAULT VAPPL statement to INCL. It is better to create a VAPPL for each included application as that will allow you to view statistics by application. Also, some of the VAPPL options are application dependent.

If you never do any file transfers to PCs, remove the VFTRAN statements. If you have file transfer packages other than or instead of the IBM package, you will, at some point, need to code VFTRAN statements to define that package. Refer to the discussion of the VFTRAN statement in section 03 - *CUSTOMIZATION* for details.

[Note]: If you code INCL on the \$DEFAULT statement and do not have a VAPPL statement for each application participating in compression, the on-line statistics display will total all statistics by application into the \$DEFAULT application.

[Note]: If you have a VTAM session manager, refer to *COMPRESSION FOR VTAM SESSION MANAGERS* in section 03 - *CUSTOMIZATION* for a discussion of the considerations thereof.

[Note]: If you are running MRO or ISC in CICS, the rule for including or excluding applications is:

If the terminals are connected locally to the application, that application should be included for compression. If the terminals are remote to the application, exclude the application.

Thus, the TOR of a CICS MRO/ISC system would be included but the remote applications would not.

If you are running VTAM cross-domain, the rule is:

AUTOMON®-VTAM must be installed and activated on the system where the applications to be included for compression reside.

Thus, for cross-domain, AUTOMON®-VTAM would run on the system with the applications, and those applications would be included for compression.

MINIMUM TERMINAL INPUT SIZES

This subject is covered as one of the installation steps. However, it is re-documented here for more permanent reference. This is an important part of the installation of AUTOMON®-VTAM and must not be overlooked.

Changes to application system terminal definitions may be required. If CICS is in use, there are almost always changes required to the CICS TCT and the SIT. If inbound compression is to be used, VTAM applications must provide a terminal I/O area large enough for the complete input datastream. In CICS, for instance, it is possible to specify a minimum TIOA size which is smaller than many input datastreams. This causes CICS to perform multiple VTAM receive requests in order to accept an input datastream.

With AUTOMON®-VTAM, in order for inbound compression to work properly, the minimum TIOA length must be large enough to contain the entire TIOA. The CICS parameters that are affected are 'RAMAX', in the system initialization table (DFHSIT), and 'TIOAL' in a macro TCT or 'IOAREALEN' in RDO. TIOAL and IOAREALEN apply to conversational tasks, while RAMAX sets the maximum length for VTAM 'receive-any' requests (unsolicited input).

'RAMAX' should be set to at least 1536 for model-2 (24x80) terminals. A safer value is 2048. If you have large-screen terminals, the value may need to be set higher still. 'TIOAL' (or 'IOAREALEN' is usually sufficient at 1536. 'RAMAX' is often not even coded in DFHSIT. If this is the case, the default value is 256, which is always too small for AUTOMON®-VTAM.

If you specify the customization option 'RBGEN' (read-buffer generation), which causes AUTOMON®-VTAM to suppress read-buffer commands and generate the buffer from the terminal image, you will need to add another 512 bytes to the value coded for TIOAL or IOAREALEN.

A recommended value for each terminal model is:

Model-2 (24x80)	- 2560	Model-3 (32x80)	- 3328
Model-4 (43x80)	- 4096	Model-5 (27x132)	- 4352

IMS has similar parameters that control the input area sizes. They are 'RECANY' in the 'COMM' macro and 'BUFSIZE' on the 'LINE' macro. The IBM defaults for these parameters seem to generally be adequate, although it is possible that they would need to be increased also. TSO has no problem with input areas (at least none have been encountered) but other VTAM applications may have similar considerations and should be checked. If the input area is too small to contain the entire input datastream, you will receive the following message at the terminal when the transaction generating the datastream is executed:

VMAIN015. BUFFER OVERRUN HAS OCCURRED, INPUT TIOA AREA SIZE MUST BE INCREASED

SNA RU SIZE CONSIDERATIONS

For SNA terminals, you should consider increasing the Request Unit (RU) size of the terminals in the application systems if you are using AUTOMON@-VTAM for IMS systems. AUTOMON@-VTAM performs its outbound image saving at the RU level. This means that if the RU size is smaller than the entire outbound datastreams, EXPRESS does not see all of the data going to the terminal at one time. It must construct the terminal image one RU at a time.

AUTOMON@-VTAM cannot perform outbound image compression when multiple request units are sent for one datastream. The result is that data will be sent to the terminal that does not need to be sent any time multiple request units are used for one datastream.

This situation is not a problem for CICS because it uses a technique called 'buffer lists' to send multiple request units. All of the request units are presented to VTAM at one time with a buffer list which points to each unit. Therefore, all request units are available to AUTOMON@-VTAM to construct the terminal image.

IMS, on the other hand, physically breaks up the outgoing datastreams into multiple request units and sends them one at a time, each with a separate VTAM send request. Tests with IMS have shown that by increasing the RU size from 1500 to 2500 bytes results in as much as a 40% improvement in outbound compression savings.

Another argument for larger RU sizes is the overall improvement in performance, even without AUTOMON@-VTAM in the picture. When datastreams are broken down into multiple request units, a single write to the terminal can result in several VTAM send requests. This significantly increases the path length of the transaction, resulting in additional CPU overhead and a corresponding slow-down in response time. And while it is true that increasing the RU size will require larger VTAM buffers for each send, this is dynamic storage that is released at the completion of the request.

We recommend that you increase your request unit size to the largest heavily-used datastream size if you are using IMS. You can use the IMS trace or the AUTOMON@-VTAM trace to obtain a sampling of typical datastreams in order to determine the optimum size. The parameter which controls the outbound RU size in IMS is the OUTBUF keyword on the TERMINAL macro.

RU sizes are not a consideration for TSO and CICS, but they could be for other VTAM applications. If you have other VTAM applications besides CICS and TSO and you are getting outbound compression statistics that are less than you expected, you should consider this option as a possible solution.

DOS/VSE INSTALLATION - BASIC PRODUCT INSTALL

Installation consists of the following steps:

- 1). Link-edit the four program modules from file 1 of the tape.
- 2). Run the STSINST program to print the full installation instructions and create the installation JCL.
- 3). Tailor the installation JCL for your library names.
- 4). Run the installation JCL to load all programs and files.
- 5). Define the required CICS table entries.

6). Make any required changes to your application terminal tables.

For DOS/VSE, you must prepare initial JCL for two steps:

Step 1. Load the installation programs and password table

The following JCL can be used to install the first file of the tape for DOS/VSE users:

```
* $$ JOB STSINST,,C
// JOB LOAD STSINST
// ASSGN SYSIPT,XXX [Note 1]
// MTC REW,SYSIPT
// LIBDEF PHASE,CATALOG=L.S [Note 2]
// OPTION CATAL
  INCLUDE
// EXEC LNKEDT
// RESET SYSIPT
/*
```

[Notes]:

1. Assign SYSIPT to a tape drive that can read the BPI of your installation tape.
2. This LIBDEF is for DOS/VSE/SP, where:
L.S = Library, Sub-library

The library for the install program, password processor and password table should be the same library where VTEXPRES is to reside.

Step 2. Print installation instructions and punch installation JCL

The following JCL executes the STSINST program to print the installation documentation and punch the install JCL:

```
// ASSGN SYS011,XXX [Note 1]
// LIBDEF *,SEARCH=L.S [Note 2]
// EXEC STSINST
PRODUCT=EXPRESS [Note 3]
MODE=PRINT [Note 3]
OPSYS=DOS [Note 3]
LINES=56 [Note 3]
JCL=INSTALL [Note 3]
/*
/&
* $$ EOJ
```

[Notes]:

1. Assign SYS011 to a tape drive that can read the BPI of your installation tape.
2. The LIBDEF must identify the library where STSINST was link-edited as a search library.
L.S = Library, Sub-library.
3. See *KEYWORDS OF THE UNICOM SYSTEMS INC. INSTALLATION PROGRAM*, later in this section, for the meaning of these keywords.

At this point, the full installation instructions will be printed. The installation JCL will be punched into the POWER punch queue. Retrieve or view the printed installation documentation and follow the instructions provided there to complete the installation of AUTOMON®-VTAM.

CREATION OF A CUSTOMIZATION TABLE

The customization table for AUTOMON®-VTAM is used to limit the applications and/or logical units which can participate in compression as well as to set the various operational options that determine the type and scope of compression that will be done. This is also where you define the destination for the compression statistics that AUTOMON®-VTAM will capture if you want to do statistics logging in addition to the on-line statistics displays. In addition, there is a VTAM trace feature which can be activated through the customization table, plus a facility for defining the datastream tokens that begin and end a PC file transfer operation.

While we don't anticipate that there will be problems, it is always a good idea to move slowly when installing a product which has system-wide ramifications, so we recommend that you set up a customization table now which limits AUTOMON®-VTAM to only one application and possibly even to just a few terminals.

The skeleton customization table that was loaded as part of the basic product install can be modified at this point and used for your test. Use the GETP command of ICCF, or the equivalent command of your text editor to retrieve the VTEXSKEL member from the punch queue and load it into an editor member. The member name is not important, since you include the customization table as a SYSIPT file at execution time, however, to conform to the conventions used with PARMLIB statements, we suggest you use the first six characters, VEXPRS, as the member name and the last two characters as a suffix code.

Refer to section 03 - *CUSTOMIZATION* of this manual for an explanation of the various parameters of the customization table. Then edit the VTEXSKEL member (or a copy of it) and make the desired changes.

The VTEXSKEL member appears as follows:

```

*****
*
*          AUTOMON®-VTAM SAMPLE CUSTOMIZATION TABLE
*
*****
*
*****      System Options
*
          VSTART    LU=INCL                Include all LUs
          VSTART    MAXBUF=1024K           Max storage = 1 Megabyte
          VSTART    STATMON=N              No CANDLE status monitor
*
*****      CICS Application, provide your CICS APPLID
*
CICS      VAPPL    INCL                    Include this application
CICS      VAPPL    APTYPE=CICS              Application type
CICS      VAPPL    CPRIMG=Y                 Compressed terminal images
CICS      VAPPL    DELAY=02                 Log-on delay = 2 sends
CICS      VAPPL    EWIMG=B                  Byte-level compression
CICS      VAPPL    EXTATTR=CMPRS           Compress extended attributes
CICS      VAPPL    INBND=Y                  Inbound compression
CICS      VAPPL    KEY=8                    Storage protect key
CICS      VAPPL    SCRFSIZE=CICS           Get screen sizes from CICS
CICS      VAPPL    RBGEN=Y                  Read-buffer generation

```

```

CICS          VAPPL      STATS=N          No statistics logging
*
*****
OTHER Application, provide your APPLID
*
OTHER        VAPPL      INCL          Include this application
OTHER        VAPPL      APTYPE=OTHER Application type
OTHER        VAPPL      CPRIMG=Y       Compressed terminal images
OTHER        VAPPL      DELAY=0       No log-on delay
OTHER        VAPPL      EWIMG=B       Byte-level compression
OTHER        VAPPL      EXTATTR=CMPRS Compress extended attributes
OTHER        VAPPL      INBND=Y       Inbound compression
OTHER        VAPPL      PSIZE=1920    Primary screen size
OTHER        VAPPL      ASIZE=1920    Alternate screen size
OTHER        VAPPL      RBGEN=Y       Read-buffer generation
OTHER        VAPPL      STATS=N       No statistics logging
*
*****
Default for all non-specified applications
*
$DEFAULT     VAPPL      EXCL          Exclude all other applications
*
*****
Individual or generic LUs to be included or excluded. If only these LUs are
*
to be compressed, change the EXCL on the VLU to 'INCL' and change VSTART to
LU=EXCL.
*
LName01     VLU        EXCL          Fully qualified LName
L???XX      VLU        EXCL          Exclude LUs with 'L' in pos. 1
*
'XX' in 5-6 & anything in
*
2-4 & 5-8
???++++     VLU        EXCL          Exclude all LUs with numerics
*
in pos. 5-8
?||X       VLU        EXCL          Exclude all LUs with alpha in
*
2-3, 'X' in pos. 4 and
*
anything in pos. 1, 5-8
LU?????    VLU        EXCL=CICS1   For all LUs beginning with 'LU',
*
exclude when logged-on to CICS1,
*
otherwise take action as coded
*
on VSTART LU= keyword.
LName02     VLU        EXCL=(CICS1,CICS2) For this LU, exclude from
LName02     VLU        INCL=(CICS3,OTHER) compression in CICS1 and
*
CICS2, Include in compression
*
in CICS3 and OTHER.
*****
Start/Stop token table for standard IBM file transfer
*
VFTRAN      START=(C'IND$',3)    IND$ inbound at pos. 4
VFTRAN      STOP=(C'INW0$',2)  INW0 outbound at pos. 3
VFTRAN      START=(C'IND$',6)  IND$ inbound at pos. 7
VFTRAN      STOP=(C'ICH70001I',12) ICH70001I outbound at 13
VFTRAN      STOP=(C'INW0',9)   INW0 outbound at pos. 10
*
*****
Sample VSKIP statements - Remove for production
*
VSKIP       ROWCOL=(3,4),TEXT=C'ABCD Skip if row 3 col 4=ABCD
VSKIP       OFS=163,TEXT=C1C2C3C4 Skip if pos. 164=ABCD
VSKIP       TYPE=IN,TEXT=C'ABCD Skip next output if first

```

```

*                                     input text=ABCD
*
*****      Sample VTRACE statements - Remove for production
*
LUname1      VTRACE                      Diagnostic trace for LUname1
LUname2      VTRACE I/O I/O trace for LUname2

```

The distributed skeleton customization table provides a good working model for most installations. You should, however, read section 03 - *CUSTOMIZATION* carefully to be sure that you understand all of the options and that they will be correct for your installation.

If you want to use this same table structure, change the APPLIDs in position 1 of each VAPPL statement to your APPLIDs and tailor the VAPPL options, if desired.

If you are certain that the VTAM DLOGMOD designates the correct screen size for every terminal you can remove the PSIZE and ASIZE keywords. If you are not sure, go ahead and leave them as a default since the DLOGMOD specification will override PSIZE and ASIZE if it is present.

If you want to only include certain terminals, change the LU=INCL on the VSTART statement to LU=EXCL, then change LUname1, LUname2 and LUname3 to your logical unit names and add any others desired. Specify INCL on the VLU statements. Once you have completed evaluation and want to open it up for all users, change the VSTART back to LU=INCL. If you want to include all applications you can either insert a VAPPL statement for each application or change the \$DEFAULT VAPPL statement to INCL. It is better to create a VAPPL for each included application as that will allow you to view statistics by application. Also, some of the VAPPL options are application dependent.

If you never do any file transfers to PCs, remove the VFTRAN statements. If you have file transfer packages other than or instead of the IBM package, you will, at some point, need to code VFTRAN statements to define that package. For details, refer to the discussion of the VFTRAN statement in section 03 - *CUSTOMIZATION*.

[Note]: If you code INCL on the \$DEFAULT statement and do not have a VAPPL statement for each application participating in compression, the on-line statistics display will total all statistics by application into the \$DEFAULT application.

[Note]: If you have a VTAM session manager, refer to *COMPRESSION FOR VTAM SESSION MANAGERS* in section 03 - *CUSTOMIZATION* for a discussion of the considerations thereof.

[Note]: If you are running MRO or ISC in CICS, the rule for including or excluding applications is:

If the terminals are connected locally to the application, that application should be included for compression. If the terminals are remote to the application, exclude the application.

Thus, the TOR of a CICS MRO/ISC system would be included but the remote applications would not.

If you are running VTAM cross-domain, the rule is:

AUTOMON®-VTAM must be installed and activated on the system where the applications to be included for compression reside.

Thus, for cross-domain, AUTOMON®-VTAM would run on the system with the applications, and those applications would be included for compression.

LOAD THE SVA MEMBER

Since AUTOMON®-VTAM runs in the Shared Virtual Area you must load the VTEXMAIN program into the SVA before AUTOMON®-VTAM can be activated. You should add VTEXMAIN to the list of programs following the SET SDL statement in the ASI procedure so that it will always be loaded into the SVA when an IPL occurs.

Having added it to the SET SDL list, you could IPL now to load it. However, that is not necessary. You can run the following JCL to add it to the SVA: [Note 1]

```
* $$ JOB SVALOAD,,BG [Note 2]
//JOB SVALOAD
SET SDL
VTEXMAIN,SVA
/*
/&
* $$ EOJ
```

AUTOMON®-VTAM does not automatically activate when IPL occurs. You must activate it by executing the VTEXPRES program as a batch program. The SVA load must be done before you attempt to activate it.

Once the VTEXMAIN module is loaded into the SVA, AUTOMON®-VTAM is installed. Now refer to the *OPERATION* section of this manual for instructions on activating, deactivating and controlling the product. [Note 3]

[Notes]:

1. If you need to reload the VTEXMAIN module into the SVA after activating it, refer to *RELOADING THE VTEXMAIN MODULE IN THE SVA* in section 05 before attempting it.
2. This job must run in the background partition.
3. If you have any problems getting started, refer to section 09 - *PROBLEM ANALYSIS AND DIAGNOSIS* for an overview of common problems and trouble-shooting suggestions.

MINIMUM TERMINAL INPUT SIZES

This subject is covered as one of the installation steps. However, it is re-documented here for more permanent reference. This is an important part of the installation of AUTOMON®-VTAM and must not be overlooked.

Changes to application system terminal definitions may be required. If CICS is in use, there are almost always changes required to the CICS TCT and the SIT. If inbound compression is to be used, VTAM applications must provide a terminal I/O area large enough for the complete input datastream. In CICS, for instance, it is possible to specify a minimum TIOA size which is smaller than many input datastreams. This causes CICS to perform multiple VTAM receive requests in order to accept an input datastream.

With AUTOMON®-VTAM, in order for inbound compression to work properly, the minimum TIOA length must be large enough to contain the entire TIOA. The CICS parameters that are affected are 'RAMAX', in the system initialization table (DFHSIT), and 'TIOAL' in a macro TCT or 'IOAREALEN' in RDO. TIOAL and IOAREALEN apply to conversational tasks, while RAMAX sets the maximum length for VTAM 'receive-any' requests (unsolicited input).

'RAMAX' should be set to at least 1536 for model-2 (24x80) terminals. A safer value is 2048. If you have large-screen terminals, the value may need to be set higher still. 'TIOAL' (or 'IOAREALEN' is usually sufficient at 1536. 'RAMAX' is often not even coded in DFHSIT. If this is the case, the default value is 256, which is always too small for AUTOMON®-VTAM.

If you specify the customization option 'RBGEN' (read-buffer generation), which causes AUTOMON®-VTAM to suppress read-buffer commands and generate the buffer from the terminal image, you will need to add another 512 bytes to the value coded for TIOAL or IOAREALEN.

A recommended value for each terminal model is:

Model-2 (24x80)	-	2560	Model-3 (32x80)	-	3328
model-4 (43x80)	-	4096	Model-5 (27x132)	-	4352

Other VTAM applications may have similar considerations and should be checked. If the input area is too small to contain the entire input datastream, you will receive the following message at the terminal when the transaction generating the datastream is executed:

VMAIN015. BUFFER OVERRUN HAS OCCURRED, INPUT TIOA AREA SIZE MUST BE INCREASED

NON-CICS APPLICATIONS

If you have any VTAM application other than CICS which is accessed by SNA terminals, you may need to consider increasing the outbound request unit (RU) size for that application. If CICS is your only application there is no problem, but if not, please read *SNA RU SIZE CONSIDERATIONS* in the *MVS INSTALLATION* section for discussion of the ramifications of using RU sizes that are too small.

KEYWORDS OF THE UNICOM SYSTEMS INC.INSTALLATION PROGRAM

The installation program, STSINST, which is provided on the first file of the installation tape can be used to print the documentation or perform a complete install for any of the products on the master installation tape.

The individual installation instructions of each product provide the necessary JCL and keyword parameters that are specific to that product.

This section describes all of the available keywords of the STSINST program, plus the coding format used.

FORMAT OF THE STSINST KEYWORDS

All keywords and operands of the STSINST program are coded in the SYSIN (MVS) or SYSIPT (DOS) dataset. Each keyword and operand occupies one statement. You cannot concatenate multiple keywords on one line. Each keyword must begin in position one.

Following are all of the possible keywords and operands of the STSINST program, listed in alphabetical order. A detailed explanation of each keyword and operand follows.

Underlined values listed are the default values if the associated operand is omitted. Values separated by a bold vertical bar (|) indicate mutually exclusive values which may be coded.

KEYWORD DESCRIPTIONS

- DIRECTORY** The **DIRECTORY** keyword requires no operands. If present, it will print a directory of the tape contents.
- CICS=1.6|1.7|2.1** This is the CICS release level. This only applies to CICS products. Valid entries are 1.6 and 1.7 for DOS/VSE, 1.6, 1.7 and 2.1 for MVS. If your CICS release has a third-level modification level, such as 1.6.1, or 2.1.1, specify only the first and second levels.
- FSRVFILE=xxxxxx** The **FSRVFILE** keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-FILESERV. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).
- HELPSCRN=xxxxxx** The **HELPSCRN** keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM screen file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).
- HELPTEXT=xxxxxx** The **HELPTEXT** keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM text file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).
- JCL=INSTALL|REINSTALL** The **JCL** keyword defines the action to be taken for the installation job control statements where VSAM files are involved. Valid operands are:
- | | |
|------------------|---|
| INSTALL | Create JCL which <u>will</u> delete and redefine the VSAM file(s). |
| REINSTALL | Create JCL which will neither delete nor redefine the VSAM file(s). |
- If the **JCL** keyword is omitted, the installation JCL will be printed only. If either **INSTALL** or **REINSTALL** is specified, the JCL will be punched to the POWER punch queue (DOS) or written to the PDS member identified by the SYSPCH DD statement (MVS).
- JUGLFILE=xxxxxx** The **JUGLFILE** keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-JUGGLER or CICS-JUGGLER/SVT. The xxxxxx is the 1-7 byte file ID of the DLBL statement (DOS) or 1-8 byte DDNAME of the DD statement (MVS).
- LINES=56|nn** The **LINES** keyword defines the number of lines to print on the system printer before skipping to a new page. If omitted, 56 is the default.
- MODE=PRINT|CREATE|INSTALL|REINSTALL|DOCUMENT**
- The **MODE** keyword defines the action to be taken by STSINST. Valid mode operands are:

- PRINT** Print the installation instructions for the specified product.
- CREATE** Create an installation file on the work tape for the specified product.
- INSTALL** Load the VSAM files associated with the specified product. Coding **MODE=INSTALL** indicates that this is a new installation, loading to an empty VSAM file.
- REINSTALL** Load the VSAM files associated with the specified product. **MODE=REINSTALL** indicates that this is a subsequent installation, reloading the system records to an existing VSAM file that contains user data.
- DOCUMENT** Print the overview documentation for the specified product. The overview document is a short preview of the product, telling you how to get started using it. If you need the complete reference guide, that can be obtained from your sales representative.

OPSYS=DOS|MVS|MVS/XA|MVS/SP

The **OPSYS** keyword defines the operating system where the specified product is to reside. For all products except **AUTOMON@-VTAM**, you need only to specify **DOS** or **MVS**. For the **MVS** version of **EXPRESS** you must specify either **MVS/XA** or **MVS/SP**. For **MVS/ESA**, specify **MVS/XA**.

PRODUCT=WINDOWS|JUGGLER|SVT|HELP|FILESERV|VWINDOWS|EXPRESS

The **PRODUCT** keyword identifies the product to be installed, or for which documentation is to be printed. A brief description of each product is contained in Appendix C.

The full product name corresponding to each keyword is:

EXPRESS	-	AUTOMON@-VTAM
VWINDOWS	-	VTAM-WINDOWS
WINDOWS	-	AUTOMON@-VTAM
JUGGLER	-	CICS-JUGGLER
SVT	-	CICS-JUGGLER/SVT
HELP	-	HELP-WINDOWS
FILESERV	-	CICS-FILESERV

WNDOWFILE=xxxxx The **WNDOWFILE** keyword identifies the name of the **DLBL** or **DD** statement in the job control statements for this execution of **STSINST** that names the **VSAM** file for **AUTOMON@-VTAM**. The **xxxxxx** is the file ID of the **DLBL** statement (**DOS**) or **DDNAME** of the **DD** statement (**MVS**).

VTWOWFILE=xxxxxxx The **VTWOWFILE** keyword identifies the name of the **DLBL** or **DD** statement in the job control statements for this execution of **STSINST** that names the **VSAM** file for **AUTOMON@-VTAM**. The **xxxxxxx** is the file ID of the **DLBL** statement (**DOS**) or **DDNAME** of the **DD** statement (**MVS**).

INSTALLING THE PRODUCT CONTROL PASSWORD

The product control password is a special code which controls the authorized use of any product from Unicom Systems Inc. Systems. During the trial evaluation period, the product control password defines the date at which the trial version of the product will expire and can no longer be used.

For permanent licensed users, the product password defines all of the CPUs where the product may be used. Passwords for lease or lease-purchase licenses contain an expiration date and CPU ID.

When you initially receive an installation tape, whether for a trial evaluation or not, the tape contains a temporary password that will keep the product from expiring for approximately 30 days. If you are a licensed user, part of the re-install procedure is to assemble your permanent password table after installing the new tape. If you already have a permanent password in place for the product, there is nothing further to be done.

If you are evaluating the product and have received a new temporary password from your sales representative, you must assemble the password table with the new password to keep the product from expiring.

The following job control is used to assemble and catalog the password control table, STSPASS.

DOS/VSE USERS

```
// JOB PASSWORD ASSEMBLY
// LIBDEF PHASE,CATALOG=L.S           [Note 1]
// OPTION CATAL
  PHASE STSPASS,*   [Note 2]
// EXEC ASSEMBLY
      DC    CL6'000000'           CPU ID
      DC    CL4'xxxx'           Product ID
      DC    CL6'mmddy'         Expiration Date
      DC    CL16'xxxxxxxxxx'    Password
      .
      .
      .
      DC    CL6'000000'           CPU ID
      DC    CL4'xxxx'           Product ID
      DC    CL6'mmddy'         Expiration Date
      DC    CL16'xxxxxxxxxx'    Password
      END
/*
// EXEC LNKEDT
/&
```

MVS USERS

```

//PASSTBL    JOB    1.'ACCOUNT DATA',MSGCLASS=X
//ASM        EXEC          PGM=IEV90,PARM='DECK'
//SYSPRINT   DD    SYSOUT=*
//SYSLIN     DD    DUMMY
//SYSUT1     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH DD    UNIT=SYSDA,DISP=(,PASS),
//           SPACE=(CYL,(1,1)),
//           DSN=&&TEMP1,
//           DCB=BLKSIZE=3200
//SYSIN      DD    *
              DC    CL6'000000'                CPU ID
              DC    CL4'xxxx'                  Product ID
              DC    CL6'mmddy'                 Expiration Date
              DC    CL16'xxxxxxxxxx'           Password
              .
              .
              .
              DC    CL6'000000'                CPU ID
              DC    CL4'xxxx'                  Product ID
              DC    CL6'mmddy'                 Expiration Date
              DC    CL16'xxxxxxxxxx'           Password
              END
/*
//LINK       EXEC    PGM=IEWL,PARM=(LIST,XREF),
//           COND=(4,LT)
//SYSLIB     DD    DUMMY
//SYSPRINT   DD    SYSOUT=*
//SYSLIN     DD    DSN=&&TEMP1,
//           DISP=(OLD,DELETE)
//SYSLMOD    DD    DSN=XXX.XXX(STSPASS),       [Note 2]
//           DISP=SHR                           [Note 1]
//SYSUT1     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))

```

[Notes]:

1. The library containing the link-edited module must be available to AUTOMON@-VTAM.
2. The link-edited module name must be STSPASS

DATA FIELDS OF THE PASSWORD CONTROL

TABLE

The password table is a series of define constant instructions. The same password table is used for all products from Unicom Systems Inc. Systems.

There are four DC statements required for each product, or for each occurrence of the same product on multiple CPUs. There is no limit to the number of product entries that may be present.

The four DC instructions define the following values:

- 1). The first entry in the table is a 6-position CPU ID. For temporary passwords, code this as all zeros. For licensed users, code the full 6-position CPU ID where the product will operate.

If you are a VM user and have multiple guest machines on the same CPU, you can get by with a single entry if the last four positions of the CPU ID on each guest machine is the same. If this is the case, code zero as the first two digits, followed by the four identifying digits. If each guest CPU ID is different, you must make an entry for each unique CPU ID.

- 2). The second entry is a 4-position product ID. Codes are:

EXPR	-	AUTOMON®-VTAM
VTW	-	VTAM-WINDOWS
WNDO	-	AUTOMON®-VTAM
JUGL	-	CICS-JUGGLER
SVT	-	CICS-JUGGLER/SVT
HELP	-	HELP-WINDOWS
FSRV	-	CICS-FILESERV

- 3). The third entry is the expiration date. For temporary passwords or for lease and lease-purchase licenses, you will be given an expiration date along with your password. That date must be coded here in MMDDYY format. For perpetual licenses, code this as all zeros.
- 4). The last entry is the password itself. You must define it as a 16-byte field, although the total number of characters in the password may not reach 16. Code the password that you have been given.

CUSTOMIZATION

A customization table is required for operation with AUTOMON®-VTAM. The table is used for the following purposes:

- 1). Limit the applications and/or logical units which can participate in compression.
- 2). Define the type and scope of compression that will be done by coding the various application options that are available.
- 3). Establish the destination for compression statistics recording.
- 4). Identify specific transaction datastreams to be excluded from compression.
- 5). Activate the VTAM trace feature for specified logical units.
- 6). Define the datastream tokens that begin and end a PC file transfer operation.

The customization table is a series of command statements contained in a PDS member in MVS and a text-editor member in DOS/VSE. The table is not assembled, it is interpreted at execution time.

CODING THE CUSTOMIZATION TABLE

MVS Users:

Using PDF or some other editor, create a member in a PDS which has 80-byte records. Or you may edit the VTEXSKEL member that was loaded during installation. To be consistent with this manual, you should name your source member VEXPRS00, however it does not matter what name you use or what PDS is used since you must inform the VTEXPRES program of the location and name of the customization table whenever it is executed.

DOS/VSE Users:

Using ICCF or some other editor, create a new member. Or you may edit the VTEXSKEL member that was loaded during installation. You should name your source member VEXPRS00 to be consistent with this manual, however it does not matter what name you use since you must include the member as input to the VTEXPRES program whenever it is executed.

CODING FORMAT

The statements are coded exactly like Assembler macro statements with one exception. Continuation in column 72 is not allowed, although you can string operands together on one line, separated by a comma, continuing up to position 72. You can code multiple statements using the same operator in order to use all of the operands which may be needed. When you continue statements, the label field (position 1-9) of each statement must contain the same data.

For example, the following statement:

```
VSTART LU=INCL,PASSWORD=XYZ
```

is equivalent to:

```
VSTART LU=INCL  
VSTART PASSWORD=XYZ
```

Likewise,

```
CICS1      VAPPL  INCL,INBND=Y,EWIMG=B
```

is equivalent to:

```
CICS1      VAPPL  INCL
CICS1      VAPPL  INBND=Y
CICS1      VAPPL  EWIMG=B
```

It does not matter how many statements with the same label and operator are coded, nor is the sequence important.

Once you have the table coded, you are ready to execute VTEXPRES. There is no assembly required.

Comments on the statements are allowed. There must be at least 5 spaces after the end of the statement before the comment begins.

CUSTOMIZATION STATEMENTS

There are six statement operators that can be coded. They are:

```
VSTART - Establish initial options.
VAPPL  - Identify applications for exclusion or inclusion.
VLU    - Identify logical units (terminals) for exclusion or inclusion.
VSKIP  - Identify specific output datastreams to be excluded from
         compression.
VFTRAN - Identify file transfer start/stop tokens.
VTRACE - Activate tracing for specified logical units (terminals).
```

Following is a description of each of the customization statements and their associated operands.

Underlined values listed are the default values if the associated operand is omitted. Values separated by a bold vertical bar (|) indicate mutually exclusive values which may be coded.

RULES OF CODING THE CUSTOMIZATION TABLE

The general rules of coding the customization table are as follows:

- 1). VSTART statements must appear first.
- 2). VAPPL statements must occur prior to the VLU statements.
- 3). All other statements may occur in any sequence, the only stipulation being that all entries of the same type must appear consecutively together.

As long as the above three rule are adhered-to, the statements may be arranged in any sequence; however, the preferred order is as follows:

- 1). VSTART
- 2). VAPPL
- 3). VLU
- 4). VSKIP
- 5). VTRACE
- 6). VFTRAN

VSTART - ESTABLISH INITIAL OPTIONS

Following are all of the possible operands of the VSTART statement;

```
VSTART  LU=INCL | EXCL,  
        LPADSN=SYS1.LPALIB | dataset name,  
        MAXBUF=nnnnnnK,  
        PASSWORD=symbol,  
        SMFID=nn,  
        SMFUSER=nn,  
        SMFSVC=nn,  
        STATMON=N | Y,  
        SVC=nnn
```

LPADSN (MVS only) The LPADSN keyword specifies the dataset name where the VTEXMAIN module resides. If this keyword is omitted, the default is 'SYS1.LPALIB'.

LU The LU keyword establishes the default for all logical units which are not specifically defined with a VLU statement.

Code INCL (or omit the LU keyword) if you want all logical units not identified with a VLU statement to be included in compression. If omitted, INCL is the default; meaning all LU's will be included unless VLU statements are used to exclude them.

Code EXCL if you want all logical units not identified with a VLU statement to be excluded from compression. All LU's will be excluded unless VLU statements are used to include them.

If selective inclusion/exclusion by APPLID is designated on a VLU statement, this keyword establishes the default action to be taken if the terminal logs onto an application that is not specified in the selection list. (See VLU statement.)

MAXBUF The MAXBUF keyword establishes the upper limit for static storage usage by AUTOMON@-VTAM. Static storage is storage which is long-term in duration. That is, it does not get released at the end of a request but usually remains until the operator logs off. The three main control blocks which tie up static storage are terminal control blocks, terminal images and vectors.

When you code MAXBUF, you set a limit on the amount of storage that AUTOMON@-VTAM can obtain for static use. When the limit is reached, compression is automatically

stopped for the request which exceeds the limit. If usage drops below the limit, subsequent requests will automatically compress again.

Code the storage limit value as a number followed by 'K', which designates storage in increments of 1024 bytes. For example, to limit the maximum static storage to two megabytes, code the value as

MAXBUF=2048K.

The maximum value that can be specified is 999,999K. If MAXBUF is omitted, the only limit is the available storage in the system.

- PASSWORD The PASSWORD keyword defines a master password that, if specified, will be required for anyone attempting to perform any of the following functions:
- The ACTIVATE, DEACTIVATE or BACKOUT command from the TSO command screen.
 - The STOP SYSTEM, STOP APPL or STOP LU function from the CICS VTEX transaction or the TSO application display.
 - Change any of the configuration values on the CICS VTEX transaction or the TSO application display.
 - Reset the application statistics to zero from the CICS VTEX transaction or the TSO application display.

The password may be one to eight characters in length and contain any alphanumeric characters. If coded, the password must be supplied any time one of the functions listed above is attempted on-line.

If PASSWORD is omitted, any of the functions listed above may be performed with no security checking.

PASSWORD applies when VTEXPRES is executed as a TSO command or when the VTEX transaction is used in CICS. It does not apply when VTEXPRES is executed as a batch program or started PROC.

SMFID (MVS only) If SMF is to be used for statistics recording rather than a sequential file, you must supply the SMF user record type via the SMFID keyword.

The SMFID is coded as a decimal integer and should be unique among other SMF user record types in order to facilitate record recognition in your SMF print extract.

If SMFID is omitted, STATS=Y is coded on a VAPPL statement and there is no STATDDN keyword coded on a VAPPL statement, statistics records will be written to SMF with a record type of zero.

SMFUSER (MVS only) If you wish to emit a user identification number into the SMF record, you may code it as a decimal integer on the SMFUSER keyword. This keyword only applies if SMFID is coded. If coded, the number will be placed in the SMF record at offset 18 decimal, (12 hex).

If SMFUSER is omitted, no user field will be present in the SMF record. It will contain zeros.

SMFSVC (MVS only) You may, if desired, code a user SVC routine to write the SMF record or process it in some way. If this option is selected, the user's SVC number must be coded as a decimal number on the SMFSVC keyword. Upon entry to the SVC routine, register 1 will contain the address of the SMF record, the format of which is described in section 06 - *STATISTICS CAPTURE AND REPORTING*.

If SMFSVC is coded, your user SVC routine is responsible for writing the SMF record.

If SMFSVC is omitted, the SMF statistics record will be written as a standard SMF record.

[Note]: The procedure for writing a user SVC routine is described in the MVS MACROS AND FACILITIES manual, volume 1.

[Note]: For applications using authorized path, such as TSO, SMF recording cannot be used and you cannot use the sequential file option. TSO statistics can only be viewed on-line.

STATMON The STATMON keyword specifies that the Status Monitor from Candle Corporation is present in the system. The Status Monitor performs its function by intercepting VTAM traffic using the same method as AUTOMON@-VTAM. If the Status Monitor is present and this keyword is not specified as 'Y', abends will occur in the Status Monitor any time it is used when AUTOMON@-VTAM is active

Code the keyword value as 'Y' (yes) or 'N' (no). Omitting the keyword is the same as coding 'N'.

SVC (MVS only) Code the number of the type 3 user SVC keyword that is to be allocated for use by AUTOMON@-VTAM. If you installed the user SVC module 'IGC00xy' during installation, ensure that the SVC number is the same as the one installed. If you did not install the user SVC, this keyword will cause the designated SVC to be dynamically allocated.

VAPPL - IDENTIFY APPLICATIONS FOR COMPRESSION

Following are all of the possible operands of the VAPPL statement.

APPLID	VAPPL
	<u>INCL</u> EXCL, APTYPE= <u>OTHER</u> CICS IMS TSO, ASIZE=nnnn, CPRIMG=Y <u>N</u> , DELAY=nn, EWIMG= <u>F</u> Y B N, EXTATTR= <u>CMPRS</u> STRIP LEAVE, FTRAN=Y <u>N</u> , INBND= <u>Y</u> N, KEY= <u>8</u> n, NDFSUP= <u>Y</u> N, NULLS=Y <u>N</u> , PSIZE=nnnn, RBGEN=Y <u>N</u> , RESET=Y <u>N</u> , SCRSIZE=CICS,

STATS=Y | N,
STATDDN=ddname,
STATDSN=dsname,
TRACDDN=ddname,
TRACDSN=dsname,
UEXIT=name,
VIO=Y | N,
WRAP=Y | N

APPLID The application name is coded as the label of the VAPPL statement. It must correspond to the 1-8 character APPLID defined to VTAM for this application. If less than eight characters are coded in the name it is considered to be a generic application ID. In this case, it will apply to all VTAM applications which start with the same characters as those coded.

For instance, if 'CICS' were coded as the APPLID, all applications whose first four characters start with 'CICS' would be identified by this VAPPL statement.

The generic feature of the APPLID is intended to make it easy to exclude several applications which start with the same characters. While it can also be used for inclusion, we don't recommend it, since this causes all statistics to be lumped together into the one APPLID. You cannot distinguish performance between the different applications.

The one exception to the generic inclusion recommendation above is TSO. You should always designate TSO using the first three characters of the ACB name, which in almost every case is 'TSO'. This causes all TSO activity to be recorded for one entry, even though each TSO user actually operates under a unique ACB. If you want to accumulate separate statistics by TSO user, you can code each unique TSO ACB name as a VAPPL statement.

INCL|EXCL This is a positional operand rather than a keyword operand which states whether the application identified by the APPLID name is to be included (INCL) in compression or excluded (EXCL) from compression. This operand must appear before all other keyword operands.

If this operand is omitted, INCL is the default. If EXCL is coded, none of the other operands apply.

Note that exclusion of an application will override inclusion of a logical unit on the VLU statement if the logical unit is connected to this application. In other words, for compression to occur for a given datastream, both the logical unit and the connected application must be included, either explicitly or by default. (For more information, see the discussion *INCLUSION AND EXCLUSION OF APPLICATIONS AND LUS*, later in this section.)

APTYPE This keyword is used to identify the type of application in use. Valid entries are CICS, IMS, TSO and OTHER. Currently, the only use of this keyword is to identify CICS applications in order to obtain the user ID for the LU display of the VTEX transaction. We recommend that you code the APTYPE keyword however, as it serves both to document the table entry and may be required for future releases of AUTOMON@-VTAM.

If you omit APTYPE, the default is OTHER.

ASIZE This is the alternate screen size of the terminals which will be connected to this application. The value specified for the ASIZE keyword on the VAPPL statement is the "last resort" entry that AUTOMON@-VTAM will use if it detects an alternate-write command and it can not determine the alternate screen size from any other information. The compression routines will select the screen size from:

- The VLU statement first, if present.
- The bind image provided by VTAM, if no specification is present on a VLU statement.
- The VAPPL statement, if no specification is present on the bind image.

If an alternate screen size cannot be obtained from any of the above, a default size of 3564 will be used.

ASIZE may be coded as any value from 1920 to 9999; however it should properly correspond to the alternate LU screen size. Typical screen sizes for 3270 devices are:

model 2 - 1920 (24 x 80)	model 3 - 2560 (32 x 80)	model 4 - 3440 (43 x 80)	model 5 - 3564 (27 x 132)
-----------------------------	-----------------------------	-----------------------------	------------------------------

CPRIMG This keyword is used to control compression of the in-memory image of the LU screen when the LU is inactive (between transmissions).

If CPRIMG is coded as 'Y', the in-memory image of the LU screen will be compressed during inactivity. This will require slightly more CPU time, but will save an average of 1K per terminal in static storage usage.

Image compression is performed by eliminating all repetitive duplicate characters in the image. As you know, there are a great number of screen displays which only contain a few characters - a single error message, for instance. Image compression frees the static storage in use for the expanded image and acquires an area only as large as the number of data characters on the screen (plus a few control bytes). The image is compressed each time an outbound datastream is sent and is expanded when inbound data is received. If the screen is cleared, the image is discarded altogether. This means that during the time that the operator is keying a response, the storage that is tied up is kept to an absolute minimum.

If CPRIMG is omitted, the default is 'N'.

DELAY This keyword is used to specify the count value for the logon-delay feature. Logon-delay will defer starting compression for an LU until the designated number of outbound transmissions has occurred.

Logon-delay is provided for the case when logon to an application produces an unusual display which causes problems with AUTOMON®-VTAM. Some applications use graphic logo displays, or other special displays which may not work well with compression.

You should try AUTOMON®-VTAM first with no logon delay. If you receive an incorrect display at logon time, or at the first entry following logon, then set the DELAY keyword to a value of 2 or 3. In some cases, a higher value may be required.

If you have DELAY coded and find that you are not getting compression statistics for a particular application, try resetting the DELAY value to zero, then log off and back on to see if statistics begin accumulating. When DELAY is coded, the application must send an ERASE command to the screen in order for compression to start after the delay value is exhausted. It is sometimes the case that certain applications may never (or very infrequently) send an ERASE command.

If you have both scenarios, that is, the application does not do an ERASE so that you can't use logon delay, but the logon display causes problems with compression on, you can probably use a VSKIP statement to bypass the logo display. Alternatively, you can notify Unicom Systems Inc. Systems technical support to debug the problem with the logo display.

If not coded, DELAY will be inactive. A value of 00 means that logon-delay is not active as well. In this case, compression will begin at logon.

EWIMG

The EWIMG keyword is used to control the erase/write imaging feature of AUTOMON®-VTAM. Erase/write imaging is the feature which changes ERASE commands to a WRITE with no erase, sending only data that changed to the terminal, and clearing fields that would normally have been erased via the ERASE command and not replaced in the datastream.

If EWIMG is coded as 'F' or 'Y', field-level compression will be active. When field boundaries change, as they would when a different screen display is output, the erase command will be left in the datastream. Thus, on screen changes, the new screen is sent with only repetitive character compression. This is the default if omitted.

If coded as 'B', byte-level compression will be active. Field boundaries will be ignored and any outbound data that is already present on the screen will be suppressed. The ERASE command is always changed to no ERASE unless the resultant compressed datastream is longer than the original datastream.

As a general rule, field-level compression will produce slightly higher compression percentages than byte-level, although this is not the case if frequent screen changing occurs. Byte-level compression will use significantly less CPU overhead.

If 'N' is coded, image compression will not be active. This option is provided as a means of controlling your operation should some application be found that does not work correctly with the erase/write imaging feature active. Running with EWIMG=N will significantly degrade the amount of compression that is performed but will require less storage overhead since no image buffer is required.

EXTATTR

This keyword controls the type of compression done for extended attributes.

If EXTATTR is coded as 'CMPRS', extended attributes already present on the screen will be removed from the outbound datastream. Although some extended attribute combinations can not be suppressed, most can. This is the default if the keyword is omitted. With this option, AUTOMON®-VTAM will attempt to replace extended attribute sequences with default field attributes where they are equivalent. For instance, the extended sequence

unprotected, red requires 6 bytes for the attribute definition. The same result at the terminal can be obtained by sending a normal field attribute of

unprotected, high-intensity

which only requires two bytes.

If coded as 'STRIP' extended attributes will be stripped and only the field attribute will be sent. STRIP should be used if you have applications that send extended attributes but have terminals that do not support them.

If coded as 'LEAVE', AUTOMON®-VTAM will leave the extended attribute as is.

[Note]: EXTATTR may also be specified on a VLU statement.

FTRAN Code 'Y' for the FTRAN keyword if you expect that file transfers can operate in this VTAM application. For more information on file transfers, see *HANDLING PC TRANSFERS* in the *OPERATION* section.

INBND The INBND keyword specifies whether inbound compression is to be used. Code a 'Y' if inbound mirroring is desired, otherwise code an 'N'.

Coding INBND=N may slightly reduce CPU overhead, but will also reduce overall compression results.

If INBND is omitted, 'Y' is the default.

KEY (MVS only)
T
his keyword specifies the storage protect key to be used for temporary buffer storage obtained by AUTOMON@-VTAM. If the application runs VTAM authorized path, this keyword is a necessity. Incorrect setting or omission can result in an SOC4 abend of the application.

For CICS, NETMASTER, NETVIEW and ROSCOE, code KEY=8.

For IMS, code KEY=7.

For TSO, omit KEY.

For all other applications, if you are not sure whether VTAM authorized path is in use, code KEY=8. If KEY is omitted and the application ABENDs with a SOC4, code KEY=8.

If KEY is omitted, '8' is the default for all applications except TSO.

NDFSUP This keyword designates that non-display fields are to be suppressed in outbound datastreams. If INBND=Y, fields with an attribute of non-display can in almost all cases be eliminated from the outbound datastream.

This feature can sometimes result in significant savings. It is a fairly common practice to send non-display data to the terminal with MDT on, so that when the operator presses ENTER, this data comes back in to the application and is used to determine the function to be performed. As long as inbound compression is being used, AUTOMON@-VTAM will return this data to the application anyway, so there is no need to send it to the terminal at all.

Code the keyword value as 'Y' (yes) or 'N' (no). The default is 'Y' if omitted.

NULLS This keyword designates that this application may have null characters on the terminal screen when data is input. It may be necessary to code NULLS=Y in order for AUTOMON@-VTAM to properly handle the input and subsequent display.

In general, this keyword is only applicable for text editors such as TSO, ROSCOE, ICCF, etc. These applications often leave nulls at the end of each text line in order to facilitate keyboard insertion of data.

It is not necessary to code NULLS=Y just because the application may invoke a text editor. This keyword is provided to handle the following situation:

- When null characters are on the text line, if the operator moves the cursor out past the data on the line using the cursor movement keys (rather than using the space bar), then keys some data, then presses ENTER, the normal result is for the data keyed to shift left and concatenate with the last character on the line, or the beginning of the line if there were no characters present.
- When compression is active and this scenario is performed, it is sometimes the case that the data that was keyed will either disappear from the screen altogether or may stay on the screen where it was keyed (in addition to shifting left as normal). The text record is properly updated, it simply does not redisplay correctly until the screen is refreshed.

Coding NULLS=Y will prevent these erroneous display conditions from happening.

Note, however, that you should not arbitrarily code NULLS=Y for all applications without knowing whether you need to or not. The null character control logic causes slightly more storage usage, as well as a minor reduction in compression results.

Again, NULLS=Y is not necessary unless your operators are in the habit of entering data out beyond the null characters and are expecting it to shift to the left. This is an unusual method of operating, but text editors lend themselves to the situation.

Code the keyword value as 'Y' (yes) or 'N' (no). The default is 'Y' if omitted.

PSIZE

This is the primary screen size of the terminals which will be connected to this application. The value specified for the PSIZE keyword on the VAPPL statement is the "last resort" entry that AUTOMON®-VTAM will use if it can not determine the screen size from any other information. The compression routines will select the screen size from:

- The VLU statement first, if present,
- The bind image provided by VTAM, if no specification is present on a VLU statement,
- The VAPPL statement, if no specification is present on the bind image.

If a primary screen size cannot be obtained from any of the above, a default size of 1920 will be used.

PSIZE may be coded as any value from 1920 to 9999; however it should properly correspond to the actual LU screen size. Typical screen sizes for 3270 devices are:

model 2 - 1920 (24 x 80)	model 3 - 2560 (32 x 80)	model 4 - 3440 (43 x 80)	model 5 - 3564 (27 x 132)
-----------------------------	-----------------------------	-----------------------------	------------------------------

RBGEN

This keyword controls the suppression of read-buffer commands in the application.

If coded as 'Y', when a read-buffer command is intercepted, AUTOMON®-VTAM will generate the read-buffer datastream from the internal image and pass it to the application. This eliminates the line time required to transmit a full terminal buffer.

Read-buffer commands are very time-consuming on a remote line since the entire terminal buffer is transmitted. Many session manager products, on-line help products and other products make use of the read-buffer command.

Note that when extended attributes are in use, some attribute combinations may negate the RBGEN option for that one request only. In this case, the read-buffer command will execute normally.

If coded as 'N', RBGEN will be inactive. 'N' is the default if omitted.

[Note]: If RBGEN=Y is coded, you must be sure that the minimum input terminal I/O area sizes of the application are large enough to contain an entire read-buffer datastream. For CICS, this means that the minimum TIOL or IOAREALEN value on each terminal definition must be at least 2560 for model-2 terminals. If the I/O area sizes are too small, a AUTOMON®-VTAM buffer-overflow message will appear at the terminal when a read-buffer command is attempted.

RESET If coded 'Y' this keyword specifies that application statistics are to be reset to zero any time the application is re-initiated. If RESET is omitted or coded 'N', application statistics are accumulated continually and will only be reset to zero when an on-line reset is performed from the VTEX transaction or from TSO, or when an IPL occurs.

SCRSIZE This controls whether or not the primary and alternate screen sizes are to be determined from the CICS TCTTE.

If coded as 'CICS', the default and alternate screen sizes in the TCTTE will override both the bind image (LOGMODE) and any occurrence of PSIZE and ASIZE in the customization table. SCRSIZE is a special keyword for use in CICS only. If this is not desired or the application is not CICS, omit the keyword.

STATS The STATS keyword specifies whether compression statistics are to be captured for this application or not. Code a "Y" for yes, or "N" for no.

If "Y" is coded, a statistic record is output each time a "CLOSE DESTINATION" command is issued by the application. This would normally be when the user logs off of an application or when the application is shut down, either normally or abnormally, but can also occur if a VARY OFFLINE command for the application is issued to VTAM.

If STATS is omitted, "N" is the default.

STATDDN (MVS only) If statistics records are to be written to a sequential disk file, the DDNAME of the disk file must be supplied via the STATDDN keyword. The absence of the STATDDN keyword indicates that SMF recording is to be used to capture the statistics records.

If coded, either the STATDSN keyword must be present or a DD statement with the specified DDNAME must be present in the startup JCL for the application system. If neither is present, statistics will not be recorded for this application.

STATDSN (MVS only) If statistics records are to be written to a sequential disk file, the DSNAME of the disk file can be supplied via the STATDSN keyword. It only applies if STATS=Y and a sequential file is to be used rather than SMF for the statistics file. The presence of STATDDN and STATDSN causes AUTOMON®-VTAM to do dynamic allocation for the statistics file.

If not coded, a DD statement with the specified DSNAME must be present in the startup JCL for the application system. If not present, statistics will not be recorded for this application.

TRACDDN (MVS only) If the VTAM trace feature of AUTOMON®-VTAM is to be used, the DDNAME of a sequential file for use in capturing trace records must be provided with the TRACDDN keyword.

The trace feature is activated by the presence of VTRACE statements in the customization table for specified logical units.

If coded, either the TRACDSN keyword must be present or a DD statement with the specified DSNAME must be present in the startup JCL for the application system. If neither is present, trace records will not be output.

For more information on the trace feature, see section 07 - USING THE VTAM TRACE FEATURE.

TRACDSN (MVS only) If the VTAM trace feature of AUTOMON@-VTAM is to be used, the DSNAME of a sequential file for use in capturing trace records can be provided with the TRACDSN keyword. The trace feature is activated by the presence of VTRACE statements in the customization table for specified logical units. It only applies if there are VTRACE statements present in the customization table. The presence of TRACDDN and TRACDSN causes AUTOMON@-VTAM to do dynamic allocation for the trace file.

If not coded, a DD statement with the specified DSNAME must be present in the startup JCL for the application system. If not present, trace records will not be output.

For more information on the trace feature, see section 07 - USING THE VTAM TRACE FEATURE.

UEXIT The UEXIT keyword specifies the name of a user exit that is to be invoked for all terminals that are logged on to this VTAM application.

If used, code the name of your outbound user exit program. For details concerning outbound user exits, see section 08 - CODING AN OUTBOUND USER EXIT.

VIO (DOS/VSE only) The VIO keyword designates whether the VIO buffer pool is to be used for terminal image storage or not.

If coded 'Y', all terminal image buffers will be moved to VIO during periods of inactivity for the LU. The GETVIS virtual storage required for the terminal image is freed when the image is moved to VIO.

If coded 'N', the VIO buffer pool is not used and all terminal image storage is obtained from the GETVIS area of the application partition.

If omitted, "N" is the default.

Note that if the VIO buffer pool fills up, subsequent requests for image storage will be automatically taken from the GETVIS area.

It is sometimes necessary to increase the available amount of VIO storage in the buffer pool in order to accommodate both AUTOMON@-VTAM and other applications which may use VIO. CICS, for instance, uses VIO to store its table of error messages used by the interactive interface. You control the amount of VIO storage in the system by the VIO keyword on the IPL supervisor parameters command of the IPL procedure. This is documented in *SYSTEM CONTROL STATEMENTS*.

It could be the case that a heavily loaded system with a very high transaction rate could experience degradation by using the VIO option. If you feel that this is the case, you could reduce the amount of VIO activity by decreasing the available amount of VIO storage in the buffer pool. VIO usage can be calculated - it requires 4K of VIO storage per active terminal. If you reduced the total

amount of VIO such that only half of your terminals could use VIO, for instance, you would still obtain significant GETVIS storage reduction benefits. Remember that AUTOMON®-VTAM will automatically switch to using GETVIS storage if the VIO pool fills up.

WRAP (MVS only)
T

The WRAP keyword applies both to the STATDDN and to the TRACDDN keywords. It defines the action to be taken when end-of-file is reached on either the statistics sequential file or on the trace sequential file.

If "Y" is coded, statistics and/or trace records will start over at the beginning of the file when end-of-file is reached. When this occurs, the previously-written records are lost.

If "N" is coded, statistics/trace recording is stopped when end-of-file is reached.

If omitted, "Y" is the default.

SET DEFAULT ACTION FOR APPLICATIONS

AUTOMON®-VTAM must have a default action to take when it encounters VTAM traffic from an application system that is not identified with a VAPPL statement. This is accomplished by coding a special VAPPL statement with \$DEFAULT as the label, or APPLID.

You may code any keyword which is valid for a normal VAPPL statement on the \$DEFAULT VAPPL statement. Whatever is coded will determine whether these applications participate in compression or not, establish a default screen size and direct statistical recording.

If you omit the \$DEFAULT VAPPL statement, VTEXPRES will internally generate one when the customization table is interpreted. The parameters that will be set are:

\$DEFAULT VAPPL EXCL

[Note]: The \$DEFAULT statement must appear as the last VAPPL statement if more than one are coded.

COMPRESSION FOR VTAM SESSION MANAGERS

It is generally recommended that you do not include VTAM session managers for compression. The reasons for this are as follows:

- 1). If compression is active for both the session manager and the application system you are incurring double overhead. The datastreams going from the application to the session manager are being compressed, then the same datastream from the session manager to the terminal is compressed again.
- 2). Most VTAM session managers define their virtual terminals as VTAM APPL statements. Thus, as far as AUTOMON@-VTAM is concerned, each virtual terminal is a separate application. One of the control blocks that AUTOMON@-VTAM allocates is called a VECTOR, which takes approximately 8K. There is one vector per application. Therefore, each virtual terminal, being considered as an application, would have a vector allocated, meaning that the storage overhead would be 8K per session.
- 3). If compression is operating for the VTAM session manager, you have little or no control over individual applications that may operate under the session manager. That is, if you have a unique application for which you do not want compression, you cannot exclude the application since the datastreams are flowing through the session manager.

We recommend, therefore, that you control the compression at the application level, with individual VAPPL statements per application. If you are running a session manager, the outgoing datastreams will be compressed as they come from the application to the session manager. The session manager will simply pass on the compressed datastream to the terminal. Incoming datastreams will be expanded as they pass from the session manager to the application. You obtain the same amount of savings and incur a minimum of overhead.

[Note]: The previous discussion does not apply to AUTOMON@-VTAM or CICS-JUGGLER from Unicom Systems Inc. Systems. These products are CICS-based session managers and run as transactions in CICS. IF CICS is included for compression, datastreams from these products will be compressed just like any other CICS transaction datastream.

??LX90??	Accept all terminals with an LU name that has 'LX90' in positions 3 thru 6 as a match for this VLU.
??LX90	This example will do the same as the previous example since the length of the VLU label controls the number of characters that are compared to the LU name.
??++???	Accept all terminals with numeric characters in positions 2 and 3 as a match for this VLU.
99+	Accept all terminals with an alpha character in position 1, '99' in position 2 & 3, a numeric character in position 4, and any characters in positions 5 thru 8 as a match for this VLU.
>L3	Accept all terminals where the last two non-blank characters of the LU name are 'L3'.
>L??	Accept all terminals where the character 2 positions back from the last non-blank character of the LU name is an 'L'. The next two positions (the last 2 non-blank characters of the name) can be anything.

Note that a wild-card VLU or a right-justified VLU acts just like a fully justified VLU. That is, if the LU name of the terminal matches the VLU label after applying wild-card compare logic, then the action specified by the VLU will be taken and the terminal will be either included or excluded.

[Note]: If any of your LU names have the characters >, ?, | or + in the name, you cannot code those LU names as a fully justified name. The presence of those characters in the VLU label will be considered to be wild-card coding.

INCL | EXCL | INCL=symbol | EXCL=symbol

This can be a positional operand or a keyword operand which states whether the logical unit identified by the LUname is to be included (INCL) in compression or excluded (EXCL) from compression.

If coded as 'INCL' with no '=' following, this LU will be included in compression for all applications.

If coded as 'EXCL' with no '=' following, this LU will be excluded from compression for all applications.

If coded as 'INCL=symbol', this LU will be included in compression for the application(s) specified by 'symbol'. Multiple applications may be coded; however, the applications must be enclosed in parenthesis and must be separated by a comma. for example:

LUname VLU INCL=(APPL1,APPL2,APPL3)

'EXCL=symbol' is coded just like the 'INCL=symbol' statement, but is used to exclude this LU from compression in the specified application(s). Furthermore, both 'EXCL=symbol' and 'INCL=symbol' statements may be coded on the same VLU statement.

When this type of selective inclusion or exclusion is used, and the LU logs onto an application that is not specified by an application in the 'symbol' list, then the action taken by AUTOMON@-VTAM will be dependant on the 'VSTART LU=INCL | EXCL' statement. In other words, if this LU is using selective inclusion or exclusion, and if the application is not specified by the selective statement(s), then EXPRESS will follow the action specified by the VSTART LU=INCL | EXCL' statement.

When using selective inclusion or exclusion, the symbol(s) coded must correspond to one of the labels of a previously specified VAPPL statement. There must be a VAPPL statement present for every symbolic name coded on an 'INCL=' or 'EXCL=' operand.

If this operand is omitted, INCL is the default. If EXCL is coded with no '=' following, none of the other operands apply.

For more information on inclusion and exclusion of applications on LU's, see *INCLUSION/EXCLUSION OF APPLICATIONS AND LU'S*, later in this section.

PSIZE This is the primary screen size of the logical unit. The value coded for the PSIZE keyword in the VLU statement is the "first choice" entry that AUTOMON®-VTAM will use when trying to determine the screen size. The compression routines will select the screen size from the VLU statement first, if present, from the bind image provided by VTAM if no specification is present on a VLU statement and finally from the VAPPL statement if no specification is present on the bind image.

ASIZE This is the alternate screen size of the logical unit. The value coded for the ASIZE keyword on the VLU statement is the "first choice" entry that AUTOMON®-VTAM will use if it detects an alternate-write command and it needs to determine the alternate screen size. The compression routines will select the screen size from the VLU statement first, if present, from the bind image provided by VTAM if no specification is present on a VLU statement and finally from the VAPPL statement if no specification is present on the bind image.

EXTATTR This keyword controls the type of compression done for extended attributes.

If EXTATTR is coded as 'CMPRS', extended attributes already present on the screen will be removed from the outbound datastream. Although some extended attribute combinations can not be suppressed, most can. This is the default if the keyword is omitted.

If coded as 'STRIP' extended attributes will be stripped and only the field attribute will be sent. STRIP should be used if you have applications that send extended attributes but have terminals that do not support them.

If coded as 'LEAVE', AUTOMON®-VTAM will leave the extended attribute as is.

[Note]: If not coded on the VLU statement, the action specified for EXTATTR on the VAPPL will be taken.

UEXIT The UEXIT keyword specifies the name of a user exit that is to be invoked for this terminal.

If used, code the name of your outbound user exit program. For details concerning outbound user exits, see section 08 - *CODING AN OUTBOUND USER EXIT*.

PRINTER COMPRESSION

In addition to LUs, some on-line printers have repeat-to-address capability as well. Amongst these are BSC and LUtype3.

AUTOMON®-VTAM will automatically identify any printers that can take advantage of the RA sequence, and compress repetitive characters from the printer datastream. This is the only type of compression that can be performed for printers.

[Note]: LUtype1 printers do not have this capability, so no compression can be performed for their datastreams. Ineligible printers do not need to be explicitly excluded from compression since AUTOMON®-VTAM will automatically recognize these printers and exclude their datastreams from compression.

VSKIP - IDENTIFY DATASTREAM TO BE EXCLUDED

From time to time it may be discovered that certain transactions of an application do not work correctly when compression is active for that application. If this occurs it is because the transaction is producing an unusual datastream which AUTOMON®-VTAM is not equipped to handle.

The VSKIP statement is provided to temporarily exclude specific datastreams while the technicians at Unicom Systems Inc. Systems determine the problem and correct the product.

Since AUTOMON®-VTAM operates at the VTAM level it cannot recognize transactions in the way that it could if it operated only in CICS or IMS. Nevertheless, most screen displays have some text on the screen which is unique to that screen. This identifying text can be used to define the screen display to be excluded from compression.

There are three methods of coding the VSKIP statement.

- 1). The easiest method is to use the ROWCOL keyword, along with the TEXT keyword to designate the position on the screen where the identifying text occurs.

With this method, display the screen where the problem occurs while AUTOMON®-VTAM is inactive, or on a terminal which is not participating in compression. Determine the text to be used to identify the screen, then calculate the screen row and column where the first byte of the text occurs by counting down from row 1 and left from position 1 of the row where the text is.

You can use text which includes imbedded spaces if desired. The text does not need to be a protected field, although it probably should be to ensure that it will always be the same. You do not need to be concerned about field attributes, either preceding or imbedded in the text data.

- 2). The second method is to code the OFS keyword, which is the offset, or actual position of the first byte of the identifying text in the output datastream.

In order to use this method, you must view the actual datastream produced by the program in order to code the VSKIP statement correctly. This can be accomplished in several ways:

- With compression off for the application or for the terminal, run a VTAM buffer trace (GTF trace) when the transaction is invoked.
- With compression on for the application or for the terminal, use the VTRACE option or the local trace option of AUTOMON®-VTAM (described in section 07 - *USING THE VTAM TRACE FEATURE*) to capture the datastream.
- Force the program to abnormally terminate with a dump at the point that it is ready to send the datastream, then examine the datastream in the dump. (Note that this will only work for programs that build native-mode 3270 datastreams, not programs that use BMS or some other mapping technique that is external to the program).
- Use CEDF, INTERTEST or any other on-line debugging tool to intercept the program at the point that it is ready to send the datastream. (Note that this will only work for programs that build native-

mode 3270 datastreams, not programs that use BMS or some other mapping technique that is external to the program).

Once the native-mode datastream has been trapped, locate a series of hexadecimal characters that will always appear in this datastream, compute the offset from the first byte of the datastream to the first character of this series (relative to zero), then code the VSKIP statement with OFS=xxx where xxx is the computed offset.

- 3). The third type of VSKIP statement is an inbound VSKIP. The first two methods work by recognizing an outbound datastream and turning off compression for that single output transmission. An inbound VSKIP recognizes data coming in from the terminal, then turns off compression for one or more subsequent outbound and inbound transmissions. Thus, you would code the VSKIP to recognize the input data which produces the output screen which is in error.

An inbound VSKIP statement is designated by coding TYPE=IN on the VSKIP statement. You can optionally code COUNT=nn, where nn is the number of subsequent outbound transmissions that are to occur before compression is turned back on. If COUNT is omitted, the default is a count of one.

The text that you code on an inbound VSKIP must be the first text data that appears in the incoming datastream. You do not need to be concerned with SBA sequences or other datastream orders. When processing an inbound VSKIP, datastream orders are ignored and the first text in the datastream is located, then compared to the value coded for TEXT=.

For most problem situations, using the ROWCOL keyword will work just fine. The exception is the situation where, because of something in the datastream that AUTOMON@-VTAM cannot handle, it is unable to properly build the terminal image, and therefore unable to locate the text by row/column address. We recommend that when you need to use VSKIP, you first try the ROWCOL method, or the inbound method since these are the easiest to use. If the failure still occurs, you will need to use the OFS keyword.

For more information on analyzing and handling problem situations, see section 09 - *PROBLEM ANALYSIS AND DIAGNOSIS*.

The VSKIP statement is coded as follows:

```
VSKIP      TEXT=xxxxxxxx
           [,ROWCOL=(x,y)|OFS=nnn,]
           [,TYPE=IN,]
           [,COUNT=nn]
```

Following are all of the possible operands of the VSKIP statement.

TEXT The TEXT keyword identifies a stream of characters that will always be present in this output datastream. Any number of characters may be coded. The maximum number of bytes that can be coded is 40. The text can be coded in hexadecimal or character form, as follows:

Hexadecimal format - TEXT=xxxxxx (xxxxxx = hexadecimal digits, two digits per byte)

Character format - TEXT='C'xxxxxx' (xxxxxx = character bytes)

If the OFS keyword is used to locate the text, the data bytes must be coded exactly as they appear in the datastream.

If the ROWCOL keyword is used, code any blank positions between text bytes as spaces, and do not be concerned with uppercase or lowercase letters. AUTOMON®-VTAM will translate to all uppercase and convert any attribute orders to space characters before making the compare.

If TYPE=IN is coded, the text must be the first data characters in the inbound datastream. Do not code any SBA sequences or other 3270 orders. Inbound VSKIP text would normally be coded in character format since it always refers to data that can be keyed on the terminal.

OFS The OFS keyword specifies the offset from the start of the datastream where the data coded in the TEXT keyword begins. It is coded as a decimal number, one to four digits, and it is relative to position zero. If OFS is omitted, zero is assumed. OFS does not apply and is ignored if TYPE=IN is coded.

ROWCOL The ROWCOL keyword specifies the row/column address on the screen where the data coded in the TEXT keyword begins. It is coded as two decimal numbers, one to three digits apiece and separated by a comma. It does not matter whether you enclose the row and column in parentheses or not.

Calculate the row number by positioning the cursor in the home position, which is row number 1. Move down and count one row at a time until the row is reached which contains the identifying text. Now, with the cursor in the first position of that row, move right and count 1 character at a time until the cursor is positioned on the first character of the identifying text. The position occupied by that character is the column number.

If this screen can be displayed on 132-column terminal as well as an 80-column terminal, it will be necessary to have two VSKIP statements present unless the text is on row 1. This is because the column number will be different for a field on any row past row 1 on a 132-column terminal.

ROWCOL does not apply and is ignored if TYPE=IN is coded.

TYPE=IN The TYPE keyword is only valid for an inbound VSKIP statement and IN is the only valid operand. It designates that the first data characters of inbound datastreams (coming from the terminal) are to be compared to the data coded for TEXT=. If a match is found, compression is turned off for the number of subsequent outbound transmissions specified by the COUNT keyword.

COUNT The COUNT keyword is only valid when TYPE=INPUT is coded. It designates the number of subsequent outbound transmissions for which compressions is to be stopped after an inbound datastream is found which has text characters matching the TEXT= value. If COUNT is omitted, the count is set to one. The maximum count value that can be coded is 255.

EXAMPLES OF VSKIP CODING

```
VSKIP TEXT='ACCOUNTSPAYABLE',ROWCOL=(1,35)
VSKIP ROWCOL=10,95,TEXT=C1C3C3E4D5E3E2
VSKIP TEXT=114040C1C3C3E4D5E3E2,OFS=0
VSKIP OFS=1910,TEXT='ID=ABCD'
VSKIP TYPE=INPUT,TEXT='CEMT'
VSKIP TYPE=INPUT,TEXT='CEMT',COUNT=4
```

Please contact Unicom Systems Inc. Systems technical support, (see Appendix B), about the problem that was encountered with this datastream. The technician will probably want you to run some of the diagnostics described in section 09 - *PROBLEM ANALYSIS AND DIAGNOSIS*.

VTRACE - IDENTIFY LOGICAL UNITS TO BE TRACED

The VTRACE statement indicates that tracing is to be performed for the named logical unit. Trace records are output to the sequential file identified by the TRACDDN keyword of the VAPPL statement, when the logical unit is connected to that application.

The VTRACE statement is coded as follows:

```
LUname VTRACE      [I/O]
```

- LUname The logical unit name is coded as the label of the VTRACE statement. It must correspond to a logical unit defined to VTAM. Note that the LU name for the VTRACE statement is not generic. You must code the full LU name.
- I/O Coding 'I/O' (or 'IO') in the operand field specifies that you want to trace only the sending and receiving datastreams, and not dump any of the control block areas that are used for diagnosing problems.

For more information on tracing control block areas, see section 09 - *PROBLEM ANALYSIS AND DIAGNOSIS*.

Activating and deactivating the trace feature can be accomplished by loading different copies of the customization table. See the *OPERATION* section under *OPERATION OF THE VTEXPRES PROGRAM* for discussion of this option.

For more information on the trace feature, see section 07 - *USING THE VTAM TRACE FEATURE*.

VFTRAN - IDENTIFY FILE TRANSFER BEGINNING AND END

When file transfers are done to upload or download data to a PC, compression must be turned off for the duration of the transfer. Otherwise, AUTOMON®-VTAM will attempt to apply compression algorithms to non-3270 data and the data will be corrupted. This usually results in abnormal termination of the file transfer job.

One method of accomplishing this is to manually turn off compression at the terminal (using VTEX or TSO) before starting the file transfer, then turn it back on after the transfer is complete.

Another method, is to code FTRAN=Y on the VAPPL statement. This will work as long as the file transfer program operates by means of a conversational dialog between the host and the PC, which most do. However, if the transfer program is not conversational (does not perform a VTAM READ specific), FTRAN will not work.

A method that is sure to work properly is to identify to AUTOMON®-VTAM the inbound datastream that signifies the start of a file transfer, and to identify the outbound datastream that signifies the end of the transfer. This can be accomplished with the VFTRAN statement.

The VFTRAN statement provides a method of identifying the beginning and end of a file transfer. When AUTOMON®-VTAM recognizes the beginning of the transfer, it automatically turns compression off at the terminal. When the end is recognized, it automatically turns it back on.

The VFTRAN statement is coded as follows:

```
VFTRAN      [START=(C'xxxx'|X'xxxx'[,offset]),  
             STOP=(C'xxxx'|X'xxxx'[,offset]),]
```

[LOGON=(C'xxxx'X'xxxx'[,offset])]

For a file transfer, you code the **START** and **STOP** keywords to identify the beginning and end of the file transfer. The **LOGON** keyword is mutually exclusive with **START** and **STOP** and may be used to identify a TSO log-on command, as described below.

You must code the text characters, in either hex or character form, that identify the inbound datastream that initiates a file transfer using the **START** keyword.

You must code the text characters, in either hex or character form, that identify the outbound datastream that terminates a file transfer using the **STOP** keyword.

You can code both the **START** and **STOP** keywords on one statement or each may be a single statement.

There can be any number of **START** and **STOP** statements coded. They do not have to be equal in number.

If the offset value is coded, it designates the exact position in the datastream (relative to zero) where the text bytes occur. If it is omitted, the first data characters in the datastream (after bypassing all 3270 orders) will be used to compare to the text data coded.

If offset is coded, you must separate the text data from the offset by a comma and enclose the text data and the offset in parentheses. If no offset is coded the parentheses are not necessary.

You must precede the text data with **C'** or **X'** to designate character or hexadecimal data. Hexadecimal data must have an equal number of digits coded, 2 digits equal one hex byte. Terminate the text string with a quote.

IDENTIFYING A TSO LOGON COMMAND

The **LOGON** keyword of the **VFTRAN** can be used to properly handle the following situation:

- 1). A user logs on to TSO and operates for a while, then exits TSO via the appropriate **END** commands and returns to a **READY** prompt.
- 2). A command such as **'LOGON'**, or some other **CLIST** name is used to log-on to TSO again, with the same or different user ID.

When this situation occurs with compression active, there will sometimes be erroneous displays during the log-on process. The user may have to enter the password twice, or interim messages from TSO may not appear on the screen. This situation occurs because **AUTOMON®-VTAM** cannot access certain storage during a TSO logon while the address space is being initialized.

To alleviate the problem, code a **VFTRAN** statement with **LOGON** as the only keyword, followed by the text of the log-on command or **CLIST** name. Code it exactly as you would code a **START** keyword.

With the **VFTRAN** statement present containing the **LOGON** keyword, **AUTOMON®-VTAM** will recognize the situation and temporarily turn itself off for this terminal until the log-on is completed. Then it will turn itself back on. You do not need to code anything to restart compression, it will be handled automatically.

EXAMPLES OF VFTRAN CODING

Assume that transaction code FTRN initiates a file transfer and the final message to the terminal at the end of the transfer is TRANSFER COMPLETE. The following VFTRAN coding would fully identify this example:

```
VFTRAN          START=C'FTRN',STOP=C'TRANSFER COMPLETE'
```

This example assumes that the characters FTRN will appear as the first data characters of an inbound datastream and the characters TRANSFER COMPLETE will appear as the first data characters of an outbound datastream. If, in fact, the outbound message has other data preceding it, it would be necessary to code an offset to designate the position in the datastream of the token data, as follows:

```
VFTRAN          VFTRAN  START=C'FTRN'  
VFTRAN          STOP=(C'TRANSFER COMPLETE',10)
```

In the preceding example, the text TRANSFER COMPLETE must appear in the 11th byte of the outbound datastream (offset is coded relative to position zero).

When character data is coded, it does not matter whether it appears as upper or lower case. Both the text from the datastream and the text in the VFTRAN statement are converted to upper case before the comparison is made.

Two examples of the coding for a LOGON entry are as follows:

```
VFTRAN          LOGON=C'LOGON'  
VFTRAN          LOGON=C'HELLO'
```

In these cases, if the user enters 'LOGON' or 'HELLO' at a TSO READY prompt, compression will be automatically stopped until the TSO log-on is completed, then automatically restarted.

THE IBM STANDARD FILE TRANSFER

The distributed skeleton customization table, VTEXSKEL, contains the proper VFTRAN coding to recognize the standard IBM file transfer package used both with TSO and the interactive interface of DOS/VSE. This system uses IND\$FIL as a file name and IND\$ as a transaction code. If you are using the IBM standard file transfer, leave the pre-coded VFTRAN statements in your customization table and they will work.

[Note]: If you are not sure what kind of file transfer package you use, or if you do not know what data designates the beginning and end of the process, it may be necessary to run a VTAM buffer trace of a file transfer and examine the starting and ending datastreams. If you need help with this, call Unicom Systems Inc. Systems technical support for assistance.

INCLUSION AND EXCLUSION OF APPLICATIONS AND LU'S

Deciding whether applications or LU's should be included in or excluded from compression can be made simple by keeping the following rule in mind.

For compression to be active on any level, compression must also be active on all hierarchical levels above that level.

In other words, if an LU is to participate in compression, the application that the LU is logged onto must also be participating in compression, as well as the system. Therefore, if an application is excluded, this will override the inclusion of an LU.

The 'LU=' keyword on the VSTART statement establishes the default action to be taken when a terminal logs onto an application that is included in compression. If there is no VLU statement present which matches the LU name of the terminal, the action specified on the VSTART will be taken.

If a VLU statement is present for the terminal, the action specified on the VLU overrides the action specified on the VSTART. The only exception to this is when selective inclusion or exclusion by application is specified on the VLU. In this case, if the application being logged onto is not in the list of selected applications of the VLU statement, the action specified on the VSTART statement will be taken.

Under most circumstances, compression should be active on every level. However, should the condition arise in your environment, that AUTOMON®-VTAM can not properly handle an application or an application/LU combination, then maximum flexibility of inclusion and/or exclusion has been provided with the customization options of AUTOMON®-VTAM.

INDEX

- application name
 - in VAPPL statement...30
- APTYPE keyword
 - description of.....30
- ASIZE keyword
 - description of.....30, 42
- attributes
 - extended32, 42
- authorized path
 - limitations with.....29
- CICS
 - minimum TIOA size.. 12, 18, 35
 - required table changes12, 18
 - screen size control35
 - using MRO and ISC ..11, 17
- compression
 - exclusion of terminals 42
 - for printers.....42
 - inbound.....4, 33
 - outbound.....3, 4
 - packages available.....2
 - techniques used3
- COUNT keyword
 - description of.....46
- CPRIMG keyword
 - description of.....31
- cross-domain
 - in VTAM.....12, 17
- CUSTOMIZATION*.....9
- customization options
 - activate statistics.....35
 - alternate screen size...31, 42
 - application exclusion..30
 - assigning password.....28
 - CICS screen size control35
 - control erase/write
 - imaging.....32
 - dataset name of LPA ..27
 - default action for
 - applications38
 - default action for LUs.27
 - excluding transaction
 - datastreams43
 - excluding/including LUs40
- extended attribute
 - compression.....32, 42
 - file wrap option38
 - identify application type30
 - identifying applications30
 - image compression.....31
 - inbound compression..33
 - log-on delay31
 - maximum storage control.....27
 - non-display field suppression33
 - null character control..33
 - outbound user exit
 - specification of...37, 42
 - primary screen size34, 42
 - read-buffer generation 34
 - RU consolidation.....35
 - SMF recording.....28
 - statistics file DDNAME35
 - statistics file DSNNAME35
 - statistics reset.....35
 - storage protect key.....33
 - trace file DDNAME ...35
 - trace file DSNNAME....37
 - VIO buffer pool usage 37
- customization statements
 - VAPPL29
 - VFTRAN.....47
 - VLU.....40
 - VSKIP43
 - VSTART27
 - VTRACE.....46
- customization table
 - contents of skeleton 8, 15
 - defining.....25
 - format of.....25
 - generation of defaults .38
 - name of.....8, 15, 25
 - order of26
 - purpose of.....25
 - statements of.....26
- datastreams
 - built with mapping.....2
 - exclusion of 44
 - inbound..... 4
 - inefficiency factors 2
 - outbound..... 3, 4
- default action
 - for unknown applications..... 38
- DELAY keyword
 - description of..... 31
- erase command
 - removal of..... 4
 - use of 2, 3
- EWIMG keyword
 - description of 32
- EXCL keyword
 - description of30, 41
- exclusion
 - of applications 30
 - of logical units27, 40
 - of transaction datastreams 44
- EXTATTR keyword
 - description of32, 42
- extended attributes.....32, 42
- file transfers
 - coding examples 48
 - IBM standard 49
 - method of identifying 47
- FTRAN keyword
 - description of 32
- generic application ID
 - specification of 30
- IEAPAK
 - use of 7
- image
 - in inbound compression4
 - in outbound compression 4
- image saving
 - controlling 32
- image-saving
 - description of 3
- IMS
 - considerations for 13
 - minimum TIOA size.. 12
 - performance considerations 13
- INBND keyword
 - description of 33
- inbound mirroring

description of.....4	OPSYS21	password
specification of33	PRODUCT21	for product
INCL keyword	VTWOFILE21	description of..... 22
description of.....30, 41	WDOFILE.....21	fields of..... 23
installation	of STSINST program .19	initial value of..... 5
MVS6	OFS46	installation of
of other products.....6	PASSWORD28	DOS/VSE.... 22
tape contents.....5	PSIZE34, 42	MVS 23
IPL	RBGEN34	PASSWORD keyword
creation of SVA.....18	RESET.....35	description of 28
JCL	ROWCOL.....46	functions required for 28
link-edit object members	RUCONS.....35	PC transfers(see file
.....19, 22	SCRSIZE.....35	transfers)
load install program	SMFID.....28	performance
DOS/VSE.....14	SMFSVC29	considerations for .13, 19
MVS6	SMFUSER.....29	printer compression
load source members .19,	START47	types supported..... 42
22	STATDDN35	PSIZE keyword
product password install	STATDSN.....35	description of34, 42
.....22	STATMON.....29	RA sequence
running the install	STATS.....35	use of 3
program	STOP47	RBGEN keyword
DOS/VSE.....14	SVC29	description of 34
MVS7	TEXT.....44	reading requirements
KEY keyword	TRACDDN.....35	for installation..... 1
description of33	TRACDSN37	for operation 1
keywords	TYPE=INPUT46	for problems 1
APPLID.....30	UEXIT37, 42	repetitive characters
APTYPE.....30	VIO.....37	elimination of 3, 4
ASIZE31, 42	WRAP37	request units
COUNT46	logical unit name	in VTAM.....13, 19
CPRIMG31	in VLU statement40	recommendation for.. 13,
DELAY31	in VTRACE statement47	19
EWIMG.....32	LOGON keyword	RESET keyword
EXCL30, 41	description of48	description of 35
EXTATTR.....32, 42	LPA7	response time
FTRAN.....32	listing directory of7	factors of..... 2
INBND33	LPADSN keyword	problem of 2
INCL30, 41	description of27	right-justified names
KEY.....33	LU keyword	in VLU statements.... 40
LOGON.....48	description of27	ROWCOL keyword
LPADSN27	mapping	description of 46
LU27	use of2	RU size(see request units)
LUname.....40, 47	MAXBUF keyword	RUCONS keyword
MAXBUF.....27	description of27	description of 35
NDFSUP33	MDT . (see modified data tag)	SBA sequence
NULLS33	modified-data-tag	use of 3
of installer program	removal of4	screen size
CICS.....20	use of2, 4	alternate30, 42
DIRECTORY20	NDFSUP keyword	primary34, 42
FSRVFILE20	description of33	selection of30, 34, 42
HELPSCRN20	NULLS keyword	setting default 38
HELPTXT20	description of33	SCRSIZE keyword
JUGLFILE20	OFS keyword	description of 35
LINES20	description of46	session managers
MODE.....20	PAK8	considerations for 39

Shared Virtual Area..... (see SVA)	description of.....5	keywords of..... 29
SMF recording	installation of.....6, 14	relation to VLU..... 42
specification of.....35	link-edit of	setting defaults..... 38
specifying.....28	DOS/VSE.....14	VFTRAN statement
with authorized path...29	MVS.....6	description of..... 47
SMFID keyword	suffix code	examples of..... 48
description of.....28	of customization table .9, 15	IBM standard..... 49
SMFSVC keyword	SVA	VIO keyword
description of.....29	adding VTEXMAIN...18	description of..... 37
SMFUSER keyword	creation of at IPL.....18	VLU statement.....40, 42
description of.....28	SVC keyword	description of..... 40
START keyword	description of.....29	keywords of..... 40
description of.....47	SVC number	relation to VAPPL 30, 34
STATDDN keyword	selection of.....29	wild-card coding of.... 40
description of.....35	SVC routine	VSKIP statement
STATDSN keyword	for SMF records.....29	description of..... 43
description of.....35	terminal input area	format of..... 44
statistics	minimum size required	keywords of..... 44
defining.....3512, 18	methods of..... 43
setting default.....38	TEXT keyword	VSTART statement
SMF specification.....28	description of.....44	keywords of; .i.VSTART statement
STATMON keyword	TRACDDN keyword	description of..... 27
description of.....29	description of.....35	VTAM
STATS keyword	relation to VTRACE...46	application IDs..... 30
description of.....35	TRACDSN keyword	session managers..... (see session managers)
STOP keyword	description of.....37	using cross-domain... 12, 17
description of.....47	trace	VTEXPRES
storage usage	activation of....35, 37, 47	to activate compression
controlling.....27	specification of35, 37, 47 18
STS0100	transactions	VTEXSKEL
description of.....5	exclusion from	contents of.....9, 15
installation of.....6, 14	compression.....44	explanation of.....11, 17
link-edit of	TSO	modification of..... 25
DOS/VSE.....14	coding rules for VAPPL	VTRACE statement
STSCORE30	description of..... 46
description of.....5	subsequent log-ons to .48	relation to TRACDDN
installation of.....6, 14	use of authorized path.2935, 37
STSinST	TYPE=INPUT keyword	wild-cards
description of.....5	description of.....46	in VLU statements..... 40
installation of.....6, 14	UEXIT keyword	WRAP keyword
keywords of.....19	description of.....37, 42	description of..... 37
link-edit of	user exit	
DOS/VSE.....14	specifying.....37, 42	
MVS.....6	VAPPL statement	
STSPASS	description of.....29	

