

# **AUTOMON®-VTAM**

**Technical Reference Guide**

**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

<b>PREFACE</b> .....	<b>1</b>
<b>INTRODUCTION</b> .....	<b>3</b>
MINIMUM READING REQUIREMENTS.....	3
COMPRESSION PHILOSOPHY AND TECHNIQUE.....	4
<b>INSTALLATION</b> .....	<b>7</b>
GENERAL INFORMATION.....	7
THE INSTALLATION TAPE.....	7
MVS INSTALLATION.....	8
DOS/VSE INSTALLATION - BASIC PRODUCT INSTALL.....	15
KEYWORDS OF THE UNICOM SYSTEMS INC.INSTALLATION PROGRAM.....	21
INSTALLING THE PRODUCT CONTROL PASSWORD.....	24
<b>CUSTOMIZATION</b> .....	<b>27</b>
CODING THE CUSTOMIZATION TABLE.....	27
INCLUSION AND EXCLUSION OF APPLICATIONS AND LU'S.....	51
<b>OPERATION - MVS SYSTEMS</b> .....	<b>52</b>
OPERATION OF VTEXPRES.....	53
LOADING A NEW CUSTOMIZATION TABLE.....	83
HANDLING PC TRANSFERS.....	85
RELOADING THE VTEXMAIN MODULE IN THE LPA.....	86
<b>OPERATION - DOS/VSE SYSTEMS</b> .....	<b>87</b>
OPERATION OF VTEXPRES.....	87
LOADING A NEW CUSTOMIZATION TABLE.....	101
HANDLING PC TRANSFERS.....	101
RELOADING THE VTEXMAIN MODULE IN THE SVA.....	102
<b>STATISTICS CAPTURE AND REPORTING</b> .....	<b>103</b>
MVS USERS.....	103
DOS/VSE USERS.....	105
THE STATISTICS REPORT.....	106
<b>USING THE VTAM TRACE FEATURE</b> .....	<b>108</b>
GLOBAL AND LOCAL TRACING.....	108
LOCAL TRACING - MVS AND DOS/VSE.....	113
<b>CODING AN OUTBOUND USER EXIT PROGRAM</b> .....	<b>121</b>
PROGRAMMING CONVENTIONS.....	121
SAMPLE USER EXIT PROGRAM.....	123

<b>PROBLEM ANALYSIS AND DIAGNOSIS.....</b>	<b>129</b>
NO COMPRESSION STATISTICS, PROCEDURAL CHECKLIST .....	130
DIAGNOSTIC DOCUMENTATION ACCORDING TO PROBLEM TYPE .....	132
BYPASSING THE PROBLEM.....	135
RUNNING THE SLIP TRACE .....	135
SUMMARY .....	137
<b>USING THE MEMORY DISPLAY/ALTER UTILITY .....</b>	<b>138</b>
SPECIAL COMMANDS UNIQUE TO AUTOMON@-VTAM .....	138
ALTERING MEMORY IN THE LPA OR SVA.....	139
<b>MESSAGES .....</b>	<b>140</b>
<b>APPENDIX A - STORAGE UTILIZATION.....</b>	<b>157</b>
STORAGE UTILIZATION IN MVS/XA .....	157
STORAGE UTILIZATION IN DOS/VSE .....	158
<b>APPENDIX B - INSTALLATION DOCUMENTATION .....</b>	<b>159</b>
<b>APPENDIX C - PROBLEM REPORTING .....</b>	<b>171</b>
<b>APPENDIX D - THE PRODUCT LINE .....</b>	<b>172</b>

## 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 3

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 9

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.

### OPERATION - MVS SYSTEMS (SECTION 04) Page 65

The methods used to activate and deactivate the MVS version of AUTOMON®-VTAM, either from a batch program or from a TSO command screen are described. In addition, the on-line statistics displays that may be obtained either through TSO or CICS are illustrated.

### OPERATION - DOS/VSE SYSTEMS (SECTION 05) Page 105

This section describes the methods used to activate and deactivate the DOS/VSE version of AUTOMON®-VTAM. Also illustrated are the on-line statistics displays that may be obtained through CICS.

### STATISTICS CAPTURE AND REPORTING (SECTION 06) Page 125

This section describes the procedures for printing the statistics report, and explains the various statistics of the report.

### USING THE VTAM TRACE FEATURE (SECTION 07) Page 131

This section describes the trace feature of AUTOMON®-VTAM, which may be used to capture buffer images at various points in your VTAM network. Procedures for activating and deactivating trace, as well as printing the trace report are discussed.

### CODING AN OUTBOUND USER EXIT PROGRAM (SECTION 08) Page 147

You may code a user exit program to monitor and optionally change all outbound datastream (those destined for a terminal) that flows through AUTOMON®-VTAM. This section describes the requirements and coding conventions necessary for the user exit.

PROBLEM ANALYSIS AND DIAGNOSIS (SECTION 09) Page 156

This section deals with problem situations, should they occur. The various types of diagnostic tools available are discussed, along with recommendations for dealing with the problem.

USING THE MEMORY DISPLAY/ALTER UTILITY (SECTION 10) Page 168

A core utility, along with its manual, accompanies this product. This section supplements the core manual by describing the additional commands that are exclusive to AUTOMON®-VTAM.

MESSAGES (SECTION 11) Page 170

This section lists all of the possible messages that can be produced by any of the programs in AUTOMON®-VTAM. Each message is shown in its entirety, along with a description of the condition involved and a suggested action to be taken.

APPENDIXES

The following appendixes are present:

Appendix A	Page 189
Describes the storage utilization of AUTOMON®-VTAM.	
Appendix B	Page 191
Shows the installation documentation that is printed in Step 2 of the installation procedure.	
Appendix C	Page 204
Describes the procedures to use when a problem is encountered with AUTOMON®-VTAM.	
Appendix D	Page 205
Describes the other products that are present on the Master Installation Tape.	

# 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

<i>All of OPERATION - MVS SYSTEMS,</i>	<i>section</i>	<i>04</i>
		<i>65</i>
<i>or OPERATION - DOS/VSE SYSTEMS,</i>	<i>section</i>	<i>05</i>
		<i>105</i>

## READING REQUIRED FOR OPERATION AND MAINTENANCE

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

## READING REQUIRED FOR PROBLEM HANDLING

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

## 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.COMDLIB,           [Note 1]
//           //    DISP=SHR
//MPRDIN     DD    DSN=MASTER.TAPE,UNIT=TAPE,
//           //    LABEL=(2,NL),VOL=SER=INPUT,
//           //    DCB=BLKSIZE=32000,
//           //    DISP=(SHR,PASS)
//SYSPCH     DD    DSN=????.(STSJCL),         [Note 2]
//           //    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.	
*			
LUsername01	VLU	EXCL	Fully qualified LUsername
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.
LUsername02	VLU	EXCL=(TSO,OTHER)	For this LU, exclude from TSO and
LUsername02	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
*
$DEFAULT VAPPL EXCL          Exclude all other applications
*
*****
Individual or generic LUs to be included or excluded. If only these LUs are
to be compressed, change the EXCL on the VLU to 'INCL' and change VSTART to
LU=EXCL.
*
LUname01 VLU EXCL          Fully qualified LUname
L???XX VLU EXCL          Exclude LUs with 'L' in pos. 1
*                          'XX' in 5-6 & anything in
*                          2-4 & 5-8
????++++ VLU EXCL          Exclude all LUs with numerics
*                          in pos. 5-8
?|X VLU EXCL          Exclude all LUs with alpha in
*                          2-3, 'X' in pos. 4 and
*                          anything in pos. 1, 5-8
LU????? VLU EXCL=CICS1    For all LUs beginning with 'LU',
*                          exclude when logged-on to CICS1,
*                          otherwise take action as coded
*                          on VSTART LU= keyword.
LUname02 VLU EXCL=(CICS1,CICS2) For this LU, exclude from
LUname02 VLU INCL=(CICS3,OTHER) compression in CICS1 and
*                          CICS2, Include in compression
*                          in CICS3 and OTHER.
*
*****
Start/Stop token table for standard IBM file transfer
*
VFTRAN START=(C'IND$',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
*
LUsername1   VTRACE                   Diagnostic trace for LUsername1
LUsername2   VTRACE I/O               I/O trace for LUsername2

```

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 LUsername1, LUsername2 and LUsername3 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:
- |                  |   |
|------------------|---|
| <b>INSTALL</b>   | Create JCL which <u>will</u> delete and redefine the VSAM file(s).  |
| <b>REINSTALL</b> | 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

**WNOFILE=xxxxx**        The WNOFILE 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).

**VTWOFIle=xxxxxx**    The VTWOFIle 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).

## 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'xxxxxxxxxxx'   Password
      .
      .
      .
      DC      CL6'000000'           CPU ID
      DC      CL4'xxxx'           Product ID
      DC      CL6'mmddy'         Expiration Date
      DC      CL16'xxxxxxxxxxx'   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 LU'S*, 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.



- 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

he 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. 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	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 and 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=xxxxxxx
           [,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]
```

- LUnitname      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.

## OPERATION - MVS SYSTEMS

Once you have completed the installation, defined a customization table and performed a re-IPL, you are ready to activate AUTOMON®-VTAM and begin operation.

The primary operational program is VTEXPRES. This program is used to activate and deactivate compression, as well as perform other functions which may be needed from time to time. The functions handled by this program are:

- 1). Display statistics for a TSO user.
- 2). Display statistics for each included application.
- 3). Display statistics by terminal within each application.
- 4). Display the table of VLU statements.
- 5). Activate compression
- 6). Deactivate compression
- 7). Deactivate compression and remove hooks (BACKOUT)
- 8). Turn compression on or off for a specific application
- 9). Turn compression on or off for a specific LU
- 10). Alter the customization options for an application.
- 11). Reset compression statistics to zero
- 12). Activate local tracing for a terminal.
- 13). Display local trace records.
- 14). Purge local trace records.
- 15). Load a new copy of VTEXMAIN in the LPA.

A second on-line operational program is the VTEX transaction which runs in CICS. This program performs similar functions to the TSO command processor in CICS. VTEX can be used to:

- 1). Display statistics for each included application.
- 2). Display statistics by terminal within each application.
- 3). Stop compression for the entire system.
- 4). Restart compression for the entire system.
- 5). Turn compression on or off for a specific application
- 6). Turn compression on or off for a specific LU
- 7). Alter the customization options for an application.
- 8). Reset compression statistics to zero
- 9). Activate local tracing for a terminal.
- 10). Display local trace records.
- 11). Purge local trace records.

In addition to the VTEXPRES program and the VTEX transaction, you will want to print the compression statistics that are gathered throughout the day. This function is described in section 06 - *STATISTICS CAPTURE AND REPORTING*.

If you use the VTAM trace feature of the product, you will need to print the trace entries that are captured. The procedure for this is described in section 07 - *USING THE VTAM TRACE FEATURE*.

## OPERATION OF VTEXPRES

For MVS systems, the VTEXPRES program can be run either as a batch program or as a TSO command.

For day-to-day operation, you should set up a batch job or procedure to run VTEXPRES and activate compression each time VTAM is cycled. VTAM must be up before compression can be activated, however the application systems should not be up. This means that you only need to activate compression after bringing up VTAM or after deactivating it for some reason.

### RUNNING VTEXPRES AS A BATCH PROGRAM

VTEXPRES can be initiated as a stand-alone job or as a started task. To initiate it as a started task you must create a PROC as in the example below. The following JCL can be used to run VTEXPRES as a stand-alone job in a batch region:

```
//VTEXPRES  JOB  (NNNN,NNNN,N),CLASS=A
//STEP1    EXEC  PGM=VTEXPRES,
//          PARM='option parameters'
//STEPLIB  DD   DSN=SYS1.CMDLIB,          [Note 1]
//          DISP=SHR
//SYSUDUMP DD   SYSOUT=A
```

The following PROC could be defined to allow VTEXPRES to be initiated as a started task:

```
//VTEXPRES  PROC  OPT=DSN,MEM=00
//VTEXPRES  EXEC  PGM=VTEXPRES,
// PARM='&OPT(SYS1.PARMLIB(VEXPRS&MEM))'  [Note 2]
//STEPLIB  DD   DSN=SYS1.CMDLIB,          [Note 1]
//          DISP=SHR
```

You could then initiate VTEXPRES with a START command such as ...

```
S VTEXPRES                To execute a DSN command.
S VTEXPRES,MEM=01         To load a different customization
                           table.
S VTEXPRES,OPTION=DEACT   To execute a DEACTIVATE
                           command.
```

**[Notes]:**

1. This is the load library where VTEXPRES was link-edited during installation. Programs STS0100 and STSPASS must either reside in this library or in a library which is concatenated with this one.
2. Specify the dataset where your customization table statements reside.

## VTEXPRES PARM STATEMENTS

The PARM statement of the EXEC command is used to specify the type of function that VTEXPRES is to perform. Possible PARM statements are:

**PARM='DSN(dataset(member))'**

Activate compression specifying the dataset and member name of the customization table to be loaded. This is the preferred method of activating compression.

**PARM='DEACT[ivate]'**

Deactivate compression by immediately quiescing. Deactivate allows immediate restart with an ACTIVATE or DSN command.

**PARM='BACKOUT'**

Deactivate compression permanently by removing the intercept points from VTAM.

**PARM='NEWCOPY[,DSN(dataset(member))]**

Load a new copy of the VTEXMAIN module into the LPA. The DSN keyword is optional.

Following is a description of each of the PARM statement commands:

**DSN** The DSN command performs the same function as the ACTIVATE command, but in addition, the dataset and member of the customization table is dynamically allocated and loaded. The sequence of events is:

- 1). The dataset and member specified is dynamically allocated.
- 2). The customization table is loaded, interpreted and the option table is built in common storage.
- 3). If not already present, the compression 'hook' is placed in VTAM to allow AUTOMON®-VTAM to intercept outbound and inbound datastreams. (This occurs on first activate only).
- 4). The quiesce flag in the system option control block is turned off.
- 5). A message is output to the system log that compression has been activated.
- 6). VTEXPRES terminates normally.

### DEACT or DEACTIVATE

The DEACTIVATE command, which may be abbreviated as DEACT, is used to stop or quiesce compression. After a deactivate, compression can be immediately restarted without recycling the applications or logging off and back on.

Upon receipt of the DEACTIVATE command, the following functions are performed:

- 1). The quiesce flag in the system option table is set on, causing all traffic through AUTOMON®-VTAM to be routed to VTAM with no further action.
- 2). A message is output to the system log that compression has been deactivated.
- 3). VTEXPRES terminates normally.

## BACKOUT

The BACKOUT command is used to terminate compression and remove the VTAM hooks.

Upon receipt of the BACKOUT command, the following functions are performed:

- 1). The quiesce flag in the system option table is set on, causing all traffic through AUTOMON®-VTAM to be routed to VTAM with no further action.
- 2). The compression 'hook' which allows AUTOMON®-VTAM to intercept outbound and inbound datastreams is removed from VTAM.
- 3). A message is output to the system log that compression has been deactivated.
- 4). VTEXPRES terminates normally.

## NEWCOPY

The NEWCOPY command loads the VTEXMAIN module into a temporary address in the CSA, then alters the LPA directly to point to it. This allows a new version of the program to be used, which will stay in effect in the CSA until an IPL is performed. At that time, the new module will be loaded into the LPA.

The sequence of events during a NEWCOPY command is as follows:

- 1). If the DSN keyword is coded, the option table specified by the dataset and member name is used. Otherwise, the option table currently in effect is used.
- 2). The VTEXMAIN module is loaded from the dataset specified by the LPADSN keyword in the user option table.
- 3). The SVC specified on the VSTART is dynamically allocated (if not already allocated).
- 4). All pointers and directory addresses are altered to point to the new module in the CSA.

**[Note]:** The old copy of VTEXMAIN is not (and cannot) be released. This is because there can be outstanding VTAM requests destined for the old module. However, once a NEWCOPY is performed, any request flowing to the old module is automatically rerouted to the new module. Since the old module is not released, repeated NEWCOPY requests will tie up more and more CSA storage (approximately 72K per NEWCOPY), which will not be released until an IPL is performed.

## CONSIDERATIONS FOR ACTIVATION OF COMPRESSION.

### **If this is the first activation or the first activation after a previous BACKOUT command:**

- When compression is activated, it will begin operating for each included application as users log on.
- If an application is already active before compression is activated, those users already logged on will not be compressed until they log off and back on.
- The exception to the previous point is TSO. If TSO is included, all included users will begin compressing immediately without logging off and on.
- The best way to ensure that all terminals and applications will participate in compression is to activate AUTOMON®-VTAM prior to initializing the on-line applications.
- A second activation can be done without a deactivation in order to load a different copy of the customization table. In this case, all users who are participating in compression will continue to do so.

### **CONSIDERATIONS FOR DEACTIVATION OF COMPRESSION**

- When compression is deactivated with a DEACT command, it immediately stops operating for all connected logical units. It also stops all terminals, which connect after the DEACT command, from using compression.
- If an application is recycled while compression is deactivated, compression will not restart when the application initializes. If a subsequent ACTIVATE or DSN command is issued, compression will restart.
- Traffic still flows through AUTOMON@-VTAM after a DEACT command. The system is considered to be in the "quiesced" state and no action is taken on any inbound or outbound datastream. It is simply passed on to VTAM for normal processing.
- Users who log on while compression is quiesced will begin compressing when compression is activated.

### **CONSIDERATIONS FOR BACKING OUT COMPRESSION**

- When compression is backed out, it immediately stops operating and all traffic stops flowing through AUTOMON@-VTAM.
- The BACKOUT command need only be used if you want to stop all activity in AUTOMON@-VTAM and remove all hooks. Once this is performed, a subsequent activation will require that users log off and back on to begin compressing.

## RUNNING VTEXPRES AS A TSO COMMAND

The VTEXPRES program may be executed at any time through TSO.

When executed as a TSO command, there are more functions that can be performed than when running as a batch program.

The functions which may be performed are:

- 1). Display statistics for a TSO user.
- 2). Display statistics for each included application.
- 3). Display statistics by terminal within each application.
- 4). Display the table of VLU statements.
- 5). Activate compression.
- 6). Deactivate compression.
- 7). Deactivate compression and remove hooks (BACKOUT).
- 8). Turn compression on or off for a specific application.
- 9). Turn compression on or off for a specific LU.
- 10). Alter the customization options for an application.
- 11). Reset compression statistics to zero.
- 12). Activate local tracing for a terminal.
- 13). Display local trace records.
- 14). Purge local trace records.

To execute the program from TSO, go to the TSO COMMAND SCREEN, and enter VTEXPRES as the command name. Or you can enter 'TSO VTEXPRES' from any TSO command area. Upon executing VTEXPRES, a screen will appear which automatically displays compression statistics for the current terminal, then provides a menu of PF key options for the various functions which can be performed:

The screen will appear as follows:

VTAM-EXPRESS RELEASE 2.0 PTF LEVEL 91xxxx		
USER: RPJ	NODE: V2002	COMPRESSION SYSTEM: ACTIVE
CURRENT OPTION CONTROL FILE: SYS1.PARMLIB(VTEXTABL)		
TSO COMPRESSION: ACTIVE	INBOUND COMPRESSION: ACTIVE	
	TS: RPJ	TOTAL SYSTEM
	-----	-----
TOTAL SEND REQUESTS .....	22	866
TOTAL RECEIVE REQUESTS .....	14	773
BYTES RECEIVED FROM LU .....	60	18,652
BYTES RECEIVED BY APPLICATION .....	289	25,226
IN-BOUND COMPRESSION PERCENT .....	79.24	26.07
BYTES SENT FROM APPLICATION .....	17,707	709877
BYTES SENT TO LU .....	9,318	345,311
OUT-BOUND COMPRESSION PERCENT .....	47.38	51.36
TOTAL BYTES COMPRESSED .....	8,618	371,140
PERCENT COMPRESSED .....	47.89	50.49
=>		
PF1-APPLS	PF2-VLU'S	PF3-EXIT
PF4-ACTIVATE	PF5-DEACTIVATE	PF6-BACKOUT
PF7-SET ON FOR USER	PF8-SET OFF FOR USER	PF9-RESET USER STATS
(C)OPYRIGHT SOFTOUCH SYSTEMS INC., 1988		

## THE TSO TERMINAL STATISTICS DISPLAY

On the display illustrated above, two columns of statistics are present. The first column is the current statistics for this terminal which is connected to TSO. The second column is the current statistics for all terminals and all applications which are participating in compression.

The following fields are present on this display:

**USER** This is the current terminal TSO user ID.

**NODE** The node name of the logical unit.

### COMPRESSION SYSTEM

This field contains ACTIVE, INACTIVE or QUIESCED indicating the status of compression for the entire system. QUIESCED means that a deactivate has been performed.

### CURRENT OPTION CONTROL FILE

This is the dataset and member name of the currently loaded customization table.

### TSO COMPRESSION

This field contains ACTIVE, QUIESCED or INACTIVE, indicating the status of compression for this terminal in TSO.

#### INBOUND COMPRESSION

This field contains ACTIVE or INACTIVE, indicating whether inbound compression is in effect or not.

#### TOTAL SEND REQUESTS

This is the number of outbound sends from the application to this LU.

#### TOTAL RECEIVE REQUESTS

This is the number of inbound receives from this LU to the application.

#### BYTES RECEIVED FROM LU

This is the total number of bytes of data transmitted from this LU to the application, prior to inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this number should be smaller than 'BYTES RECEIVED BY APPLICATION', else these two numbers will be equal. The difference in the two numbers is the number of bytes that did not have to be re-transmitted back from the terminal.

#### BYTES RECEIVED BY APPLICATION

This is the total number of inbound bytes of data that have been passed to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this is the number of bytes that would have been re-transmitted back from the LU if AUTOMON®-VTAM was not present.

#### IN-BOUND COMPRESSION PERCENT

This is the percent of savings using inbound compression. This number is calculated as 'BYTES RECEIVED FROM LU' divided by 'BYTES RECEIVED FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### BYTES SENT FROM APPLICATION

This is the total number of outbound bytes of data that have been sent from the application to the logical unit. This is the number of bytes that would have been transmitted if AUTOMON®-VTAM was not present. The difference in this number and 'BYTES SENT TO LU' is the savings obtained from outbound compression.

#### BYTES SENT TO LU

This is the total number of outbound bytes of data that have been sent to the LU from the application, prior to applying any outbound compression logic. This number is the number of bytes that would normally be sent to the terminal if outbound compression were not in effect. This number should be smaller than the previous number, 'BYTES SENT FROM APPLICATION'.

#### OUT-BOUND COMPRESSION PERCENT

This is the percent of savings using outbound compression. This number is calculated as 'BYTES SENT TO LU' divided by 'BYTES SENT FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### TOTAL BYTES COMPRESSED

This is the total number of bytes that have been eliminated from transmission by using both inbound and outbound compression. It is the sum of the differences between bytes sent and received, both inbound and outbound.

## PERCENT COMPRESSED

This is the overall percent of savings using both inbound and outbound compression. It is calculated as:

$$[100 \times (1 - ((\text{'BYTES RECEIVED FROM LU'} + \text{'BYTES SENT TO LU'}) / (\text{'BYTES RECEIVED FROM APPLICATION'} + \text{'BYTES SENT FROM APPLICATION'})))]$$

==> This is the command entry area. The only commands that can be entered at this point are the local trace commands, TRACE, TDISP and PURGE. Local trace is a facility for capturing a series of datastreams for your terminal, then displaying the datastreams on-line for analysis. Local trace is described in detail in section 07 - *USING THE VTAM TRACE FEATURE*.

## ENTERING COMMANDS AT THE TSO STATISTICS DISPLAY

Certain commands may be entered in the command area (==>). Press ENTER to input a command. There are five available commands. The first four apply to the local trace feature, which is documented in section 07 - *USING THE VTAM TRACE FEATURE*. These are:

TRACE  
TDISP  
PURGE  
TSAVE

The fifth command is NEWCOPY, which loads the VTEXMAIN module into a temporary address in the CSA, then alters the LPA directly to point to it. This allows a new version of the program to be used, which will stay in effect in the CSA until an IPL is performed. At that time, the new module will be loaded into the LPA.

**[Note]:** The old copy of VTEXMAIN is not (and cannot) be released. This is because there can be outstanding VTAM requests destined for the old module. However, once a NEWCOPY is performed, any request flowing to the old module is automatically rerouted to the new module. Since the old module is never released, repeated NEWCOPY requests will tie up more and more CSA storage (approximately 72K per NEWCOPY), which will not be released until an IPL is performed.

## PF KEY ACTIONS AT THE TSO STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. Throughout the TSO command processor, any time a PF key is listed, you may press either the designated PF key or its upper row equivalent. That is, PF1 is the same as PF13, PF2 = PF14, etc. The PF key options at this point are:

PF1-APPLS	Pressing PF1 will produce the statistics display by application. This display shows the current compression statistics for each application which is participating in compression. For a description of this screen, refer to <i>THE APPLICATION STATISTICS DISPLAY</i> , later in this section
PF2-VLU'S	Pressing PF2 will produce a display of all logical units which are coded in the customization table with a VLU statement. This display shows the logical unit name and whether it is included or excluded from compression. For a description of this screen, refer to <i>THE VLU TABLE DISPLAY</i> , later in this section.
PF3-EXIT	Pressing PF3 will exit altogether from VTEXPRES and return to the TSO command screen.
PF4-ACTIVATE	Pressing PF4 will cause an ACTIVATE command to be performed to activate compression system-wide. This would be equivalent to using the PARM=DSN(MEMBER) statement when running VTEXPRES as a batch program.

If compression has been activated at least once and if **PASSWORD** was coded in the customization table, you will be prompted for a password before reactivation will be allowed. The password must match the value coded in the customization table.

Upon entry, the following prompt message will display:

*VEXPR061. TYPE NEW OPTION DSN OR PRESS ENTER*

If the customization table has not yet been loaded, in other words, if an **ACTIVATE** or **DSN** command has not yet been done, you must enter the dataset and member name of the customization table.

If a previous activate has been done, this message allows you to load a different version of the customization table than is currently present, if desired.

If entered, the customization table must be entered as a fully qualified dataset and member name. That is, key the full dataset name with the member name in parentheses. Do not enclose the entry in quotes.

If a new customization table is not needed, press **ENTER** without keying anything. This will reload the customization table designated by **CURRENT OPTION CONTROL FILE**.

Upon receipt of the **ACTIVATE** command, the following functions are performed:

- 1). The dataset and member name of the customization table is dynamically allocated, if necessary.
- 2). The customization table is loaded, interpreted and the option table is built in common storage. If this is a reactivate, the common storage occupied by the previous table is freed.
- 3). If not already present, the compression 'hook' is placed in VTAM to allow AUTOMON®-VTAM to intercept outbound and inbound datastreams. (This occurs on first activate only).
- 4). The quiesce flag in the system option control block is turned off.
- 5). VTEXPRES returns to the display of the terminal compression statistics screen.

#### PF5-DEACTIVATE

Pressing PF5 will cause a **DEACTIVATE** command to be performed to quiesce compression system-wide. This would be equivalent to using the **PARM=DEACTIVATE** statement when running VTEXPRES as a batch program.

If **PASSWORD** was coded in the customization table, you will be prompted for a password before deactivation will be allowed. The password must match the value coded in the customization table.

Upon receipt of the **DEACTIVATE** command, the following functions are performed:

- 1). The quiesce flag in the system option table is set on, causing all traffic through AUTOMON®-VTAM to be routed to VTAM with no further action.
- 2). A message is output to the system log that compression has been deactivated.
- 3). VTEXPRES terminates normally.

#### PF6-BACKOUT

The BACKOUT command is used to terminate compression and remove the VTAM hooks.

If PASSWORD was coded in the customization table, you will be prompted for a password before backout will be allowed. The password must match the value coded in the customization table.

When PF6 is pressed, the following functions are performed:

- 1). The compression 'hook' which allows AUTOMON®-VTAM to intercept outbound and inbound datastreams is removed from VTAM.
- 2). A message is output to the system log that compression has been deactivated.
- 3). VTEXPRES terminates normally.

#### PF7-SET ON FOR USER

Pressing PF7 will reactivate compression for this terminal only. It only applies if a previous PF8 was issued to turn compression off for this user.

Upon entry of PF7, the following functions are performed:

- 1). The terminal is reset to participate in compression.
- 2). VTEXPRES returns to the display of the terminal compression statistics screen.

#### PF8-SET OFF FOR USER

Pressing PF8 will deactivate compression for this terminal only.

Upon entry of PF8, the following functions are performed:

- 1). The terminal is set to no longer participate in compression.
- 2). VTEXPRES returns to the display of the terminal compression statistics screen.

#### PF9-RESET USER STATS

Pressing PF9 will zero the compression statistics for this terminal only.

Upon receipt of the RESET command, the following functions are performed:

- 1). The compression statistics counters for this terminal are set to all zero.
- 2). VTEXPRES returns to the display of the terminal compression statistics screen.

## THE TSO APPLICATION STATISTICS DISPLAY

If PF1 is pressed from the TSO terminal statistics display, the following screen will display for the first application that is participating in compression:

VTAM-EXPRESS RELEASE 2.0, PTF LEVEL 91xxxx APPLICATION: TSO												
COMPRESSION ON	PSIZE	1920	ASIZE	3564	INBND	Y	EWIMG	B	EXTATTR	CPRIMG	Y	
SCRSI	ZE	CICS	NDFSUP	Y	STATS	Y	DELAY	04	RBGEN	Y	RUCONS	N
STATDDN			STATDSN									
TRACDDN			TRACDSN									
STARTED	12/28/90	15:10					TSO			TOTAL SYSTEM		
TOTAL SEND REQUESTS						457			763			
TOTAL RECEIVE REQUESTS						457			711			
BYTES RECEIVED FROM LU						6,442			10,790			
BYTES RECEIVED BY APPLICATION						34,583			38,931			
IN-BOUND COMPRESSION PERCENT						81.3			72.2			
BYTES SENT FROM APPLICATION						571,896			808,342			
BYTES SENT TO LU						135,853			336,716			
OUT-BOUND COMPRESSION PERCENT						76.2			58.3			
TOTAL BYTES COMPRESSED						464,184			499,767			
PERCENT COMPRESSED						76.5			58.9			
TOTAL CPU SECONDS USED						1.767			35.778			
AVG CPU SECONDS PER SEND/RECEIVE						.028			.026			
TOTAL STATIC STORAGE IN USE						5K			10K			
TOTAL DYNAMIC STORAGE IN USE						9K			9K			
TOTAL USERS LOGGED ON						5			11			
PF1=HELP	PF3=EXIT	PF4=LU'S	PF5=RESTART	PF6=RESET	PF7=PREV	PF8=NEXT						
PF9=SYSTEM STOP	PF10=APPL STOP	PF11=LU STOP	PF12=CHANGE OPTION									

On the display illustrated above, two columns of statistics are present. The first column is the current statistics for the application being displayed. The second column is the current statistics for all terminals and all applications which are participating in compression.

There will be one application statistics display available for every VAPPL statement present in the customization table. This includes the \$DEFAULT statement, whether it was explicitly coded or not. Thus, all applications which are defined to be included by default will have combined statistics displayed under the application name of \$DEFAULT.

The following fields are present on this display:

- APPLICATION** This is the application ID of which the statistics being displayed correspond.
- You may view the statistics of another application by keying the application ID in this field and pressing ENTER; or by pressing PF7 and PF8, to browse through the applications.
- COMPRESSION** This field contains an 'ON' or 'OFF', indicating the current status for this application. Compression will be off if any of the following conditions exist:
- 1). Compression is stopped for the system.
  - 2). Compression is stopped for the application.
  - 3). The APPLID is excluded with a VAPPL statement.
  - 4). The APPLID is included and compression is on but the applications was started before AUTOMON®-VTAM was activated.
- Applications that are specifically excluded in the customization table will not be present in the statistics display.
- PSIZE** PSIZE is the primary screen size default for this application. PSIZE will be used as the screen size for an LU if the bind image (LOGMODE) does not contain the primary size and there is not a VLU statement present which designates it.
- This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; however, it should properly correspond to the actual LU screen size. Typical screen sizes for 3270 devices are:
- |                             |                             |                             |                              |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
| model 2 - 1920<br>(24 x 80) | model 3 - 2560<br>(32 x 80) | model 4 - 3440<br>(43 x 80) | model 5 - 3564<br>(27 x 132) |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
- ASIZE** ASIZE is the alternate screen size default for this application. ASIZE will be used as the alternate screen size for an LU if the bind image (LOGMODE) does not contain the alternate size and there is not a VLU statement present which designates it.
- This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; however, it should properly correspond to the alternate LU screen size. Typical screen sizes for 3270 devices are:
- |                             |                             |                             |                              |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
| model 2 - 1920<br>(24 x 80) | model 3 - 2560<br>(32 x 80) | model 4 - 3440<br>(43 x 80) | model 5 - 3564<br>(27 x 132) |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
- INBND** This field contains a 'Y' or 'N', indicating whether inbound compression is active or not for this application. Inbound compression is the technique of turning off MDT in all outgoing fields, thereby reducing the amount of data transmitted back from the LU to only that which is keyed. All fields which had MDT on are inserted back into the inbound datastream passed to the application ID.
- You may change the field by keying a 'Y' or 'N' and pressing PF12.
- If inbound compression is not active, CPU overhead will be reduced slightly, however overall compression will be reduced significantly.
- If erased, 'Y' is the default.
- EWIMG** This field contains an 'N', 'F' or 'B', indicating whether outbound image compression is active for this application or not.
- This field may be changed by keying an entry and pressing PF12. Valid entries are:

- N - Image compression not active.
- F - Field-level compression active. When field boundaries change, the erase command is not removed.
- Y - Same as 'F'.
- B - Byte-level compression active. Field boundaries are ignored. Any outbound data is suppressed if already on screen.

EXTATTR This field contains a 'C', 'S' or 'L', indicating the type of compression done for extended attributes.

This field may be changed by keying an entry and pressing PF12. Valid entries are:

- C - Compress extended attribute if already on screen. Although some extended attribute combinations can not be suppressed, most can. This is the default if erased.
- S - Strip extended attributes and send the field attribute only. This should be used if you have applications that send extended attributes but have terminals that do not support them.
- L - Leave the extended attribute as-is.

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

CPRIMG This field contains a 'Y' or 'N', indicating whether or not the in-memory image of the LU screen is compressed as small as possible when the LU is inactive (between transmissions). CPRIMG will require slightly more CPU time, but will save an average of 1K per terminal in static storage usage.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. The default is 'N'.

SCRSIZE This field contains 'CICS' or spaces, indicating whether or not the primary and alternate screen sizes are to be determined from the CICS TCTTE. 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.

This field may be changed by keying a 'CICS' or spaces (indicating that it does not apply).

NDFSUP This field contains a 'Y' or 'N', indicating whether or not 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.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'Y' is the default.

STATS This field contains 'Y' or 'N', indicating whether or not statistics logging is in effect for this application. Statistics logging will write a record to the statistics file each time a user logs off of a terminal. The record contains the terminal statistics to that point.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'N' is the default.

DELAY This field contains a value from 00 to 99. DELAY is 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.

This field may be changed by keying a number from 00 to 99 and pressing PF12. A value of 00 means that logon-delay is not active. (compression will begin at logon). If erased, '00' is the default, meaning compression will begin at logon.

RBGEN	<p>(READ BUFFER GENERATION) This field contains a 'Y' or 'N', indicating whether or not the execution of read-buffer commands in the application will be suppressed. If activated, 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.</p> <p>Read-buffer commands are very time-consuming on a remote line since the entire terminal buffer is transmitted. Many session managers, on-line help products and other products make use of the read-buffer command.</p> <p>This field may be changed by keying a 'Y' or 'N' and pressing PF12. 'N' is the default if neither is present.</p> <p>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.</p> <p><b>[Note]:</b> If PASSWORD is coded on the VSTART statement, password entry will be required to change any of the fields listed above.</p>
STATDDN	<p>This field is the DDNAME for the statistics logging file. 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, thereby eliminating the need for a DD statement in the application JCL. If the STATDDN field is erased, statistics logging will not occur unless a DD statement is present in the application JCL.</p> <p>If a sequential file is to be used for statistics logging, rather than SMF, the DDNAME of the statistics logging file, should be entered here.</p>
STATDSN	<p>This field is the DSNAME for the statistics logging file. 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, thereby eliminating the need for a DD statement in the application JCL. If the STATDSN field is erased, statistics logging will not occur unless a DD statement is present in the application JCL.</p> <p>If a sequential file is to be used for statistics logging, rather than SMF, the DSNAME of the statistics logging file, should be entered here.</p>
TRACDDN	<p>This field is the DDNAME for the trace file. 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, thereby eliminating the need for a DD statement in the application JCL. If the TRACDDN field is erased, tracing will not occur unless a DD statement is present in the application JCL.</p> <p>If the trace feature is to be used, the DDNAME of the trace file should be entered here.</p>
TRACDSN	<p>This field is the DSNAME for the trace file. 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, thereby eliminating the need for a DD statement in the application JCL. If the TRACDSN field is erased, tracing will not occur unless a DD statement is present in the application JCL.</p> <p>If the trace feature is to be used, the DSNAME of the trace file should be entered here.</p> <p><b>[Note]:</b> If PASSWORD is coded on the VSTART statement, password entry will be required to change any of the fields listed above.</p>
STARTED	<p>This is the time and date that this application was last initialized with compression activated.</p>

If the RESET keyword is coded 'Y', this is the time and date that the statistics were last reset. Otherwise the statistics will be accumulated from the time that AUTOMON®-VTAM was last activated.

**[Note]:** For the following fields the left number applies to this application only, whereas the right number applies to the entire system.

#### TOTAL SEND REQUESTS

This is the number of outbound sends from the application to this LU.

#### TOTAL RECEIVE REQUESTS

This is the number of inbound receives from this LU to the application.

#### BYTES RECEIVED FROM LU

This is the total number of bytes of data transmitted from this LU to the application, prior to inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this number should be smaller than 'BYTES RECEIVED BY APPLICATION', else these two numbers will be equal. The difference in the two numbers is the number of bytes that did not have to be re-transmitted back from the terminal.

#### BYTES RECEIVED BY APPLICATION

This is the total number of inbound bytes of data that have been passed to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this is the number of bytes that would have been re-transmitted back from the LU if AUTOMON®-VTAM was not present.

#### IN-BOUND COMPRESSION PERCENT

This is the percent of savings using inbound compression. This number is calculated as 'BYTES RECEIVED FROM LU' divided by 'BYTES RECEIVED FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### BYTES SENT FROM APPLICATION

This is the total number of outbound bytes of data that have been sent from the application to the logical unit. This is the number of bytes that would have been transmitted if AUTOMON@-VTAM was not present. The difference in this number and 'BYTES SENT TO LU' is the savings obtained from outbound compression.

#### BYTES SENT TO LU

This is the total number of outbound bytes of data that have been sent to the LU from the application, prior to applying any outbound compression logic. This number is the number of bytes that would normally be sent to the terminal if outbound compression were not in effect. This number should be smaller than the previous number, 'BYTES SENT FROM APPLICATION'.

#### OUT-BOUND COMPRESSION PERCENT

This is the percent of savings using outbound compression. This number is calculated as 'BYTES SENT TO LU' divided by 'BYTES SENT FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### TOTAL BYTES COMPRESSED

This is the total number of bytes that have been eliminated from transmission by using both inbound and outbound compression. It is the sum of the differences between bytes sent and received, both inbound and outbound.

#### PERCENT COMPRESSED

This is the overall percent of savings using both inbound and outbound compression. It is calculated as:

$$[100 \times (1 - ((\text{'BYTES RECEIVED FROM LU'} + \text{'BYTES SENT TO LU'}) / (\text{'BYTES RECEIVED FROM APPLICATION'} + \text{'BYTES SENT FROM APPLICATION'})))]$$

#### TOTAL CPU SECONDS USED

This is the number of CPU seconds used by AUTOMON@-VTAM to process both inbound and outbound compression. It is measured by the difference in the clock time when the datastream is received and when it is passed on.

#### AVG CPU SECONDS PER SEND/RECEIVE

This is the average number of CPU seconds that AUTOMON@-VTAM used per inbound or outbound transmission. It is calculated as the total number of seconds used by the CPU, (both inbound and outbound), divided by the total number of sends and receives.

#### TOTAL STATIC STORAGE IN USE

Static storage is the storage used for terminal control blocks, terminal images and vectors. There is one vector per APPLID, and one terminal control block and image per LU. The image is freed when the CLEAR key is pressed.

#### TOTAL DYNAMIC STORAGE IN USE

Dynamic storage is the storage used for request blocks and miscellaneous dynamic areas. It is acquired when a transmission begins and is released at the end of the transmission. RU consolidation will keep request blocks across multiple sends.

#### TOTAL USERS LOGGED ON

This is the number of LUs that have been logged on to this application and are participating in compression. Excluded LUs are not counted.

## ENTERING COMMANDS AT THE TSO APPLICATION STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. Throughout the TSO command processor, any time a PF key is listed, you may press either the designated PF key or its upper row equivalent. That is, PF1 is the same as PF13, PF2 = PF14, etc. The PF key options at this point are:

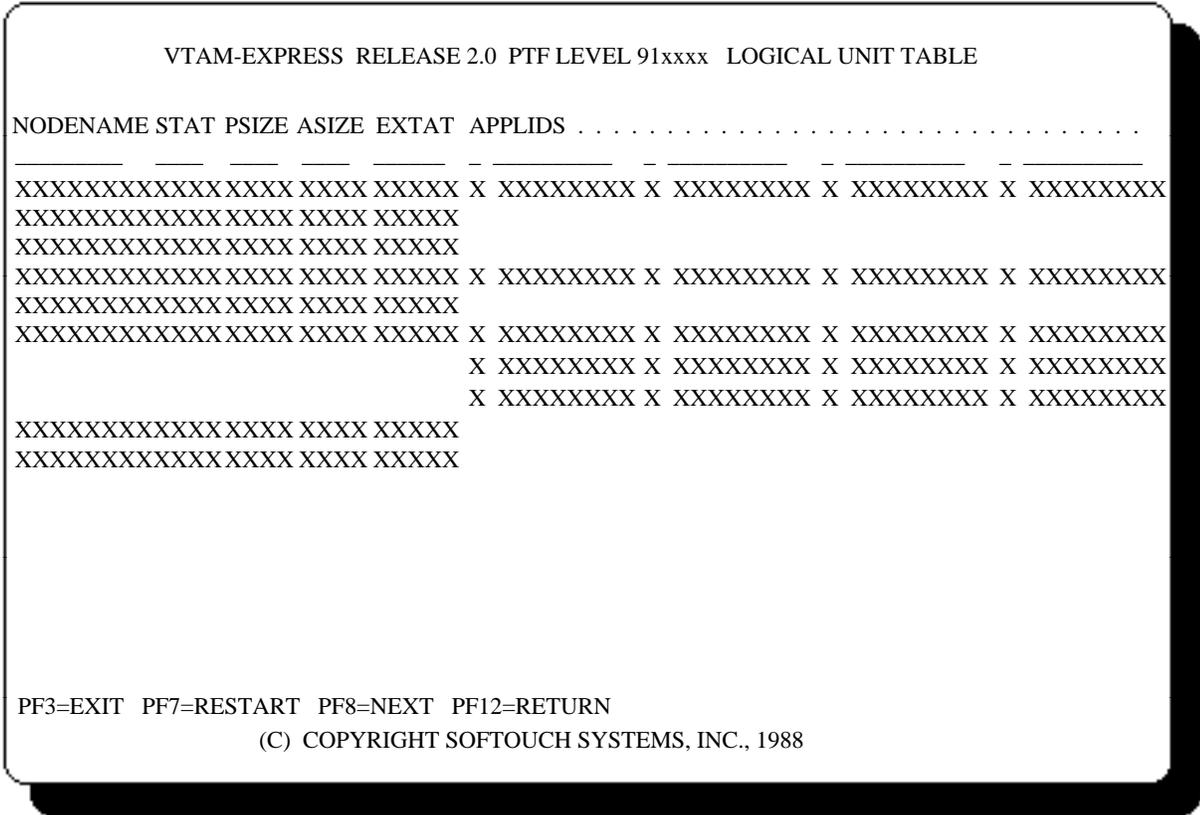
- PF1=HELP            Pressing PF1 will display a help screen. The help text shown will be dependent on the position of the cursor at the time PF1 was pressed. In other words, place the cursor in the field that you need help for, then press PF1. The help display appears in a window on the screen. When a help display is present, you can move the cursor to another field and press PF1 again for help on that field, or respond to the display with ENTER or any other PF key to remove the help window.
- PF3=EXIT            Pressing PF3 will exit VTEXPRES.
- PF4=LU'S            Pressing PF4 will invoke the LU display, which shows statistics by individual terminal for the application which is currently displayed. For a description of the LU display screen, see *THE TSO LU STATISTICS DISPLAY*, later in this section.
- PF5=RESTART        Pressing PF5 will display the first APPLID in the table.
- PF6=RESET          Pressing PF6 will reset the application statistics to zero. This will require password entry if PASSWORD is coded on the VSTART statement.
- PF7=PREV           Pressing PF7 will browse backward to the statistics display for the previous application.
- PF8=NEXT            Pressing PF8 will browse forward to the statistics display for the next application.
- ENTER=REFRESH    Pressing ENTER will refresh the current application display with any data that may have changed since the screen was last displayed.
- PF9=SYSTEM START/STOP  
                      Pressing PF9 will start compression for the system if it is stopped, else it will stop compression for the system if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.
- PF10=APPL START/STOP  
                      Pressing PF10 will start compression for the application if it is stopped, else it will stop compression for the application if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.
- PF11=LU START/STOP  
                      Pressing PF11 will start compression for this terminal if it is stopped, else it will stop compression for this terminal if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

PF12=CHANGE OPTIONS

Pressing PF12 will effect any changes you have made to any unprotected fields on this display. This will require password entry if PASSWORD is coded on the VSTART statement.

### THE VLU TABLE DISPLAY

If PF2 is pressed from the TSO terminal statistics display, the following screen will display showing the logical units which are coded in the customization table with VLU statements:



On the display illustrated above, six types of fields are present for each logical unit. They are:

- NODENAME This is the logical unit node name.
- STAT This field contains EXCL, INCL or SLCT, indicating whether this LU is to be excluded from or included in compression, or if compression is to be selective (dependant on the application).
- PSIZE This is the primary screen size as coded in the VLU statement.
- ASIZE This is the alternate screen size as coded in the VLU statement.
- EXTAT This field contains CMPRS, STRIP, or LEAVE, indicating the type of extended attribute compression to be performed on this logical unit. (For more information, see section 03 - CUSTOMIZATION under the topic VLU - IDENTIFY LOGICAL UNITS FOR COMPRESSION.)
- APPLIDS These fields will display data only if the STAT field contains SLCT. The single digit field contains an I or E, indicating whether this LU is to be excluded from or included in compression while this LU is logged onto the application designated by the following eight digit field.

Multiple pairs of fields may be present for one LU. If more than four pairs are present, then additional lines will be used to display the remainder of the selective inclusions or exclusions.

#### **ENTERING COMMANDS AT THE VLU TABLE DISPLAY**

The PF key list at the bottom of the screen displays the available options which can be performed at this point. They are:

- |             |  |
|-------------|--|
| PF3=EXIT    | Pressing PF3 will exit altogether from VTEXPRES and return to the TSO command screen.                          |
| PF7=RESTART | Pressing PF7 will restart the display at the top of the logical unit list.                                     |
| PF8=NEXT    | Pressing PF8 will browse forward to the display for the next 54 logical units in the table, if there are more. |
| PF12=RETURN | Pressing PF12 will cause a return to the TSO terminal statistics display and the main command options.         |

# THE CICS APPLICATION STATISTICS DISPLAY

Compression statistics can be viewed on-line for each of the applications in the customization table through the use of the VTEXDISP program in CICS. Statistics are kept for on-line viewing regardless of the specification of STATS in the customization table. That parameter only affects the disk logging of statistics.

Statistics are kept by application, but only for those applications specified in the customization table with a VAPPL statement. Any application which is implicitly included in compression by means of the \$DEFAULT VAPPL statement will still have compression statistics accumulated for it, but they will be kept in the table entry for \$DEFAULT. Thus, if you want separate statistics by identifiable application name, you should have a VAPPL statement in the table for every VTAM application.

The CICS transaction code to invoke the application statistics display is VTEX. It may be entered in any of the following ways to produce the associated results:

VTEX                    If no operands are entered, the display will begin with the current CICS application where VTEX is being operated.

VTEX,appl-ID            If the transaction is followed by a comma, then the full application ID, the display will begin with that application, if it appears in the table.

VTEX, generic appl-ID Any number of characters of an application-ID may be entered, whereupon the display will begin with the first occurrence of an application ID that matches those starting characters. Do not enter an asterisk or any other character to denote a generic entry.

When the VTEX transaction is invoked using any of the methods described above, the following screen will display for the requested application:

VTAM-EXPRESS RELEASE 2.0, PTF LEVEL 90xxxx APPLICATION: DBDCCICS										
COMPRESSION ON	PSIZE	ASIZE	3564	INBND	Y	EWIMG	BEXTATTR	C	CPRIMG	Y
SCRSI	ZE	CICS	NDFSUP	YSTATS	Y	DELAY	05BGEN	Y	RUCONS	N
STATDDN	CVSTATFL	STATDSN	STS.EXPRESS.CICS.STATS							
TRACDDN	CVTRACFL	TRACDSN	STS.EXPRESS.CICS.TRACE							
STARTED	01/10/91	15:07	DBDCCICS			TOTAL SYSTEM				
TOTAL SEND REQUESTS	.....				457			763		
TOTAL RECEIVE REQUESTS	.....				457			711		
BYTES RECEIVED FROM LU	.....				6,442			10,790		
BYTES RECEIVED BY APPLICATION	.....				34,583			38,931		
IN-BOUND COMPRESSION PERCENT	.....				81.3			72.2		
BYTES SENT FROM APPLICATION	.....				571,896			808,342		
BYTES SENT TO LU	.....				135,853			336,716		
OUT-BOUND COMPRESSION PERCENT	.....				76.2			58.3		
TOTAL BYTES COMPRESSED	.....				464,184			499,767		
PERCENT COMPRESSED	.....				76.5			58.9		
TOTAL CPU SECONDS USED	.....				1.767			35.778		
AVG CPU SECONDS PER SEND/RECEIVE	.....				.028			.026		
TOTAL STATIC STORAGE IN USE	.....				5K			10K		
TOTAL DYNAMIC STORAGE IN USE	.....				9K			9K		
TOTAL USERS LOGGED ON	.....				5			11		
PF1=HELP	PF3=EXIT	PF4=LU'S	PF5=RESTART	PF6=RESET	PF7=PREV	PF8=NEXT				
PF9=SYSTEM STOP	PF10=APPL STOP	PF11=LU STOP	PF12=CHANGE OPTION							

On the display illustrated above, two columns of statistics are present. The first column is the current statistics for the application being displayed. The second column is the current statistics for all terminals and all applications which are participating in compression.

There will be one application statistics display available for every VAPPL statement present in the customization table. This includes the \$DEFAULT statement, whether it was explicitly coded or not. Thus, all applications which are defined to be included by default will have combined statistics displayed under the application name of \$DEFAULT.

The following fields are present on this display:

- APPLICATION** This is the application ID of which the statistics being displayed correspond.
- You may view the statistics of another application by keying the application ID in this field and pressing ENTER; or by pressing PF7 and PF8, to browse through the applications.
- COMPRESSION** This field contains an 'ON' or 'OFF', indicating the current status for this application. Compression will be off if any of the following conditions exist:
- 1). Compression is stopped for the system.
  - 2). Compression is stopped for the application.
  - 3). The APPLID is excluded with a VAPPL statement.
  - 4). The APPLID is included and compression is on but the applications was started before AUTOMON®-VTAM was activated.
- Applications that are specifically excluded in the customization table will not be present in the statistics display.
- PSIZE** PSIZE is the primary screen size default for this application. PSIZE will be used as the screen size for an LU if the bind image (LOGMODE) does not contain the primary size and there is not a VLU statement present which designates it.
- This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; however, it should properly correspond to the actual LU screen size. Typical screen sizes for 3270 devices are:
- |                             |                             |                             |                              |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
| model 2 - 1920<br>(24 x 80) | model 3 - 2560<br>(32 x 80) | model 4 - 3440<br>(43 x 80) | model 5 - 3564<br>(27 x 132) |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
- ASIZE** ASIZE is the alternate screen size default for this application. ASIZE will be used as the alternate screen size for an LU if the bind image (LOGMODE) does not contain the alternate size and there is not a VLU statement present which designates it.
- This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; however, it should properly correspond to the alternate LU screen size. Typical screen sizes for 3270 devices are:
- |                             |                             |                             |                              |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
| model 2 - 1920<br>(24 x 80) | model 3 - 2560<br>(32 x 80) | model 4 - 3440<br>(43 x 80) | model 5 - 3564<br>(27 x 132) |
|-----------------------------|-----------------------------|-----------------------------|------------------------------|
- INBND** This field contains a 'Y' or 'N', indicating whether inbound compression is active or not for this application. Inbound compression is the technique of turning off MDT in all outgoing fields, thereby reducing the amount of data transmitted back from the LU to only that which is keyed. All fields which had MDT on are inserted back into the inbound datastream passed to the application ID.
- You may change the field by keying a 'Y' or 'N' and pressing PF12.

If inbound compression is not active, CPU overhead will be reduced slightly, however overall compression will be reduced significantly.

If erased, 'Y' is the default.

**EWIMG** This field contains an 'N', 'F' or 'B', indicating whether outbound image compression is active for this application or not.

This field may be changed by keying an entry and pressing PF12. Valid entries are:

**N** - Image compression not active.  
**F** - Field-level compression active. When field boundaries change, the erase command is not removed.  
**Y** - Same as 'F'.  
**B** - Byte-level compression active. Field boundaries are ignored. Any outbound data is suppressed if already on screen.

**EXTATTR** This field contains a 'C', 'S' or 'L', indicating the type of compression done for extended attributes.

This field may be changed by keying an entry and pressing PF12. Valid entries are:

**C** - Compress extended attribute if already on screen. Although some extended attribute combinations can not be suppressed, most can. This is the default if erased.  
**S** - Strip extended attributes and send the field attribute only. This should be used if you have applications that send extended attributes but have terminals that do not support them.  
**L** - Leave the extended attribute as-is.

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

**CPRIMG** This field contains a 'Y' or 'N', indicating whether or not the in-memory image of the LU screen is compressed as small as possible when the LU is inactive (between transmissions). CPRIMG will require slightly more CPU time, but will save an average of 1K per terminal in static storage usage.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. The default is 'N'.

**SCRSIZE** This field contains 'CICS' or spaces, indicating whether or not the primary and alternate screen sizes are to be determined from the CICS TCTTE. 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.

This field may be changed by keying a 'CICS' or spaces (indicating that it does not apply).

**NDFSUP** This field contains a 'Y' or 'N', indicating whether or not 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.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'Y' is the default.

**STATS** This field contains 'Y' or 'N', indicating whether or not statistics logging is in effect for this application. Statistics logging will write a record to the statistics file each time a user logs off of a terminal. The record contains the terminal statistics to that point.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'N' is the default.

DELAY	<p>This field contains a value from 00 to 99. DELAY is 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.</p> <p>This field may be changed by keying a number from 00 to 99 and pressing PF12. A value of 00 means that logon-delay is not active. (compression will begin at logon). If erased, '00' is the default, meaning compression will begin at logon.</p>
RBGEN	<p>(READ BUFFER GENERATION) This field contains a 'Y' or 'N', indicating whether or not the execution of read-buffer commands in the application will be suppressed. If activated, 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.</p> <p>Read-buffer commands are very time-consuming on a remote line since the entire terminal buffer is transmitted. Many session managers, on-line help products and other products make use of the read-buffer command.</p> <p>This field may be changed by keying a 'Y' or 'N' and pressing PF12. 'N' is the default if neither is present.</p> <p>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.</p> <p><b>[Note]:</b> See the discussion of SNA RU size considerations in the <i>INSTALLATION</i> section for more information.</p>
STATDDN	<p>This field is the DDNAME for the statistics logging file. 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, thereby eliminating the need for a DD statement in the application JCL. If the STATDDN field is erased, statistics logging will not occur unless a DD statement is present in the application JCL.</p> <p>If a sequential file is to be used for statistics logging, rather than SMF, the DDNAME of the statistics logging file, should be entered here.</p>
STATDSN	<p>This field is the DSNAME for the statistics logging file. 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, thereby eliminating the need for a DD statement in the application JCL. If the STATDSN field is erased, statistics logging will not occur unless a DD statement is present in the application JCL.</p> <p>If a sequential file is to be used for statistics logging, rather than SMF, the DSNAME of the statistics logging file, should be entered here.</p>
TRACDDN	<p>This field is the DDNAME for the trace file. 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, thereby eliminating the need for a DD statement in the application JCL. If the TRACDDN field is erased, tracing will not occur unless a DD statement is present in the application JCL.</p> <p>If the trace feature is to be used, the DDNAME of the trace file should be entered here.</p>
TRACDSN	<p>This field is the DSNAME for the trace file. 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, thereby eliminating the need for</p>

a DD statement in the application JCL. If the TRACDSN field is erased, tracing will not occur unless a DD statement is present in the application JCL.

If the trace feature is to be used, the DSNNAME of the trace file should be entered here.

**[Note]:** If PASSWORD is coded on the VSTART statement, password entry will be required to change any of the fields listed above.

#### STARTED

This is the time and date that this application was last initialized with compression activated.

If the RESET keyword is coded 'Y', this is the time and date that the statistics were last reset. Otherwise the statistics will be accumulated from the time that AUTOMON®-VTAM was last activated.

**[Note]:** For the following fields the left number applies to this application only, whereas the right number applies to the entire system.

#### TOTAL SEND REQUESTS

This is the number of outbound sends from the application to this LU.

#### TOTAL RECEIVE REQUESTS

This is the number of inbound receives from this LU to the application.

#### BYTES RECEIVED FROM LU

This is the total number of bytes of data transmitted from this LU to the application, prior to inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this number should be smaller than 'BYTES RECEIVED BY APPLICATION', else these two numbers will be equal. The difference in the two numbers is the number of bytes that did not have to be re-transmitted back from the terminal.

#### BYTES RECEIVED BY APPLICATION

This is the total number of inbound bytes of data that have been passed to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this is the number of bytes that would have been re-transmitted back from the LU if AUTOMON®-VTAM was not present.

#### IN-BOUND COMPRESSION PERCENT

This is the percent of savings using inbound compression. This number is calculated as 'BYTES RECEIVED FROM LU' divided by 'BYTES RECEIVED FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### BYTES SENT FROM APPLICATION

This is the total number of outbound bytes of data that have been sent from the application to the logical unit. This is the number of bytes that would have been transmitted if AUTOMON®-VTAM was not present. The difference in this number and 'BYTES SENT TO LU' is the savings obtained from outbound compression.

#### BYTES SENT TO LU

This is the total number of outbound bytes of data that have been sent to the LU from the application, prior to applying any outbound compression logic. This number is the number of bytes that would normally be sent to the terminal if outbound compression were not in effect. This number should be smaller than the previous number, 'BYTES SENT FROM APPLICATION'.

#### OUT-BOUND COMPRESSION PERCENT

This is the percent of savings using outbound compression. This number is calculated as 'BYTES SENT TO LU' divided by 'BYTES SENT FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### TOTAL BYTES COMPRESSED

This is the total number of bytes that have been eliminated from transmission by using both inbound and outbound compression. It is the sum of the differences between bytes sent and received, both inbound and outbound.

#### PERCENT COMPRESSED

This is the overall percent of savings using both inbound and outbound compression. It is calculated as:

$$[100 \times (1 - ((\text{'BYTES RECEIVED FROM LU'} + \text{'BYTES SENT TO LU'}) / (\text{'BYTES RECEIVED FROM APPLICATION'} + \text{'BYTES SENT FROM APPLICATION'})))]$$

#### TOTAL CPU SECONDS USED

This is the number of CPU seconds used by AUTOMON®-VTAM to process both inbound and outbound compression. It is measured by the difference in the clock time when the datastream is received and when it is passed on.

#### AVG CPU SECONDS PER SEND/RECEIVE

This is the average number of CPU seconds that AUTOMON®-VTAM used per inbound or outbound transmission. It is calculated as the total number of seconds used by the CPU, (both inbound and outbound), divided by the total number of sends and receives.

#### TOTAL STATIC STORAGE IN USE

Static storage is the storage used for terminal control blocks, terminal images and vectors. There is one vector per APPLID, and one terminal control block and image per LU. The image is freed when the CLEAR key is pressed.

#### TOTAL DYNAMIC STORAGE IN USE

Dynamic storage is the storage used for request blocks and miscellaneous dynamic areas. It is acquired when a transmission begins and is released at the end of the transmission. RU consolidation will keep request blocks across multiple sends.

## TOTAL USERS LOGGED ON

This is the number of LUs that have been logged on to this application and are participating in compression. Excluded LUs are not counted.

## ENTERING COMMANDS AT THE CICS APPLICATION STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. Throughout the VTEX transaction, any time a PF key is listed, you may press either the designated PF key or its upper row equivalent. That is, PF1 is the same as PF13, PF2 = PF14, etc. The PF key options at this point are:

- PF1=HELP      Pressing PF1 will display a help screen. The help text shown will be dependent on the position of the cursor at the time PF1 was pressed. In other words, place the cursor in the field that you need help for, then press PF1. The help display appears in a window on the screen. When a help display is present, you can move the cursor to another field and press PF1 again for help on that field, or respond to the display with ENTER or any other PF key to remove the help window.
- PF3=EXIT      Pressing PF3 will exit VTEXDISP.
- PF4=LU        Pressing PF4 will invoke the LU display, which shows statistics by individual terminal for the application which is currently displayed. For a description of the LU display screen, see *THE CICS LU STATISTICS DISPLAY*, later in this section.
- PF5=RESTART    Pressing PF5 will display the first APPLID in the table.
- PF6=RESET     Pressing PF6 will reset the application statistics to zero. This will require password entry if PASSWORD is coded on the VSTART statement.
- PF7=PREV      Pressing PF7 will browse backward to the statistics display for the previous application.
- PF8=NEXT      Pressing PF8 will browse forward to the statistics display for the next application.
- ENTER=REFRESH    Pressing ENTER will refresh the current application display with any data that may have changed since the screen was last displayed.
- PF9=SYSTEM START/STOP    Pressing PF9 will start compression for the system if it is stopped, else it will stop compression for the system if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

PF10=APPL START/STOP

Pressing PF10 will start compression for the application if it is stopped, else it will stop compression for the application if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

PF11=LU START/STOP

Pressing PF11 will start compression for this terminal if it is stopped, else it will stop compression for this terminal if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

PF12=CHANGE OPTIONS

Pressing PF12 will effect any changes you have made to any unprotected fields on this display. This will require password entry if PASSWORD is coded on the VSTART statement.

# THE CICS LU STATISTICS DISPLAY

If PF4 is pressed from the CICS application statistics display, the following screen will appear showing the logical unit statistics display. The terminals shown are the terminals that are active in the application which was displayed when PF4 was pressed.

FARTING LU ==> LOCL311				APPLID ==> DBDCCICS		
VTAM-EXPRESS LOGICAL UNIT STATISTICS						
LU AME	TOTAL BYTE COUNT	COMPRESSED BYTE COUNT	BYTES SAVED	PER CENT	NO. OF XMIT	BYTES SAVED PER XMIT
PPL	1,222,323	318,819	903,504	73.9	1,828	494
OCL311	342,651	101,550	241,101	70.3	623	387
OCL315	69,510	12,137	57,373	82.5	83	691
T2CIS01	4,979	2,405	2,574	51.6	5	514

PF1=HELP PF3=EXIT PF4=TOP PF5=APPL PF6=USERS PF7=PREV PF8=NEXT ENTER=REFRESH

On the display illustrated above, there are 15 rows, each row has seven fields. The first row, designated by 'APPL' in the LU name field, contains statistics for the entire application. The remaining rows contain statistical information for the logical units that are active in this application.

The fields present are:

**STARTING LU** This is the first logical unit present in the list. You can enter another LU name in this field and press ENTER, whereupon the display will refresh starting with the entered LU name or the closest name to it alphabetically.

**APPLID** This is the application ID in which the logical units of this display are active.

**LU NAME** This is the logical unit name.

**TOTAL BYTE COUNT** This is the total amount of bytes of data that would have been transmitted had AUTOMON@-VTAM not been present.

**COMPRESSED BYTE COUNT** This is the number of bytes of data that have been transmitted. This number would be equal to the TOTAL BYTE COUNT if compression was not active.

BYTES SAVED	This is the total number of bytes of data that have been suppressed from the data stream. These bytes would have been transmitted had AUTOMON®-VTAM not been present. This is calculated by subtracting 'COMPRESSED BYTE COUNT' from 'TOTAL BYTE COUNT'.
PERCENT	This is the compression percentage. It is arrived at by dividing 'BYTES SAVED' by 'TOTAL BYTE COUNT', then multiplying the quotient by 100.
NO. OF XMITS	This is the total number of transmissions both to and from this terminal.
BYTES SAVED PER XMIT	This is the average number of excess bytes of data per send or receive that would have been transmitted had AUTOMON®-VTAM not been present. It is calculated as 'BYTES SAVED' divided by 'NUMBER OF TRANSMISSIONS'.

### ENTERING COMMANDS AT THE CICS LU STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. They are:

PF1=HELP	Pressing PF1 will display a help screen. The help text shown will be dependent on the position of the cursor at the time PF1 was pressed. In other words, place the cursor in the field that you need help for, then press PF1. The help display appears in a window on the screen. When a help display is present, you can move the cursor to another field and press PF1 again for help on that field, or respond to the display with ENTER or any other PF key to remove the help window.
PF3=EXIT	Pressing PF3 will exit and return to the application display.
PF4=TOP	Pressing PF4 will return to the top of the logical unit list. The display will begin with the alphabetically first terminal for the application.
PF5=APPL	Pressing PF5 will return to the application statistics display.
PF6=USERS	Pressing PF6 will show the user IDs instead of LU names for each entry. User IDs are available for CICS and TSO only. Other applications will continue to display the LU name.

**[Note]:** For CICS, in order for the user ID to be displayed, the VAPPL statement for this CICS system must specify APTYPE=CICS or SCRSIZE=CICS.

- PF7=PREV      Pressing PF7 will browse backward to the display for the preceding 14 logical units.
- PF8=NEXT      Pressing PF8 will browse forward to the display for the next 14 logical units.
- ENTER=REFRESH      Pressing ENTER will refresh the current display with any information that may have changed since the screen was last displayed.

## LOADING A NEW CUSTOMIZATION TABLE

It may be the case that a new version of the customization table needs to be loaded after compression has been activated, in order to add or change any of the parameters therein.

This is accomplished by executing VTEXPRES again, either as a batch program or as a TSO command, then using the PARM=DSN(MEMBER) statement (batch program) or the PF4-ACTIVATE SYSTEM command (TSO). Either of these methods will allow you to specify a new dataset and member name of a customization table which will then be loaded and replace the previous table. Or you may simply change the existing member and reload it. You do not have to deactivate compression to accomplish this.

It's important to understand, however, that all of the options of the new table do not immediately go into effect. Some go into effect immediately, some go into effect when a user logs off and back on to an application, and some do not go into effect until the application is recycled or the ACB for the application is closed and reopened.

The following chart illustrates each of the three timing conditions described above, with the options of the table which pertain to that condition. Across the top of the chart are the major customization statements (VSTART, VAPPL, etc.). Down the side are all of the possible keywords that can be coded. At each intersection, one of the following codes indicates the timing for that option:

Immed - Takes effect immediately.  
 Logoff - Takes effect when user logs off and back on.  
 Recycle - Takes effect when the application is recycled or the application ACB is closed and reopened.

Verb-->	VSTART	VAPPL	VLU	VSKIP	VTRACE	VFTRAN
<b>Keyword</b>						
LU	Logoff					
PASSWORD	Immed					
MAXBUF	Immed					
STATMON	Recycle					
SMFID	Immed					
SMFUSER	Immed					
SMFSVC	Immed					
INCL		Recycle	Logoff			
EXCL		Immed	Logoff			
APTYPE		Immed				
ASIZE		Logoff	Logoff			
CPRIMG		Immed				
DELAY		Logoff				
EWIMG		Immed				
EXTATTR		Immed	Logoff			
FLDCLR		Immed				
FLDSTRIP		Immed				
INBND		Immed				
KEY		Immed				
NDFSUP		Immed				
PSIZE		Logoff	Logoff			
RBGEN		Immed				
RESET		Recycle				
RUCONS		Immed				
SCRSIZE		Immed				
STATS		Logoff				
STATDDN		Logoff				
STATDSN		Logoff				
TRACDDN		Immed				
TRACDSN		Immed				
WRAP		Immed				
VIO		Immed				
TEXT				Immed		
OFS				Immed		
ROWCOL				Immed		
TYPE				Immed		
COUNT				Immed		
LUNAME					Immed	
I/O					Immed	
START						Immed
STOP						Immed

## HANDLING PC TRANSFERS

If you have application programs that handle down-loading and up-loading of data from personal computers, there may be special handling that must be done for those applications. Compression must not be active during a PC transfer, since the normal 3270 datastream compression techniques will corrupt the data being transferred.

There are two methods of transparently handling file transfers:

- 1). 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.
- 2). AUTOMON®-VTAM can use the VFTRAN statements in the customization table to recognize a PC transfer when it occurs and automatically deactivate itself for that terminal until the transfer completes. The distributed customization table, VTEXSKEL, contains VFTRAN statements for the standard IBM file transfer package that is used in both TSO and the DOS/VSE Interactive Interface. If you are using a different file transfer package, you will need to code VFTRAN statements to designate the datastream characters that denote the start and end of the transfer operation. Refer to the VFTRAN statement in the *CUSTOMIZATION* section for a description of the coding requirements. This method will work whether the transfer program operates conversationally or not.

## RELOADING THE VTEXMAIN MODULE IN THE LPA

Any time you need to reload the VTEXMAIN module into the LPA, such as when you install a new version or apply a PTF, the way to permanently refresh the LPA module is to perform a re-IPL with the CLPA option.

However, you can load a temporary copy of VTEXMAIN into the CSA by using the NEWCOPY command of VTEXPRES.

The NEWCOPY command loads the VTEXMAIN module into a temporary address in the CSA, then alters the LPA directly to point to it. This allows a new version of the program to be used, which will stay in effect in the CSA until an IPL is performed. At that time, the new module will be loaded into the LPA.

**[Note]:** The old copy of VTEXMAIN is not (and cannot) be released. This is because there can be outstanding VTAM requests destined for the old module. However, once a NEWCOPY is performed, any request flowing to the old module is automatically rerouted to the new module. Since the old module is never released, repeated NEWCOPY requests will tie up more and more CSA storage (approximately 72K per NEWCOPY), which will not be released until an IPL is performed.

## OPERATION - DOS/VSE SYSTEMS

Once you have completed the installation, defined a customization table and loaded the SVA module, you are ready to activate AUTOMON®-VTAM and begin operation.

The primary operational program is VTEXPRES. This program is used to activate and deactivate compression, as well as perform other functions which may be needed from time to time. The functions handled by this program are:

- 1). Activate compression
- 2). Deactivate compression
- 3). Deactivate compression and remove hooks (BACKOUT)
- 4). Load a different customization table

A second on-line operational program is the VTEX transaction which runs in CICS. VTEX can be used to:

- 1). Display statistics for each included application.
- 2). Display statistics by terminal within each application.
- 3). Stop compression for the entire system.
- 4). Restart compression for the entire system.
- 5). Turn compression on or off for a specific application
- 6). Turn compression on or off for a specific LU
- 7). Alter the customization options for an application.
- 8). Reset compression statistics to zero
- 9). Activate local tracing for a terminal.
- 10). Display local trace records.
- 11). Purge local trace records.

In addition to the VTEXPRES program, you will want to print the compression statistics that are gathered throughout the day. This function is described in section 06 - *STATISTICS CAPTURE AND REPORTING*.

If you use the VTAM trace feature of the product, you will need to print the trace entries that are captured. The procedure for this is described in section 07 - *USING THE VTAM TRACE FEATURE*.

## OPERATION OF VTEXPRES

For day-to-day operation, you should set up a batch job or procedure to run VTEXPRES and activate compression each time VTAM is cycled. VTAM must be up before compression can be activated; however the application systems should not be up. This means that you only need to activate compression after bringing up VTAM or after deactivating it for some reason.

VTEXPRES must be initiated as a stand-alone job in an inactive partition. It does not permanently tie up the partition. It only runs for a few seconds. It terminates with a message on the system console indicating successful or unsuccessful completion.

The following JCL can be used to run VTEXPRES in batch :

```
* $$ JOB VTEXPRES,,C
// JOB VTEXPRES
// OPTION DUMP
// LIBDEF *,SEARCH=L.S [Note 1]
// DLBL IJSYS01,'WORK.FILE',,0 [Note 2]
// EXTENT SYS001,xxxxxx,1,0,yyyy,zzzz [Note 2]
// ASSGN SYS001,ddd [Note 2]
// EXEC VTEXPRES,PARM='option parameters' [Note 3]
* $$ SLI VEXPRS00 [Note 4]
/*
```

/&  
\* \$\$ EOJ

**[Notes]:**

1. This is the library where VTEXPRES was link-edited during installation. Programs STS0100 and STSPASS must either reside in this library or in a library which is included in the search string.
2. You must have a DLBL, EXTENT and ASSGN for a disk work file named IISYS01. Since this is one of the same work files that the ASSEMBLER and COBOL compiler uses, you may have standard labels already defined for it. The amount of space required should never exceed 10 tracks unless you have an exceptionally large customization table.
3. The PARM statements are discussed below.
4. This is the customization table, and must be present if the PARM statement is anything other than DEACTIVATE. You could use an SLI statement, as in this example, or a /INCLUDE statement (ICCF), or you could code the customization table in-line, following the EXEC statement.

## VTEXPRES PARM STATEMENTS

The PARM statement of the EXEC command is used to specify the type of function that VTEXPRES is to perform. Possible PARM statements are:

### **PARM='ACT[ivate]'**

Activate compression and load the customization table. There must be a customization table in SYSIPT when the ACTIVATE parm is used.

### **PARM='DEACT[ivate]'**

Deactivate compression by immediately quiescing. Deactivate allows immediate restart with an ACTIVATE command.

### **PARM='BACKOUT'**

Deactivate compression permanently by removing the intercept points from VTAM.

Following is a description of each of the PARM statement commands:

**ACT or ACTIVATE** The ACTIVATE command, which may be abbreviated as ACT, is used to activate compression or load a new version of the customization table. Upon receipt of the ACTIVATE command, the following functions are performed:

- 1). The customization table is loaded, interpreted and the option table is built in memory.
- 2). If not already present, the compression 'hook' is placed in VTAM to allow AUTOMON®-VTAM to intercept outbound and inbound datastreams. (This occurs on first activate only).
- 3). The quiesce flag in the system option control block is turned off.
- 4). A message is output to the system log that compression has been activated.
- 5). VTEXPRES terminates normally.

### **DEACT or DEACTIVATE**

The DEACTIVATE command, which may be abbreviated as DEACT, is used to stop or quiesce compression. After a deactivate, compression can be immediately restarted without recycling the applications or logging off and back on.

Upon receipt of the DEACTIVATE command, the following functions are performed:

- 1). The quiesce flag in the system option table is set on, causing all traffic through AUTOMON®-VTAM to be routed to VTAM with no further action.
- 2). A message is output to the system log that compression has been deactivated.
- 3). VTEXPRES terminates normally.

**BACKOUT** The BACKOUT command is used to terminate compression and remove the VTAM hooks.

Upon receipt of the BACKOUT command, the following functions are performed:

- 1). The quiesce flag in the system option table is set on, causing all traffic through AUTOMON®-VTAM to be routed to VTAM with no further action.

- 2). The compression 'hook' which allows AUTOMON@-VTAM to intercept outbound and inbound datastreams is removed from VTAM.
- 3). A message is output to the system log that compression has been deactivated.
- 4). VTEXPRES terminates normally.

## CONSIDERATIONS FOR ACTIVATION OF COMPRESSION

### **If this is the first activation or the first activation after a previous BACKOUT command:**

- When compression is activated, it will begin operating as soon as the application systems are initialized.
- If an application is already active before compression is activated, you must either recycle the application or close and re-open the VTAM ACB in use by the application. For CICS, the command to close and re-open the VTAM ACB is ...  
  
CSMT CLO,VTAM  
then, after receiving the message that the ACB is closed ...  
CSMT OPE,VTAM
- The best way to ensure that all terminals and applications will participate in compression is to activate AUTOMON@-VTAM prior to initializing the on-line applications.
- A second activation can be done without a deactivation in order to load a different copy of the customization table. In this case, all users who are participating in compression will continue to do so.

### **If this is a subsequent activation after a previous DEACTIVATE command:**

- When compression is activated, it will begin operating immediately for all users currently logged on who were participating in compression when the deactivate occurred.
- Compression will operate normally for all users who log on after the activation.
- You do not need to recycle the application, close and open the VTAM ACB or log off and back on. Compression will restart immediately for all included applications and LUs.

## CONSIDERATIONS FOR DEACTIVATION OF COMPRESSION

- When compression is deactivated with a DEACT command, it immediately stops operating for all connected logical units. It also stops all terminals which connect after the DEACT command from using compression.
- If an application is recycled while compression is deactivated, compression will not restart when the application initializes. If a subsequent ACTIVATE or DSN command is issued, compression will restart.
- Traffic still flows through AUTOMON@-VTAM after a DEACT command. The system is considered to be in the "quiesced" state and no action is taken on any inbound or outbound datastream. It is simply passed on to VTAM for normal processing.



When the VTEX transaction is invoked using any of the methods described above, the following screen will display for the requested application:

```

VTAM-EXPRESS RELEASE 2.0, PTF LEVEL 91xxxx  APPLICATION: DEVLICIS

COMPRESSION ON PSI ZE 1920 ASIZE 3564 INBND Y EWIMG B EXTATTR CCPRIMG Y
SCRSI ZE CICS NDFSUP Y STATS Y VIO Y DELAY 04 RBGEN Y RUCONS N

STARTED 12/31/90 19:33 DEVLICIS TOTAL SYSTEM
TOTAL SEND REQUESTS..... 457 763
TOTAL RECEIVE REQUESTS..... 457 711
BYTES RECEIVED FROM LU..... 6,442 10,790
BYTES RECEIVED BY APPLICATION..... 34,583 38,931
IN-BOUND COMPRESSION PERCENT..... 81.3 72.2
BYTES SENT FROM APPLICATION..... 571,896 808,342
BYTES SENT TO LU..... 135,853 336,716
OUT-BOUND COMPRESSION PERCENT.... 76.2 58.3
TOTAL BYTES COMPRESSED..... 464,184 499,767
PERCENT COMPRESSED..... 76.5 3.180
TOTAL CPU SECONDS USED..... 23.961 35.778
AVG CPU SECONDS PER SEND/RECEIVE.. .028 .026
TOTAL STATIC STORAGE IN USE..... 5K 10K
TOTAL DYNAMIC STORAGE IN USE..... 9K 9K
TOTAL USERS LOGGED ON..... 6 18
PF1=HELP PF3=EXIT PF4=LU'S PF5=RESTART PF6=RESET PF7=PREV PF8=NEXT
PF9=SYSTEM STOP PF10=APPL STOP PF11=LU STOP PF12=CHANGE OPTIONS

```

On the display illustrated above, two columns of statistics are present. The first column is the current statistics for the application being displayed. The second column is the current statistics for all terminals and all applications which are participating in compression.

There will be one application statistics display available for every VAPPL statement present in the customization table. This includes the \$DEFAULT statement, whether it was explicitly coded or not. Thus, all applications which are defined to be included by default will have combined statistics displayed under the application name of \$DEFAULT.

The following fields are present on this display:

- APPLICATION This is the application ID of which the statistics being displayed correspond.  
You may view the statistics of another application by keying the application ID in this field and pressing ENTER; or by pressing PF7 and PF8, to browse through the applications.
- COMPRESSION This field contains an 'ON' or 'OFF', indicating the current status for this application. Compression will be off if any of the following conditions exist:
  - 1). Compression is stopped for the system.
  - 2). Compression is stopped for the application.
  - 3). The APPLID is excluded with a VAPPL statement.
  - 4). The APPLID is included and compression is on but the applications was started before AUTOMON®-VTAM was activated.

Applications that are specifically excluded in the customization table will not be present in the statistics display.

PSIZE

PSIZE is the primary screen size default for this application. PSIZE will be used as the screen size for an LU if the bind image (LOGMODE) does not contain the primary size and there is not a VLU statement present which designates it.

This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; 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)
-----------------------------	-----------------------------	-----------------------------	------------------------------

ASIZE

ASIZE is the alternate screen size default for this application. ASIZE will be used as the alternate screen size for an LU if the bind image (LOGMODE) does not contain the alternate size and there is not a VLU statement present which designates it.

This field may be changed by keying the size and pressing PF12. Any value from 1920 to 9999 may be entered; 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)
-----------------------------	-----------------------------	-----------------------------	------------------------------

INBND This field contains a 'Y' or 'N', indicating whether inbound compression is active or not for this application. Inbound compression is the technique of turning off MDT in all outgoing fields, thereby reducing the amount of data transmitted back from the LU to only that which is keyed. All fields which had MDT on are inserted back into the inbound datastream passed to the application ID.

You may change the field by keying a 'Y' or 'N' and pressing PF12.

If inbound compression is not active, CPU overhead will be reduced slightly, however overall compression will be reduced significantly.

If erased, 'Y' is the default.

EWIMG This field contains an 'N', 'F' or 'B', indicating whether outbound image compression is active for this application or not.

This field may be changed by keying an entry and pressing PF12. Valid entries are:

N	-	Image compression not active.
F	-	Field-level compression active. When field boundaries change, the erase command is not removed.
Y	-	Same as 'F'.
B	-	Byte-level compression active. Field boundaries are ignored. Any outbound data is suppressed if already on screen.

EXTATTR This field contains a 'C', 'S' or 'L', indicating the type of compression done for extended attributes.

This field may be changed by keying an entry and pressing PF12. Valid entries are:

C	-	Compress extended attribute if already on screen. Although some extended attribute combinations can not be suppressed, most can. This is the default if erased.
S	-	Strip extended attributes and send the field attribute only. This should be used if you have applications that send extended attributes but have terminals that do not support them.
L	-	Leave the extended attribute as-is.

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

CPRIMG This field contains a 'Y' or 'N', indicating whether or not the in-memory image of the LU screen is compressed as small as possible when the LU is inactive (between transmissions). CPRIMG will require slightly more CPU time, but will save an average of 1K per terminal in static storage usage.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. The default is 'N'.

SCRSIZE This field contains 'CICS' or spaces, indicating whether or not the primary and alternate screen sizes are to be determined from the CICS TCTTE. 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.

This field may be changed by keying a 'CICS' or spaces (indicating that it does not apply).

NDFSUP This field contains a 'Y' or 'N', indicating whether or not 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.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'Y' is the default.

**STATS** This field contains 'Y' or 'N', indicating whether or not statistics logging is in effect for this application. Statistics logging will write a record to the statistics file each time a user logs off of a terminal. The record contains the terminal statistics to that point.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'N' is the default.

**VIO** This field contains 'Y' or 'N', indicating whether or not the VIO buffer pool usage option is in effect for this application. When VIO=Y, all terminal image storage is moved to the VIO buffer pool during periods of inactivity for the terminal. This feature greatly reduces the virtual storage requirement of AUTOMON®-VTAM.

If VIO=N, all terminal image storage is taken from the GETVIS area of the application partition.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. If erased, 'N' is the default.

**DELAY** This field contains a value from 00 to 99. DELAY is 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.

This field may be changed by keying a number from 00 to 99 and pressing PF12. A value of 00 means that logon-delay is not active. (compression will begin at logon). If erased, '00' is the default, meaning compression will begin at logon.

**RBGEN** (READ BUFFER GENERATION) This field contains a 'Y' or 'N', indicating whether or not the execution of read-buffer commands in the application will be suppressed. If activated, 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 managers, on-line help products and other products make use of the read-buffer command.

This field may be changed by keying a 'Y' or 'N' and pressing PF12. 'N' is the default if neither is present.

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.

**[Note]:** If PASSWORD is coded on the VSTART statement, password entry will be required to change any of the fields listed above.

**STARTED** This is the time and date that this application was last initialized with compression activated.

If the RESET keyword is coded 'Y', this is the time and date that the statistics were last reset. Otherwise the statistics will be accumulated from the time that AUTOMON®-VTAM was last activated.

**[Note]:** For the following fields the left number applies to this application only, whereas the right number applies to the entire system.

**TOTAL SEND REQUESTS**

This is the number of outbound sends from the application to this LU.

TOTAL RECEIVE REQUESTS

This is the number of inbound receives from this LU to the application.

BYTES RECEIVED FROM LU

This is the total number of bytes of data transmitted from this LU to the application, prior to inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this number should be smaller than 'BYTES RECEIVED BY APPLICATION', else these two numbers will be equal. The difference in the two numbers is the number of bytes that did not have to be re-transmitted back from the terminal.

#### BYTES RECEIVED BY APPLICATION

This is the total number of inbound bytes of data that have been passed to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed. If INBND=Y, this is the number of bytes that would have been re-transmitted back from the LU if AUTOMON@-VTAM was not present.

#### IN-BOUND COMPRESSION PERCENT

This is the percent of savings using inbound compression. This number is calculated as 'BYTES RECEIVED FROM LU' divided by 'BYTES RECEIVED FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### BYTES SENT FROM APPLICATION

This is the total number of outbound bytes of data that have been sent from the application to the logical unit. This is the number of bytes that would have been transmitted if AUTOMON@-VTAM was not present. The difference in this number and 'BYTES SENT TO LU' is the savings obtained from outbound compression.

#### BYTES SENT TO LU

This is the total number of outbound bytes of data that have been sent to the LU from the application, prior to applying any outbound compression logic. This number is the number of bytes that would normally be sent to the terminal if outbound compression were not in effect. This number should be smaller than the previous number, 'BYTES SENT FROM APPLICATION'.

#### OUT-BOUND COMPRESSION PERCENT

This is the percent of savings using outbound compression. This number is calculated as 'BYTES SENT TO LU' divided by 'BYTES SENT FROM APPLICATION', the quotient subtracted from 1, then the difference multiplied by 100.

#### TOTAL BYTES COMPRESSED

This is the total number of bytes that have been eliminated from transmission by using both inbound and outbound compression. It is the sum of the differences between bytes sent and received, both inbound and outbound.

#### PERCENT COMPRESSED

This is the overall percent of savings using both inbound and outbound compression. It is calculated as:

$$[100 \times (1 - ((\text{'BYTES RECEIVED FROM LU'} + \text{'BYTES SENT TO LU'}) / (\text{'BYTES RECEIVED FROM APPLICATION'} + \text{'BYTES SENT FROM APPLICATION'})))]$$

#### TOTAL CPU SECONDS USED

This is the number of CPU seconds used by AUTOMON@-VTAM to process both inbound and outbound compression. It is measured by the difference in the clock time when the datastream is received and when it is passed on.

#### AVG CPU SECONDS PER SEND/RECEIVE

This is the average number of CPU seconds that AUTOMON@-VTAM used per inbound or outbound transmission. It is calculated as the total number of seconds used by the CPU, (both inbound and outbound), divided by the total number of sends and receives.

#### TOTAL STATIC STORAGE IN USE

Static storage is the storage used for terminal control blocks, terminal images and vectors. There is one vector per APPLID, and one terminal control block and image per LU. The image is freed when the CLEAR key is pressed.

#### TOTAL DYNAMIC STORAGE IN USE

Dynamic storage is the storage used for request blocks and miscellaneous dynamic areas. It is acquired when a transmission begins and is released at the end of the transmission. RU consolidation will keep request blocks across multiple sends.

## TOTAL USERS LOGGED ON

This is the number of LUs that have been logged on to this application and are participating in compression. Excluded LUs are not counted.

## ENTERING COMMANDS AT THE CICS APPLICATION STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. Throughout the VTEX transaction any time a PF key is listed, you may press either the designated PF key or its upper row equivalent. That is, PF1 is the same as PF13, PF2 = PF14, etc. The PF key options at this point are:

**PF1=HELP** Pressing PF1 will display a help screen. The help text shown will be dependent on the position of the cursor at the time PF1 was pressed. In other words, place the cursor in the field that you need help for, then press PF1. The help display appears in a window on the screen. When a help display is present, you can move the cursor to another field and press PF1 again for help on that field, or respond to the display with ENTER or any other PF key to remove the help window.

**PF3=EXIT** Pressing PF3 will exit VTEXDISP.

**PF4=LU'S** Pressing PF4 will invoke the LU display, which shows statistics by individual terminal for the local CICS application (where VTEX is being operated). For a description of the LU display screen, see *THE CICS LU STATISTICS DISPLAY*, later in this section.

**PF5=RESTART** Pressing PF5 will display the statistics for the first APPLID in the table.

**PF6=RESET** Pressing PF6 will reset the application statistics to zero. This will require password entry if PASSWORD is coded on the VSTART statement.

**PF7=PREV** Pressing PF7 will browse backward to the statistics display for the previous application.

**PF8=NEXT** Pressing PF8 will browse forward to the statistics display for the next application.

**ENTER=REFRESH** Pressing ENTER will refresh the current application display with any data that may have changed since the screen was last displayed.

### PF9=SYSTEM START/STOP

Pressing PF9 will start compression for the system if it is stopped, else it will stop compression for the system if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

### PF10=APPL START/STOP

Pressing PF10 will start compression for the application if it is stopped, else it will stop compression for the application if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

### PF11=LU START/STOP

Pressing PF11 will start compression for this terminal if it is stopped, else it will stop compression for this terminal if it is currently running. This will require password entry if PASSWORD is coded on the VSTART statement.

### PF12=CHANGE OPTIONS

Pressing PF12 will effect any changes you have made to any unprotected fields on this display. This will require password entry if PASSWORD is coded on the VSTART statement.

## THE CICS LU STATISTICS DISPLAY

If PF4 is pressed from the CICS application statistics display, the following screen will appear showing the logical unit statistics display. The terminals shown are the terminals that are active in the local application only. The local application is the CICS application where you are operating the VTEX transaction. If you want to view terminal statistics for another CICS application, you must log-on to that CICS and initiate VTEX there.

On the display illustrated above, there are 15 rows, each row has seven fields. The first row, designated by 'APPL' in the LU name field, contains statistics for the entire application. The remaining rows contain statistical information for the logical units that are active in this application.

The fields present are:

STARTING LU	This is the first logical unit present in the list. You can enter another LU name in this field and press ENTER, whereupon the display will refresh starting with the entered LU name or the closest name to it alphabetically.
APPLID	This is the application ID in which the logical units of this display are active.
LU NAME	This is the logical unit name.

#### TOTAL BYTE COUNT

This is the total amount of bytes of data that would have been transmitted had AUTOMON@-VTAM not been present.

#### COMPRESSED BYTE COUNT

This is the number of bytes of data that have been transmitted. This number would be equal to the TOTAL BYTE COUNT if compression was not active.

#### BYTES SAVED

This is the total number of bytes of data that have been suppressed from the data stream. These bytes would have been transmitted had AUTOMON@-VTAM not been present. This is calculated by subtracting 'COMPRESSED BYTE COUNT' from 'TOTAL BYTE COUNT'.

#### PERCENT

This is the compression percentage. It is arrived at by dividing 'BYTES SAVED' by 'TOTAL BYTE COUNT', then multiplying the quotient by 100.

#### NO. OF XMITS

This is the total number of transmissions both to and from this terminal.

#### BYTES SAVED PER XMIT

This is the average number of excess bytes of data per send or receive that would have been transmitted had AUTOMON@-VTAM not been present. It is calculated as 'BYTES SAVED' divided by 'NUMBER OF TRANSMISSIONS'.

## ENTERING COMMANDS AT THE CICS LU STATISTICS DISPLAY

The PF key list at the bottom of the screen displays the available options which can be performed at this point. They are:

#### PF1=HELP

Pressing PF1 will display a help screen. The help text shown will be dependent on the position of the cursor at the time PF1 was pressed. In other words, place the cursor in the field that you need help for, then press PF1. The help display appears in a window on the screen. When a help display is present, you can move the cursor to another field and press PF1 again for help on that field, or respond to the display with ENTER or any other PF key to remove the help window.

#### PF3=EXIT

Pressing PF3 will exit and return to the application statistics display.

#### PF4=TOP

Pressing PF4 will return to the top of the logical unit list. The display will begin with the alphabetically first terminal for the application.

#### PF5=APPL

Pressing PF5 will return to the application statistics display.

- PF6=USERS            Pressing PF6 will show the user IDs instead of LU names for each entry. User IDs are available for CICS only. Other applications will continue to display the LU name.
- [Note]:** For CICS, in order for the user ID to be displayed, the VAPPL statement for this CICS system must specify APTYPE=CICS or SCRSIZE=CICS.
- PF7=PREV            Pressing PF7 will browse backward to the display for the preceding 14 logical units.
- PF8=NEXT            Pressing PF8 will browse forward to the display for the next 14 logical units.
- ENTER=REFRESH    Pressing ENTER will refresh the current display with any information that may have changed since the screen was last displayed.

## LOADING A NEW CUSTOMIZATION TABLE

It may be the case that a new version of the customization table needs to be loaded after compression has been activated, in order to add or change any of the parameters therein.

This is accomplished by executing VTEXPRES again, using the PARM=ACTIVATE statement. The new customization table on SYSIPT which will then be loaded and replace the previous table. You do not have to deactivate compression to accomplish this.

It's important to understand, however, that all of the options of the new table do not immediately go into effect. Some go into effect immediately, some go into effect when a user logs off and back on to an application, and some do not go into effect until the application is recycled or the ACB for the application is closed and reopened.

The following chart illustrates each of the three timing conditions described above, with the options of the table which pertain to that condition. Across the top of the chart are the major customization statements (VSTART, VAPPL, etc.). Down the side are all of the possible keywords that can be coded. At each intersection, one of the following codes indicates the timing for that option:

Immed	-	Takes effect immediately.
Logoff	-	Takes effect when user logs off and back on.
Recycle	-	Takes effect when the application is recycled or the application ACB is closed and reopened.

## HANDLING PC TRANSFERS

If you have application programs that handle down-loading and up-loading of data from personal computers, there may be special handling that must be done for those applications. Compression must not be active during a PC transfer, since the normal 3270 datastream compression techniques will corrupt the data being transferred.

There are two methods of transparently handling file transfers:

- 1). 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.
- 2). AUTOMON®-VTAM can use the VFTRAN statements in the customization table to recognize a PC transfer when it occurs and automatically deactivate itself for that terminal until the transfer completes. The distributed customization table, VTEXSKEL, contains VFTRAN statements for the standard IBM file transfer package that is used in both TSO and the DOS/VSE Interactive Interface. If you are using a different file transfer package, you will need to code VFTRAN statements to designate the datastream characters that denote the start and end of the transfer operation. Refer to the VFTRAN statement in the

*CUSTOMIZATION* section for a description of the coding requirements. This method will work whether the transfer program operates conversationally or not.

## RELOADING THE VTEXMAIN MODULE IN THE SVA

Any time you need to reload the VTEXMAIN module into the SVA, such as when you install a new version or apply a PTF, the safest way to do it is to perform a re-IPL.

However, you can perform another SET SDL to refresh VTEXMAIN in the SVA if you adhere carefully to the following procedure:

- 1). Do a BACKOUT command for AUTOMON®-VTAM.
- 2). For every application which is included in compression, either bring the application down and back up or close and open the VTAM ACB.
- 3). Once AUTOMON®-VTAM is backed-out and inactive for all applications you can reload the SVA module, then re-activate.

If you fail to follow this procedure and reload the SVA module while the old module is in use by an application, you will not be able to re-activate compression until you IPL. Upon attempting to activate, you will receive the following message:

*VEXPR011. - VTEXPRES - UNABLE TO LOCATE ATCVT POINTER*

## STATISTICS CAPTURE AND REPORTING

If statistics recording is in effect for AUTOMON®-VTAM, recording will begin as soon as VTEXPRES is run with an activate command. A terminal statistics record is output each time a user logs off, or when the application terminates, for each application where STATS=Y is coded.

To print the statistics reports, a print program is provided called VTEXSTAT. This program accepts a sequential file as input.

### MVS USERS

You cannot do statistical recording on applications that are running authorized path, which would be TSO, CICS when using the HPO option (high-performance option) and IMS when using the fast-path option.

## SMF STATISTICAL RECORDING

To capture statistics in the SMF recording file, you must code the SMFID on the VSTART statement and optionally you can code SMFSVC and SMFUSER. Then you must specify STATS=Y on each VAPPL statement where statistics recording is desired.

If SMF recording is used to capture statistics, you must run IFASMFDP to extract the AUTOMON®-VTAM statistics records and output these to a sequential file. This file can then be input to VTEXSTAT.

The procedure for handling and extracting SMF records is described in the MVS SYSTEM MANAGEMENT FACILITIES (SMF) manual, but the following example can serve as the JCL required to run the SMF extract:

```
//STEP1      EXEC  PGM=IFASMFDP
//SYSPRINT   DD    SYSOUT=*
//DUMPIN     DD    DSN=SYS1.MANX,                [Note 1]
//           DD    DISP=SHR
//DUMPOUT    DD    DSN=SMF.SEQ.DATA,            [Note 2]
//           DD    DISP=SHR
//SYSIN      DD    *
              INDD(DUMPIN,OPTIONS(DUMP))
              OUTDD(DUMPOUT,TYPE(xxx))        [Note 3]
/*
```

#### [Notes]:

1. This is the main SMF file containing all SMF records.
2. This is the output sequential file to contain the AUTOMON®-VTAM SMF records.
3. The xxx must correspond to the value coded for SMFID in the VSTART statement of the customization table.

## SEQUENTIAL FILE STATISTICAL RECORDING

To use a sequential file for statistics recording, you must do the following:

- 1). Allocate a sequential file for each application where statistics are to be recorded.
- 2). Code the STATDSN keyword on the VAPPL statement of each application identifying the DSNAME of the sequential statistics file.

#### [Note 1]

- 3). Code STATS=Y and STATDDN=xxxxxxx on each VAPPL statement where statistics are to be recorded. The xxxxxxxx is the DDNAME of the sequential file for that application.

You can either allocate the file through TSO or use JCL to create it when the application starts up. If JCL is used, the DD statement needs to appear as follows:

```
//xxxxxxx      DD      DSN=EXPRESS.STATS.FILE,
//              DISP=(NEW,CATLG),           [Note 2]
//              UNIT=SYSDA,
//              SPACE=(CYL,(2,1))         [Note 3]
```

Do not specify any DCB parameters. The correct values will be plugged into the DCB at open time.

If you use TSO to allocate the file, use the following attributes:

File type	- PS (Physical-sequential)
Record format (RECFM)	- VB (Variable-blocked)
Block size (BLKSIZE)	- 32704
Record size (RECSIZE)	- 32700
File size	- see <b>Note 3</b> , below

**[Notes]:**

1. The presence of STATDSN on the VAPPL statement causes AUTOMON®-VTAM to dynamically allocate the statistics file when it is opened. You can also include a DD statement in the start-up JCL of the application with a DDNAME which corresponds to the STATDDN value if you prefer.
2. You may want to use DISP=(NEW,KEEP), then delete the file after you run the print job. If you do catalog it and never delete it, you must change your DD statement to DISP=SHR after it is cataloged.
3. Statistics records are about 60 characters in length. One record is created for each LU each time the user logs off of the application. Calculate file size according to the number of users involved.

## RUNNING THE STATISTICS PRINT PROGRAM

When the VTAM application terminates, run the SMF extract if SMF recording is being used. If a sequential file is being used, you are ready for the next step.

Once you have the statistics records in a sequential file, this file must first be sorted to LU node name within application sequence, then it can be input to the VTEXSTAT program. The following JCL can be used to sort the file and run VTEXSTAT:

```
//STEP1      EXEC  PGM=SORT
//SYSOUT     DD    SYSOUT=*
//SORTIN     DD    DSN=SMF.SEQ.DATA,       [Note 1]
//           DD    DISP=SHR
//SORTOUT    DD    DSN=SMF.SEQ.DATA,       [Note 3]
//           DD    DISP=SHR
//SORTWK1    DD    UNIT=SYSDA,           [Note 4]
//           DD    SPACE=(TRK,(100,100))
//SYSIN      DD    *
//           SORT FIELDS=(23,16,A),FORMAT=CH
//           /*
//STEP2      EXEC  PGM=VTEXSTAT
//STEPLIB    DD    DSN=xxxx.LOADLIB,      [Note 2]
//           DD    DISP=SHR
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=SMF.SEQ.DATA,      [Note 3]
//           DD    DISP=SHR
//SYSIN      DD    *
```

option parameters described below  
/\*

The option parameters on SYSIN can be one of the following, or may be omitted if no options are desired.

- NORND - The byte counts are normally shown as increments of 1000 to reduce the size of the numbers. This option will cause the full number to print without dividing by 1000.
- NOSUM - Statistic records are normally summarized by LU name within application ID so that one line prints per LU. This option will stop the summary logic and print every record as it appears.

**[Notes]:**

1. This is the input sequential file containing the AUTOMON@-VTAM records. If you like, you can sort several statistics files from different applications together to create a consolidated report.
2. This is the library where VTEXSTAT was link-edited.
3. This is the sorted sequential file containing the AUTOMON@-VTAM records. This example sorts and places the sorted output in the input file. You can use a different output file DSNAME if desired.
4. This is a sort work file.

## DOS/VSE USERS

You must have a DLBL and EXTENT for the file, VTX\$STA, either in partition standard labels or in the start-up JCL for each application where statistics are to be recorded. There must be a permanent or temporary ASSGN statement for the system logical unit coded on the EXTENT statement.

## RUNNING THE STATISTICS PRINT PROGRAM

When the VTAM application terminates you are ready to run the statistics print program.

Once you have the statistics records in a sequential file, this file must first be sorted to LU node name within application sequence, then it can be input to the VTEXSTAT program. The following JCL can be used to sort the file and run VTEXSTAT:

```
// JOB SORT STATISTICS
// DLBL SORTIN1,'VTAM.EXPRESS.STATISTICS',0
// EXTENT SYS006 [Note 1]
// ASSGN SYS006,ddd
// DLBL SORTOUT,'SORTED.STATISTICS',0
// EXTENT SYS001,,1,0,yyyy,zzzz [Note 3]
// ASSGN SYS001,ddd
// DLBL SORTWK1,'SORT.WORK.FILE',0
// EXTENT SYS002,,1,0,yyyy,zzzz [Note 4]
// ASSGN SYS002,ddd
// EXEC SORT,SIZE=200K
// SORT FIELDS=(23,16,A),FORMAT=CH
// INPFIL EXIT
// OUTFIL BLKSIZE=6200
// RECORD TYPE=F,LENGTH=62
// MODS PH1=(VTEXSORT,L1024,E15)
// END
/*
```

```
// LIBDEF *,SEARCH=L.S [Note 2]
// DLBL SYSUT1,'SORTED.STATISTICS',0
// EXTENT SYS001,,1,0,yyyy,zzzz
// EXEC VTEXSTAT,SIZE=AUTO
option parameters described below
/*
```

[Note 3]

The option parameters on SYSIPT can be one of the following, or may be omitted if no options are desired.

- NORND - The byte counts are normally shown as increments of 1000 to reduce the size of the numbers. This option will cause the full number to print without dividing by 1000.
- NOSUM - Statistic records are normally summarized by LU name within application ID so that one line prints per LU. This option will stop the summary logic and print every record as it appears.

**[Notes]:**

1. This is the input sequential file containing the AUTOMON®-VTAM records. If you like, you can sort several statistics files from different applications together to create a consolidated report.
2. This is the library where VTEXSTAT was link-edited.
3. This is the sorted sequential file containing the AUTOMON®-VTAM records.
4. This is a sort work file.

## THE STATISTICS REPORT

The statistics report produced by VTEXSTAT shows all statistics records in sequence by application ID followed by logical unit node name.

The following fields are listed on the report:

- APPLIC. ACB NAME This is the ACB name of the application to which the LU was connected.
- L.U. NODE NAME This is the node name of the connected logical unit.
- LAST LOG DATE This is the date of the last record present for this application and LU node name. Unless the NOSUM option is in effect, it could be possible that records for one LU could span two different days. This would be the date present on the last record for that LU.
- LAST LOG TIME This is the time of the last record present for this application and LU node name. Unless the NOSUM option is in effect, records for one LU could each contain a different time of day. This would be the time present on the last record for that LU.
- TOTAL SENDS FROM LU This is the number of times that a datastream has been received from this logical unit.
- TOTAL SENDS FROM AP This is the number of times that data has been received from an application and sent to the LU.

#### BYTES RECEIVED FROM LU

This is the total number of bytes of data that have been received from the logical unit, prior to inserting any fields which had modified-data-tag set on when the screen was displayed. This number indicates the savings obtained from inbound compression logic. If you look at BYTES SENT TO APPLICATION, the difference in the two numbers is the number of bytes that did not have to be re-transmitted back from the terminal.

#### BYTES SENT TO AP

This is the total number of bytes of data that have been sent to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed. This number indicates the number of bytes that would have been re-transmitted back from the terminal if inbound compression were not in effect.

#### BYTES RECEIVED FROM AP

This is the total number of bytes of data that have been received from the application, prior to applying any outbound compression logic. This number is the number of bytes that would normally be sent to the terminal if outbound compression were not in effect.

#### BYTES SENT TO LU

This is the total number of bytes of data that have been sent to the logical unit, after applying outbound compression logic. The difference in this number and BYTES RECEIVED FROM APPLICATION is the savings obtained from outbound compression.

#### PERCENT INBOUND COMPRESS

This is the compression percentage for inbound compression. It is arrived at by dividing the number of bytes received from the LU by the number of bytes sent to the application, then multiplying the quotient by 100.

#### PERCENT OUTBOUND COMPRESS

This is the compression percentage for outbound compression. It is arrived at by dividing the number of bytes sent to the LU by the number of bytes received from the application, then multiplying the quotient by 100.

## USING THE VTAM TRACE FEATURE

The trace feature of AUTOMON®-VTAM is provided primarily as a debugging tool for Unicom Systems Inc. Systems technical support personnel if any problems arise with applications running with compression. It is an improvement over the standard VTAM trace in that it captures the full buffer image rather than only the first 256 bytes. It could be used for other purposes any time a VTAM trace is desired.

## GLOBAL AND LOCAL TRACING

There are two types of trace facilities in AUTOMON®-VTAM. The GLOBAL trace captures terminal datastreams at the system level, crossing application boundaries. Global trace records are output to a sequential file which is then input to a batch print program.

LOCAL tracing can be performed at the terminal. It is a very quick and easy method of capturing a series of datastreams, then displaying the trace records on the terminal. Local trace records are kept in memory and never output to a file. There is no print facility for local trace records, they must be displayed on a terminal. You could, of course, do screen prints to obtain a hard-copy of local trace records. Local trace records are released when the PURGE command is issued at the terminal.

## GLOBAL TRACING - MVS USERS

Global trace cannot be used by applications which run authorized path under VTAM. This would be TSO, IMS when running the fast-path option and CICS when using the high performance option, and you may have other applications running authorized path. If this is the case and you need a trace, you can use the local trace option, or you will have to use GTF.

## DEFINING THE GLOBAL TRACE FILE

Do the following to define a sequential file for use with the global trace feature:

- 1). Allocate a sequential file for each application where statistics are to be recorded. Trace records are captured at the application level so you could have more than one trace file if you need to run trace for different applications at the same time.

You can either allocate the file through TSO or use JCL to create it when the VTAM application starts up. If JCL is used, the DD statement needs to appear as follows:

```
//xxxxxxx    DD      DSN=EXPRESS.TRACE.FILE,  
//           DISP=(NEW,CATLG),  
//           UNIT=SYSDA,  
//           SPACE=(CYL,(4,1))           [Note 2]
```

The DDNAME must be the same as coded for TRACDDN on the VAPPL statement for this application. Do not specify any DCB parameters. The correct values will be plugged into the DCB at open time.

If you use TSO to allocate the file, use the following attributes:

File type	- PS (Physical-sequential)
Record format (RECFM)	- VB (Variable-blocked)
Block size (BLKSIZE)	- 32704
Record size (RECSIZE)	- 32700
File size	- see <b>Note 2</b> , below

- 2). Code the TRACDSN keyword on the VAPPL statement of each application identifying the DSNAME of the sequential trace file.

**[Note 1]**

- 3). Code TRACDDN=xxxxxxx on each VAPPL statement where tracing is to be done. The xxxxxxx is the DDNAME of the trace file for that application.

**[Notes]:**

1. The presence of TRACDSN on the VAPPL statement causes AUTOMON®-VTAM to dynamically allocate the statistics file when it is opened and deallocate it when closed.
2. Trace records can be quite large, since they represent entire datastreams. There is a minimum of two trace records output per inbound or outbound transmission. If the diagnostic trace is being run, there will be 8 or 9 records output per transmission. The amount of disk space needed depends on the length of time that you leave trace active, but it is a good idea to allocate 3 or 4 cylinders, at least.

## ACTIVATING GLOBAL TRACE

You activate global trace by including a VTRACE statement for the desired LU name in the customization table. Since you would not normally want trace to run all of the time, you can define an alternate customization table which has a VTRACE statement in it, or simply add the VTRACE statement to your production table, then do an ACTIVATE command on-line, or a DSN command in batch, using the VTEXPRES program, to start the trace operating.

The following conditions must be met in order for global trace to operate:

- A VTRACE statement for the LU to be traced must be present in the customization table.
- The VAPPL statement must contain the TRACDDN keyword, specifying the DDNAME of the trace sequential file and the TRACDSN keyword, specifying the DSNAME of the trace file.
- There must be a sequential trace file defined with the same DSNAME as the TRACDSN specification.
- The application where tracing is to be performed must not run authorized path.

If any of the conditions above are not met, no trace records will be output.

Trace records are output for each send and receive from both the terminal and the application. Thus, for every transaction there will be four trace records, as follows:

- 1). The data sent from the terminal.
- 2). The data received by the application.
- 3). The data sent from the application.
- 4). The data received at the terminal.

Trace records contain an identifier as to the traffic point of this buffer (sent from terminal, received by application, etc.) and the buffer image that was sent or received. In addition, the LU name, date and time stamp and the data length are recorded.

Trace records begin recording as soon as a customization table is loaded with VTRACE statements present and the terminal identified in the VTRACE LUname logs on to an application. If the LU is already logged-on, trace records will begin recording with the next input or output received at that LU. A message is output to the system console when the trace file is opened.

## TERMINATING GLOBAL TRACE

To stop tracing, do an activate command in VTEXPRES (either batch or on-line) and load a customization table with no (or different) VTRACE statements present. This will immediately stop trace recording for the previous VTRACE statements. If the newly loaded customization table contains no VTRACE statements, the trace file will be automatically closed and deallocated. A message is output to the system console when the trace file is closed.

If the trace sequential file fills up while operating, action is taken based on the WRAP keyword of the VAPPL statement. If WRAP=Y is coded, tracing will restart at the beginning of the file. The previous trace records up to that point are lost. If WRAP=N is coded, tracing will cease when the end of the file is reached.

## RUNNING THE TRACE PRINT PROGRAM

You can run the trace print program as soon as you terminate global trace. You must load a customization table with no VTRACE statements present to do this. You will receive a message on the system console when the trace file has been closed and deallocated.

To print the trace records, run the VTEXTRAC program with the trace file as the input file. The following JCL can be used:

```
//STEP1      EXEC  PGM=VTEXTRAC
//STEPLIB    DD    DSN=xxx.LOADLIB,           [Note 1]
//           //    DISP=SHR
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=xxx.TRACE.DATA,       [Note 2]
//           //    DISP=SHR
//SYSIN      DD    *                          [Note 3]
/*
```

### [Notes]:

1. This is the load library where VTEXTRAC was link-edited.
2. This is the input sequential file containing the AUTOMON®-VTAM trace records.
3. VTEXTRAC does not currently use any control statements, however the SYSIN DD has been provided for future enhancement.

## GLOBAL TRACING - DOS/VSE USERS

Global trace can be used by any application in DOS/VSE. There are no limitations.

## DEFINING THE GLOBAL TRACE FILE

Do the following to define a sequential file for use with the global trace feature:

Provide a DLBL and EXTENT statement for the file, VTX\$TRA, in the JCL for the application system where the LU to be traced is to be used, or add the labels to standard or partition standard labels.

Trace records can be quite large, since they represent entire datastreams. There is a minimum of two trace records output per inbound or outbound transmission. If the diagnostic trace is being run, there will be 8 or 9 records output per transmission. The amount of disk space needed depends on the length of time that you leave trace active, but it is a good idea to allocate 3 or 4 cylinders, at least.

## ACTIVATING GLOBAL TRACE

You activate global trace by including a VTRACE statement for the desired LU name in the customization table. Since you would not normally want trace to run all of the time, you can define an alternate customization table which has a VTRACE statement in it, or simply add the VTRACE statement to your production table, then run the VTEXPRES program with an ACTIVATE command, to start the trace operating.

The following conditions must be met in order for global trace to operate:

- A VTRACE statement for the LU to be traced must be present in the customization table.
- A DLBL and EXTENT statement for the sequential trace file must be present in the partition where the VTAM application runs. The file ID of the trace file is VTX\$TRA.

If either of the conditions above are not met, no trace records will be output.

Trace records are output for each send and receive from both the terminal and the application. Thus, for every transaction there will be four trace records, as follows:

- 1). The data sent from the terminal.
- 2). The data received by the application.
- 3). The data sent from the application.
- 4). The data received at the terminal.

Trace records contain an identifier as to the traffic point of this buffer (sent from terminal, received by application, etc.) and the buffer image that was sent or received. In addition, the LU name, date and time stamp and the data length are recorded.

Trace records begin recording as soon as a customization table is loaded with VTRACE statements present and the terminal identified in the VTRACE LUname logs on to an application. If the LU is already logged-on, trace records will begin recording with the next input or output received at that LU. A message is output to the system console when the trace file is opened

## TERMINATING GLOBAL TRACE

To stop tracing, run VTEXPRES with an ACTIVATE command and load a customization table with no (or different) VTRACE statements present. This will immediately stop trace recording for the previous VTRACE statements. If the newly loaded customization table contains no VTRACE statements, the trace file will be automatically closed. A message is output to the system console when the trace file is closed.

If the trace sequential file fills up while operating, tracing will cease.

## RUNNING THE TRACE PRINT

### PROGRAM

To print the trace records, run the VTEXTRAC program with the trace file as the input file. The following JCL can be used:

```
// JOB PRINT TRACE
// LIBDEF *,SEARCH=L.S [Note 1]
// UPSI 11111111
// DLBL SYSUT1,'EXPRESS.TRACE.FILE',0 [Note 2]
// EXTENT SYS001
// ASSGN SYS001,ddd
// EXEC VTEXTRAC,SIZE=AUTO
/*
```

#### [Notes]:

1. This is the load library where VTEXTRAC was link-edited.
2. This is the input sequential file containing the AUTOMON®-VTAM trace records.

## I/O TRACES AND DIAGNOSTIC TRACES

There are two types of global trace functions available in AUTOMON®-VTAM. These are the I/O trace and the diagnostic trace. If you code 'I/O' or 'IO' as the operand of the VTRACE statement, you will get the I/O trace, which consists only of the datastreams sent and received (the four record types mentioned above).

If you do not code any operand on the VTRACE statement you will get a diagnostic trace, which consists of the I/O trace plus a snap dump of the various control blocks in use by AUTOMON®-VTAM at the moment. The diagnostic trace is used by Unicom Systems Inc. Systems technical support personnel in analyzing problem situations.

Thus, if you are running the trace for your own purposes, you will probably only be interested in the datastreams, so you should code the I/O operand to limit the output that you will receive.

## THE GLOBAL TRACE REPORT

The trace report shows each trace record in sequence as it occurred. Each entry shows the type of trace record, the length of the buffer image and the actual buffer image, both in hexadecimal and character format. The buffer image prints in coredump format, showing 32 bytes of data, first in hex and then in character, with hex offsets on the left.

The record type identifiers will read as follows:

#### FROM LU BEFORE COMPRESS

This is the buffer image that has been received from the logical unit, prior to inserting any fields which had modified-data-tag set on when the screen was displayed.

#### TO APPL AFTER COMPRESS

This is the buffer image that has been sent to the application, after inserting any fields which had modified-data-tag set on when the screen was displayed.

#### FROM APPL BEFORE COMPRESS

This is the buffer image that has been received from the application, prior to applying any outbound compression logic.

#### TO LU AFTER COMPRESS

This is the buffer image that has been sent to the logical unit, after applying outbound compression logic.

Additional records will be printed on the diagnostic trace with identifiers like VECTOR, OPTION, IMAGE, etc. These are the control blocks used by AUTOMON@-VTAM at the moment that the previous datastream was sent or received.

## LOCAL TRACING - MVS AND DOS/VSE

The local trace feature provides a quick and easy way to capture a series of terminal datastreams or a diagnostic trace. It is available for CICS and TSO only. Local trace does not use a file to record its records. Trace records are kept in memory, chained off of the terminal control block. Storage above the line is used for MVS/XA systems and GETVIS storage is used for DOS/VSE.

You view local trace records by displaying them on a terminal. This can be done only in CICS for DOS/VSE users, and in CICS or TSO for MVS users.

Local trace does not have the MVS restriction that it cannot operate for applications using authorized path. Thus, you can use local trace for TSO and CICS when running HPO.

## USING LOCAL TRACE IN CICS

In CICS, you can trace your own terminal or another user who is logged on to the same CICS system.

## ACTIVATING THE CICS LOCAL TRACE

To activate local trace for a terminal in CICS, issue the following command from a clear terminal screen:

```
VTEX,TRACE[,LU=xxxxxxx][,CT=nnn][,DIAG][,WRAP]
```

The meaning of each of the parameters in the command is as follows:

- |       |  |
|-------|--|
| VTEX  | This is the CICS transaction code.   |
| TRACE | This command activates local trace.  |
| LU=   | Code the LU name of the terminal to be traced if it is different from the one you are operating. It must be a terminal which is logged-on to the same CICS system where your terminal is logged-on.  |
| CT=   | This is the number of transmissions to be traced. Count two for every input or output that occurs on the terminal. The maximum value that can be coded is 255. The default, if CT is omitted, is 10. |

**DIAG** Code the DIAG keyword if you want a diagnostic trace, rather than an I/O trace. Generally speaking, local trace is only used to examine datastreams, so you would not need the DIAG keyword. You may be instructed to use it from a Unicom Systems Inc. technician when working on a problem.

**WRAP** Code the WRAP keyword if you want the local trace to continue after the output count is exhausted. If WRAP is requested, all local trace records will be released each time the count reaches zero. Then the original count is restored and tracing continues.

The WRAP option is intended for use when you have an intermittent problem that will not occur on demand. By turning on local trace with the WRAP option, the operator can continue normally until the problem occurs, then immediately stop. At that time, the last set of trace records available will have captured the datastreams in question.

**[Note]:** The TRACE keyword must appear immediately after the VTEX transaction code. The other keywords can appear in any sequence.

## EXECUTE THE TRANSACTION TO BE DIAGNOSED

If you are tracing a different terminal than your own, proceed now to that terminal and operate the transaction(s) to be traced. Trace records will be captured and saved in memory for each input and each output that is done on the terminal.

If you are tracing your own terminal, press CLEAR first to exit the VTEX transaction. The CLEAR function will be traced, since local trace is already active. Now proceed to operate the transaction to be diagnosed.

Tracing will automatically terminate when the number of input and output transmissions equal the value specified for CT when local trace was activated.

## DISPLAYING THE CICS LOCAL TRACE RECORDS

To display local trace records in CICS, issue the following command from a clear terminal screen:

```
VTEX,TDISP[,LU=xxxxxxx][,CT=nnn][,OF=nnnn][,SC=xxx...]
```

The meaning of each of the parameters in the command is as follows:

**VTEX** This is the CICS transaction code.

**TDISP** This command invokes the local trace display.

**LU=** Code the LU name of the terminal that was traced if it is different from the one you are operating. It must be a terminal which is logged-on to the same CICS system where your terminal is logged-on.

**CT=** This is the record number where you wish to start the display. If omitted, the display will begin with record number 1.

**OF=** This is an offset into a trace record to begin the display. It is coded in hex. Thus, to begin the record display at position 160, enter OF=00A0. When a partial record is displayed, the offset of the next portion is computed and displayed with the OF= keyword on the command line so that pressing ENTER will display the next part of the record.

SC= This keyword provides a method of scanning through one or more records for a hex or character string. Enter the scan data as a series of 1 to 40 characters. For a character scan, you can precede the data with C' and terminate with a quote if desired, or simply enter the scan data with no delimiters. For hex data, precede the scan data with X' and terminate with a quote. You must enter an even number of hex digits. For more information on the scan operation, see *THE LOCAL TRACE RECORD DISPLAY*, later in this section.

You can now browse forward and backward through the trace records as described later under *THE LOCAL TRACE RECORD DISPLAY*.

## PRINTING THE CICS LOCAL TRACE RECORDS

To print local trace records on a CICS terminal printer, issue the following command from a clear terminal screen:

```
VTEX,TDISP,PRT=xxxx,LU=xxxxxxxx[,F=nnn][,TO=nnn]
[,COMmon][,IMage][,OPTion][,REquest][,RPI][,TCb][,TRacetbl] [,VEctor]
```

The meaning of each of the parameters in the command is as follows:

- VTEX This is the CICS transaction code.
- TDISP This command invokes the local trace display.
- PRT= Code the name of the CICS printer to which this trace is to be routed. It must be a printer that is logged-on to the same CICS system where your terminal is logged-on.
- LU= Code the LU name of the terminal that was traced. It must be a terminal that is logged-on to the same CICS system where your terminal is logged-on.
- From= This keyword provides the starting record number on which printing will begin. If this keyword is omitted, printing will begin at the first record.
- TO= This keyword provides the last record number to be printed. If this keyword is omitted, printing will terminate at the last record.

The remaining keywords apply to pertinent AUTOMON®-VTAM control blocks, tables and buffers. These keywords are only valid if the local trace was activated using the DIAG keyword and should only be used upon request by a Unicom Systems Inc. Technical Representative.

The printed report will appear in the same format as the on-line trace record display. For more information, please refer to *THE LOCAL TRACE RECORD DISPLAY*, later in this section.

**[Note]:** Upon printing, the trace records will not be automatically purged. They must be deleted with the PURGE command.

## DELETING THE CICS LOCAL TRACE RECORDS

When you exit the local trace display by pressing PF3, you will be prompted as to whether you want to purge the local trace records or not. If you answer 'Y', the trace records will be deleted.

You can also delete local trace records by issuing the following command from a clear terminal screen:

```
VTEX,PURGE[,LU=xxxxxxxx]
```

The meaning of each of the parameters in the command is as follows:

- VTEX        This is the CICS transaction code.
- PURGE      This command invokes the local trace purge.
- LU=        Code the LU name of the terminal that was traced if it is different from the one you are operating. It must be a terminal which is logged-on to the same CICS system where your terminal is logged-on.

The PURGE command will proceed through all local trace records for this terminal and free the storage in use.

Local trace records will be automatically purged if the user logs off of CICS at the terminal where tracing was done.

## USING LOCAL TRACE IN TSO

In TSO, you can only trace your own terminal. You cannot trace other TSO users.

## ACTIVATING LOCAL TRACE FOR TSO

To activate local trace for a terminal in TSO, invoke the VTEXPRES command processor by entering VTEXPRES from the command panel or entering TSO VTEXPRES from the command area of any panel. When the TSO statistics display appears enter the following command at the command input area (at the ==> indicator):

```
TRACE[,CT=nnn][,DIAG][,WRAP]
```

The meaning of each of the parameters in the command is as follows:

- TRACE      This command activates local trace.
- CT=        This is the number of transmissions to be traced. Count one for every input or output that occurs on the terminal. Local trace automatically deactivates when the count is reached. The maximum value that can be coded is 255. The default, if CT is omitted, is 10. For TSO, you generally need a larger count than for CICS in order to capture the desired datastream. It will take one or two transactions to exit VTEXPRES and get to the screen to be diagnosed.
- DIAG      Code the DIAG keyword if you want a diagnostic trace, rather than an I/O trace. Generally speaking, local trace is only used to examine datastreams, so you would not need the DIAG keyword. You may be instructed to use it from a Unicom Systems Inc. technician when working on a problem.
- WRAP      Code the WRAP keyword if you want the local trace to continue after the output count is exhausted. If WRAP is requested, all local trace records will be released each time the count reaches zero. Then the original count is restored and tracing continues.

The WRAP option is intended for use when you have an intermittent problem that will not occur on demand. By turning on local trace with the WRAP option, the operator can continue normally until the problem occurs, then immediately stop. At that time, the last set of trace records available will have captured the datastreams in question.

**[Note]:** The TRACE keyword must appear first. The other keywords can appear in any sequence.

## PERFORM THE FUNCTION TO BE DIAGNOSED

Press CLEAR or PF3 first to exit the TRACE command. Now press PF3 again to exit the VTEXPRES program. These exit functions will be traced, since local trace is already active. Now proceed to perform the function in TSO to be diagnosed.

Tracing will automatically terminate when the number of input and output transmissions equal the value specified for CT when local trace was activated.

## DISPLAYING THE TSO LOCAL TRACE RECORDS

To display local trace records in TSO, invoke the VTEXPRES command processor again, then issue the following command at the command area:

```
TDISP[,CT=nnn][,OF=xxx][,SC=xxxxxxxx]
```

The meaning of each of the parameters in the command is as follows:

- TDISP** This command invokes the local trace display.
- CT=** This is now the record number where you wish to start the display. If omitted, the display will begin with record number 1.
- OF=** This is an offset into a trace record to begin the display. It is coded in hex. Thus, to begin the record display at position 160, enter OF=00A0. When a partial record is displayed, the offset of the next portion is computed and displayed with the OF= keyword on the command line so that pressing ENTER will display the next part of the record.
- SC=** This keyword provides a method of scanning through one or more records for a hex or character string. Enter the scan data as a series of 1 to 40 characters. For a character scan, you can precede the data with C' and terminate with a quote if desired, or simply enter the scan data with no delimiters. For hex data, precede the scan data with X' and terminate with a quote. You must enter an even number of hex digits. For more information on the scan operation, see *THE LOCAL TRACE RECORD DISPLAY*, later in this section.

You can now browse forward and backward through the trace records as described later under *THE LOCAL TRACE RECORD DISPLAY*.

## DELETING THE TSO LOCAL TRACE RECORDS

When you exit the local trace display by pressing PF3, you will be prompted as to whether you want to purge the local trace records or not. If you answer 'Y', the trace records will be deleted.

You can also delete local trace records by issuing the following command to the VTEXPRES command processor:

```
PURGE
```

The PURGE command will proceed through all local trace records for this terminal and free the storage in use.

Local trace records will be automatically purged if the user logs off of TSO at the terminal where tracing was done.

## THE LOCAL TRACE RECORD DISPLAY

1). FROM APPL BEFORE COMPRESS

This is the outbound datastream as it was sent from the application before any compression has been done.

2). TO LU AFTER COMPRESS

This is the outbound datastream after AUTOMON®-VTAM has done all compression that can be done to it.

3). FROM LU BEFORE COMPRESS

This is the inbound datastream as it came from the terminal before inserting MDT fields.

4). TO APPL AFTER COMPRESS

This is the inbound datastream that is passed to the application after inserting MDT fields.

If a diagnostic trace was done, there will be other record identifiers for the various control blocks that are captured. Some of these are OPTION, VECTOR, REQUEST, RPL, TCB and IMAGE.

The first record displayed is the record number requested with the CT= keyword of the TDISP command. If CT= was omitted, it is record number one.

## ENTERING COMMANDS AT THE LOCAL TRACE RECORD DISPLAY

The PF key prompt line at the bottom of the screen indicates the available options at this point. They are:

PF1=HELP	Display a help window describing the TDISP command and the local trace display.
PF3=EXIT	Terminate the local trace display. After pressing PF3 you will be prompted whether to purge the local trace records or not.
PF5=RESTART	Restart the display by displaying local trace record number one.
PF7=PREV	Move backward one record.
PF8=NEXT	Move forward one record.

ENTER=REFRESH If the display stops at the end of a trace record, ENTER will simply refresh the same display. If a partial record is displayed, the prompt line shows 'NEXT' beside the ENTER prompt and pressing ENTER will display the next portion of the current record, starting at the location in the record specified by OF= in the command line.

In addition to the PF keys, you can alter the command line in any way desired and press ENTER. You could change the CT= value to skip several records, or enter an OF= value to position into a record, for instance.

You can perform a hex or character scan by keying the SC= keyword on the command line. Enter the scan data following the equal sign as a series of 1 to 40 characters. For a character scan, you can precede the data with C'

and terminate with a quote if desired, or simply enter the scan data with no delimiters. For hex data, precede the scan data with X' and terminate with a quote. You must enter an even number of hex digits.

When the scan data is located in a trace record, that record will display with the line containing the scan data as the first line of the display. To continue the scan from that point, simply press ENTER. When no more records are found which match the scan data, you will receive a message that the end of trace records has been reached.

If you press PF5 (restart) while a scan keyword is active on the command line, the scan will restart with record number 1 and stop at the first record containing the matching data. You can also change the CT= keyword to another record number, either before or after the current record and the scan will continue from that point.

It is also valid to change the major command, TDISP, to one of the other commands and press ENTER. You could change it to PURGE in order to delete this set of trace records, or change it to TRACE in order to activate local trace again. Note that if you activate trace again without purging, the additional trace records will be chained to the end of the first set of trace records.

## CODING AN OUTBOUND USER EXIT PROGRAM

You can code a user exit program to monitor and optionally change all outbound datastreams (those destined for a terminal) that flow through AUTOMON®-VTAM.

To do this, develop your program according to the conventions described below, link it into the same dataset where VTEXPRES resides, then code the module name on the UEXIT keyword of the VAPPL statement or a VLU statement. Coding it on the VAPPL means it will apply to all terminals logged on to the application. A user exit coded on a VLU will override the VAPPL specification for that terminal only. There may be more than one exit program present for different VLU statement.

## PROGRAMMING CONVENTIONS

Upon entry, R1 points to the following parmlist:

Word	Position	Length	-	Contains:
1	+0	(4)	-	Address of uncompressed datastream
2	+4	(4)	-	Address of RPL
3	+8	(4)	-	Address of flag byte with following settings: BIT0 (80) - Not used BIT1 (40) - Not used BIT2 (20) - Not used BIT3 (10) - 1=in SRB mode, 0=in TCB mode BIT4 (08) - 1=no command present (not first RU) BIT5 (04) - Not used BIT6 (02) - Not used BIT7 (01) - Not used
4	+12	(4)	-	Address of 8-byte LU name
5	+16	(4)	-	Datastream length word
6	+20	(4)	-	Address of application ACBname
7	+24	(4)	-	Address of 600-byte dynamic workarea
8	+28	(4)	-	Address of 24-byte permanent savearea
9	+32	(4)	-	Address of datastream copy area

The datastream pointed-to by the first address in the parmlist is always in its complete form, even though the original request may have sent a buffer-list or 3270DS structured field. That is, the first byte of the datastream contains the command code, the second contains the WCC and the outbound data begins at the third position.

The one exception to this is applications that send true multiple request units with separate sends. In this case, the command code in position 1 will only be there if this is the first request unit. The flag byte will contain X'08' if this datastream is not the first request unit. In this case, the first byte is a data byte or an order.

**[Note]:** It is possible for any data field to span across request units. AUTOMON®-VTAM ensures that orders will not span an RU. Thus, an order will always be completely contained in a single RU, although the data following the order could begin with the next RU. There are no rules governing request unit boundaries within a field.

Applications that are known to send true multiple RUs are IMS, TSO and IDMS. CICS does not. If your exit program cannot easily handle RU splits, it may be advantageous to increase the outbound RU size of the application system.

The fifth parameter, the datastream length word, contains the length of the uncompressed datastream upon entry. Upon return, the exit program must place the new length in this field, if it was changed. Note that this field in the parmlist is **not** the address of a length word, it is the length word itself.

The sixth word, the address of the application ACB name, points to an 8-byte name which is the true ACB name of the application system. This is the same as the APPLID coded on the VAPPL statement of the customization table unless a generic APPLID was coded.

The seventh word, address of dynamic workarea, points to 600 bytes that may be used for any purpose in the exit program. The storage is associated with the current VTAM request and is therefore safe to use in a reentrant program. Note that any information stored in this area will not be present upon the next entry to the exit.

The eighth word, address of permanent savearea, points to 24 bytes that is associated with the terminal and will be undisturbed across multiple calls to the exit. The area contains a copy of the VTAM bind image at the time that a log-on occurred. The exit program may store any data in this area and it will remain there until the terminal is logged-off. Or, if it is undisturbed by the exit, the bind image may be referenced for information about the VTAM logmode entry, if desired.

The ninth word, address of datastream copy area, points to an area that is the same size as the uncompressed datastream. It can be used as a copy area if the exit program needs to insert data into the datastream.

The exit program may change the datastream in any way except that it cannot be increased in length by more than 256 bytes.

Registers upon entry to the exit program are:

- R15 - Exit program entrypoint
- R14 - Return address in VTEXMAIN
- R13 - Address of 72-byte register savearea
- R1 - Address of parmlist

All registers must be saved and restored by the exit program.

If more than 600 bytes of dynamic storage is required by the exit program, GETMAINS and FREEMAINS must be performed in the exit. DOS users can do GETVIS and FREEVIS.

The addressing mode is always 31-bit when the exit is called for MVS users. It must be in 31-bit mode on return.

The exit program must not modify any other areas, such as the RPL, the flag byte, etc. The only data fields that can be altered by the exit program are:

- The datastream
- The datastream length word
- The 600-byte workarea
- The 24-byte permanent savearea
- The datastream copy area

If the flag byte contains X'10', then execution is currently in SRB mode rather than TCB mode. When in SRB mode, no SVC instructions can be performed or the application system will abend.

If MVS macros must be performed while in SRB mode, the exit program must use the branch-entry form of the macro. Note that, depending on the macro used, it may be necessary to obtain a lock before issuing the macro.

The exit program must reside in the LPA (MVS) or SVA (DOS) in order to be accessed by VTEXMAIN.

All code in the exit program must be fully reentrant.

Care must be taken in the exit program to never overrun any of the passed areas such as the 600-byte workarea, the 24-byte savearea or the datastream copy area. If the exit program should move anything

beyond the end of these areas it will cause corruption in the AUTOMON@-VTAM control blocks, which will abend the application.

## SAMPLE USER EXIT PROGRAM

This sample program scans each datastream for a predefined key word. If found, it removes the key word from the datastream by shifting the low-order remaining bytes over it and decreasing the datastream length.

This example handles the possibility that the key word may split across an RU in any manner. This makes the exit rather more complex, since, when a partial match of the keyword is found at the end of one RU, that data must be removed on the possibility that a full match will result with the next RU. If this does not occur, however, it is necessary to re-insert the data that was removed at the beginning of the next RU.

Note also that in this example, the keyword must appear as the first data in a field in order to qualify for removal, rather than appearing anywhere within the field.

```

EX21UXIT    EJECT
            CSECT
            USING    *,R12                Base reg addressability
            STM      R14,R12,12(R13)      Save caller regs
            LR       R12,R15              Set base reg
            LR       R8,R1                Save parmlist address
            USING    PARMLIST,R8          Address the parmlist
            L        R4,PDATSADR          R4 = Datastream
            L        R5,PFLAGADR          R5 = Flag byte
            L        R6,PLENGTH           R6 = Datastream length
            L        R7,PPERMAADR        R7 = Perm savearea
            USING    PERMAREA,R7          Address perm savearea
            CLI      PERMFLAG,X'FF'      Is this first call this LU ?
            BE       INIT0100             No, we've been here before
            XC       PERMAREA(24),PERMAREA Clear the area
            MVI      PERMFLAG,X'FF'      Show we've been called
INIT0100    EQU     *
            L        R9,PWORKADR          R9 = 600-Byte workarea
            USING    WORKAREA,R9          Address It
            TM       0(R5),X'08'          Is there a common code ?
            BO       INIT0200             BR if no
            LA       R4,2(R4)             Position R4 to first data byte
            SH       R6,=H'2'             Decrement remaining length
            BNP      RETURN                Exit if no data
INIT0200    EQU     *
            ST       R4,WSTRTADR          Save data start address
            AR       R6,R4                R6 = End of datastream
            L        R1,PRPLADR           Load RPL address
            USING    IFGRPL,R1            Address the RPL
            TM       RPLCHN,RPLHRST+RPLMIDLE+RPLLAST+RPLONLY Any chains?
            BZ       INIT0400             BR if no
            TM       RPLCHN,RPLLAST+RPLONLY Is this the last or only RU ?
            BNZ      INIT0400             BR if yes
            OI       WFLAG,WCHAINED      This is not the last RU
            DROP     R1
INIT0400    EQU     *
            SPACE
*
*
****      Scan datastream for keyword 'SOFTOUCH', at start of field. Handle possibility
*          of RU split in keyword

```

*			
MAINLINE	EQU	*	
MAINTRT	TRT	0(R4),ORDRTABL	Is this an order ?
	BZ	MAIN2000	No, this is 1st data byte of field
MAIN0500	EQU	*	
	LR	R14,R6	R14 = End of data
	SR	R14,R4	R14 = Remaining length
	BNP	RETURN	No more data to scan
	LA	R14,255(R14)	Set up for long TRT
	SRDL	R14,8	
	SRL	R15,24	
MAIN1000	DS	0H	
	XR	R2,R2	
	EX	R15,MAINTRT	Scan for any orders
	BNZ	MAIN1500	BR if found
	LA	R4,1(R15,R4)	Bump to next position
	LA	R15,255	Set max TRT length
	BCT	R14,MAIN1000	Continue search
	B	RETURN	No more orders
	SPACE		
MAIN1500	EQU	*	Datastream order found ...
	B	*(R2)	Branch on TRT table index code
	B	MAINSBA	04 - Process SBA
	B	MAINNL	08 - Process new line
	B	MAINSF	12 - Process start field
	B	MAINIC	16 - Process insert cursor
	B	MAINRA	20 - Process Repeat to Address
	B	MAINPT	24 - Process program tab
	B	MAINEUA	28 - Process erase unprot to adr
	B	MAINSC	32 - Process set char attribute
	B	MAINSFE	36 - Process start field extended
	B	MAINMFE	40 - Process modify field
	B	MAINGE	44 - Process graphics escape
	SPACE		
MAINSBA	EQU	*	SBA found
MAINEUA	EQU	*	EUA found
	LA	R4,3(R1)	Position past order
	B	MAINLINE	Go resume scan
	SPACE		
MAINSC	EQU	*	Set char attribute found
	LA	R4,3(R1)	Position past order
	B	MAIN0500	Resume scan, not a new field
	SPACE		
MAINNL	EQU	*	New line found
MAINPT	EQU	*	Program tab found
	LA	R4,1(R4)	Position past order
	B	MAINLINE	Go resume scan
	SPACE		
MAINIC	EQU	*	Insert cursor found
	LA	R4,1(R4)	Position past order
	B	MAIN0500	Resume scan, not a new field
	SPACE		
MAINSF	EQU	*	Start field found
MAINGE	EQU	*	Graphic escape found
	LA	R4,2(R1)	Position past order
	B	MAINLINE	Go resume scan
	SPACE		
MAINRA	EQU	*	Repeat to addr found
	LA	R4,4(R1)	Position past order

	B	MAINLINE	Go resume scan
	SPACE		
MAINSFE	EQU	*	Start field extended found
MAINMFE	EQU	*	Modify field extended found
	XR	R14,R14	
	IC	R14,1(R1)	Load number of pairs
	SLL	R14,1	Times 2
	LA	R0,2(R14)	R0 = length of order
	AR	R4,R0	Position past order
	B	MAINLINE	Go resume scan
	SPACE		
	*		
	*		
	****	First data byte of a field found ...	
	*		
	*	If this is our keyword, remove it and shorten the datastream by 8 bytes	
	*		
	*		
MAIN2000	EQU	*	
	LR	R15,R6	Get end of datastream
	SR	R15,R4	R15 = Remaining length
	CH	R15,=H'8'	At least 8 bytes left?
	BL	MAIN2400	No, possible keyword split
	XR	R14,R14	
	C	R4,WSTRTADR	Is this the first byte of RU?
	BNE	*+8	No, can't be split keyword
	IC	R14,PKSPLNG	Load split keyword length (if any)
	LA	R1,8	Load max keyword length
	SR	R1,R14	Get remaining keyword length
	BNP	RETURN	Safety check
	LA	R14,PCOMPFLD(R14)	R14 = Next byte of comparable fld
	BCTR	R1,0	Decrement length 1
	MVC	0(0,R14),0(R4)	
	EX	R1,*-6	Move data to compare field
	CLC	PCOMPFLD,=C'SOFTOUCH'	Is this our keyword?
	BE	MAIN2600	BR keyword found
	CH	R1,=H'7'	Was this a split compare?
	BE	MAIN0500	No, resume scan
	*		
	****	No match on split compare. Data was removed from the previous RU and must now be re-inserted.	
	*		
	*		
	LR	R15,R6	Get end of datastream
	SR	R15,R4	R15 = Length to end
	LR	R2,R15	Save length
	LR	R14,R4	Set up for MVCL
	L	R0,PCOPYADR	R0 = Target ADDR
	LR	R1,R15	Length
	MVCL	R0,R14	Move datastream to copy area
	IC	R15,PKSPLNG	R15 = Length of data removed
	SH	R15,=H'1'	Decrement 1
	BM	MAIN0500	Safety check
	MVC	0(0,R4),PCOMPFLD	
	EX	R15,*-6	Move data to startup of RU
	LA	R0,1(R4,R15)	R0 = Target restore address
	LR	R15,R2	Restore datastream length
	LR	R1,R15	*
	L	R14,PCOPYADR	R14 = Copied datastream
	MVCL	R0,R14	Restore remainder of datastream

	LR	R6,R0	Set new end of data address
	SR	R0,R4	Get new total length
	ST	R0,PLENGTH	Store increased datastream length
	B	MAIN0500	Go resume scan
	SPACE		
*			
****			Less than 8 bytes remaining in datastream
*			
*			If this is not the last or only request unit, then if partial keyword match on
*			remaining data, remove the partial keyword and wait for next RU to
*			complete the compare.
*			
MAIN2400	EQU	*	Less than 8 bytes left
	TM	WFLAGWCHAINED	Is this the last RU ?
	BZ	RETURN	Yes, can't be a match
	LR	R1,R15	Get remaining length
	SH	R1,=H'1'	Decrement remaining length
	BM	MAIN0500	Safety check
	CLC	0(0,R4),=C'SOFTOUCH'	
	EX	R1,*-6	Do remaining characters match?
	BNE	MAIN0500	No, resume scan
	MVC	PCOMPFLD(0),0(R4)	
	EX	R1,*-6	Move it to compare field
	LA	R15,1(R1)	Get true length
	STC	R15,PKSPLNG	Save split keyword length
	SPACE		
*			
****			Full or partial keyword match found, remove keyword data and shorten
*			datastream
*			
*			R4 -> Start of data to remove
*			R1 = Length to remove -1
*			
MAIN2600	EQU	*	
	LA	R1,1(R1)	Get true length to remove
	LA	R14,0(R4,R1)	R14 = Next byte after keyword
	LR	R15,R6	Get end of data address
	SR	R15,R14	R15 = Length to shift
	LR	R0,R4	R0 = Target address
	LR	R1,R15	Set length
	MVCL	R0,R14	Shift remaining data over keyword
	LR	R6,R0	Set new end of datastream
	S	R0,PDATSADR	Get new length
	ST	R0,PLENGTH	Store decreased datastream length
	B	MAIN0500	Go resume scan
	SPACE	3	
*			
*			
****			Return to VTEXMAIN
*			
RETURN	EQU	*	
	LM	R14,R12,12(R13)	Restore caller REGS
	BR	R14	
	EJECT		
*			
****			Constants and DSECTS
*			
REPAREA	DS	0D	
	DC	128S(*)	Patch area

```

SPACE
LTORG
SPACE
ORDRTABL DC 256X'00'
          ORG ORDRTABL+X'11' SBA character
          DC X'04'
          ORG ORDRTABL+X'15' NL character
          DC X'08'
          ORG ORDRTABL+X'1D' Start-field character
          DC X'0C'
          ORG ORDRTABL+X'13' Insert-character cursor
          DC X'10'
          ORG ORDRTABL+X'3C' Repeat to ADDR order
          DC X'14'
          ORG ORDRTABL+X'05' Program tab
          DC X'18'
          ORG ORDRTABL+X'12' Erase UNP-to-ADDR
          DC X'1C'
          ORG ORDRTABL+X'28' Set char attribute
          DC X'20'
          ORG ORDRTABL+X'29' Start field extended
          DC X'24'
          ORG ORDRTABL+X'2C' Modify field
          DC X'28'
          ORG ORDRTABL+X'08' Graphic escape
          DC X'2C'
          ORG
          SPACE 3

```

\*

\*

\*\*\*\*\*

#### REGISTER USAGE

\*

R0	EQU	0	Work reg
R1	EQU	1	Work reg, Parmlist at entry
R2	EQU	2	Work reg
R3	EQU	3	Unused
R4	EQU	4	Datastream scan reg
R5	EQU	5	Flag byte address
R6	EQU	6	End of datastream address
R7	EQU	7	Permanent savearea address
R8	EQU	8	VTEXMAIN Parmlist address
R9	EQU	9	600-byte workarea address
R10	EQU	10	Not used
R11	EQU	11	Not used
R12	EQU	12	Base reg
R13	EQU	13	Caller reg. savearea address
R14	EQU	14	Work reg, return ADDR at entry
R15	EQU	15	Work reg, entrypoint at entry

SPACE 2

\*

\*

\*\*\*\*

#### DSECT of passed parameter list from VTEXMAIN

\*

PARMLIST	DSECT		
PDATSADR	DS	F	Address of datastream
PRPLADR	DS	F	Address of the RPL
PFLAGADR	DS	F	Address of flag byte
PFSRB	EQU	X'10'	* In SRB mode if on
PFNOCMD	EQU	X'08'	* No command present if on

PLUNMADR	DS	F	Address of LUNAME
PLENGTH	DS	F	Datastream length
PACBADR	DS	F	Address of ACB name
PWORKADR	DS	F	Address of 600-byte workarea
PPERMAADR	DS	F	Address PF 24-byte perm savearea
PCOPYADR	DS	F	Address of datastream copy area
	SPACE 2		
*			
*			
****	DSECT of permanent savearea for this LU		
*			
PERMAREA	DSECT		
PERMFLAG	DS	X	Flag - X'FF' = Not first call
PKSPLNG	DS	X	Length of saved split keyword
PCOMPFLD	DS	CL8	Split keyword savearea
	SPACE 2		
*			
*			
****	DSECT of dynamic workarea for this request		
*			
WORKAREA	DSECT		
WSTRTADR	DS	F	Address of first data byte in RU
WFLAG	DS	X	Flag byte
WCHAINED	EQU	X'80	* Chained RU, This is not last
	SPACE 2		
	IFGRPL AM=VTAM		DSECT of RPL
	END		

## PROBLEM ANALYSIS AND DIAGNOSIS

If you experience any problems with AUTOMON®-VTAM, they will usually fall into one of these categories:

- 1). AUTOMON®-VTAM activates properly but the statistics displays show no compression statistics.
- 2). An application does not display correctly on the screen. Fields are missing or extra data is present which should not be there.
- 3). A transaction abend occurs when certain applications are used with compression active.
- 4). An ABEND occurs in one of the application regions which brings down the region/partition.
- 5). Poor compression results for a particular application.
- 6). VTAM applications will not accept log-ons or will not initialize properly.

The first condition is a start-up problem. It is almost always caused by an incorrect procedure or coding errors in the customization table. See the checklist, below, titled *NO COMPRESSION STATISTICS, PROCEDURAL CHECKLIST*. It is usually one of these conditions that have caused the problem.

The second condition is the most common operational problem that users experience. There are so many options in 3270 datastream handling and so many different ways to accomplish a display that it is possible that you could have a transaction that is doing something that AUTOMON®-VTAM cannot yet handle correctly. Refer to *ERRONEOUS SCREEN DISPLAYS*, later in this section.

The third condition happens very rarely. It can happen if AUTOMON®-VTAM fails to correctly construct the input datastream (using inbound compression) and passes something to the application that it cannot handle. Refer to *TRANSACTION ABENDS* later in this section.

The fourth condition, bringing down an application region is, of course, the most severe. The current stability level of AUTOMON®-VTAM should preclude this from happening in the three major VTAM applications (CICS, TSO, IMS) and it is operating successfully in many other VTAM applications at this time. However, you could have a VTAM application that we have not yet experienced which might produce this condition. Refer to *SYSTEM ABENDS*, later in this section.

The fifth condition, poor compression results, can have many causes, as discussed under *POOR COMPRESSION RESULTS*, later in this section.

If you are experiencing the sixth condition, log-ons not accepted, refer to the discussion entitled *ORDERING THE LPA*, in the *INSTALLATION* section for users.

Let's hope that you have none of these problems, but if you do, that's what technical support reps are for.

## NO COMPRESSION STATISTICS, PROCEDURAL CHECKLIST

If you have activated AUTOMON®-VTAM but the statistics displays show no compression statistics accumulating, the following list of conditions are situations that different users have experienced that have caused this problem:

### **If you are getting no statistics in TSO:**

If any edit errors occurred that pertain to the VSTART, VAPPL or VLU statements it is possible that the customization table was built incorrectly, which could prevent proper operation.

AUTOMON®-VTAM uses the ACBNAME of the APPL statement defined to VTAM when matching against the VAPPL statements to determine if an application is included. Nearly all installations use 'TSO' as the first three characters of the ACB name. The VAPPL statement for TSO must be generic in order to include all TSO users. That is why you should only code three characters of the ACB name.

If this is true, verify for certain that the label of the VLU statement has the correct LU name. The TSO statistics display which first appears when you invoke the VTEXPRES command processor displays the LU name of the terminal. Check that against your VLU statement. If there is any doubt, try changing the VSTART to LU=INCL, re-activate AUTOMON®-VTAM, then log off and back on.

The \$DEFAULT VAPPL statement signals the end of VAPPL statements. If it does not appear last, subsequent VAPPLs will be ignored.

If you are using the logon-delay feature, be sure that you press ENTER enough times to exhaust the delay count. If you have done that and still get no results, try removing the delay keyword, re-activate compression, then log off and back on. If it works now, you should not use the logon-delay feature. Logon delay requires that the application do an ERASE command before it will allow compression to start after exhausting the count. Some applications never do an ERASE, or do so very infrequently.

If all of the conditions above check out correctly and you still are unable to get any statistics in TSO, notify Unicom Systems Inc. Systems technical support.

### **If you are getting no statistics in CICS:**

If any edit errors occurred that pertain to the VSTART, VAPPL or VLU statements it is possible that the customization table was built incorrectly, which could prevent proper operation.



If compression is activated while an application system is up and running, and you log on to the application system, if you are the only one to log onto that system, compression will not begin until you log off and back on. In other words, you would have to log on twice for compression to start. This condition will occur **only** if you are the single user on the application system.

**If you are getting no statistics in other applications:**

Review the conditions listed for CICS, above. All of the same conditions could cause no statistics in other applications except for the last condition, failure to code KEY=8.

## DIAGNOSTIC DOCUMENTATION ACCORDING TO PROBLEM TYPE

Depending on the type of problem that you are having, the type of diagnostic documentation that you will need to produce for the support rep will vary. You should always discuss the problem with the support rep first (there may be an available PTF) and then obtain the documentation that he requests.

More is better in all cases. That is, the more documentation you can send to the support rep, the better chance there is of a speedy resolution to the problem.

## ERRONEOUS SCREEN DISPLAYS

For the type 2 problem, incorrect screen displays, consider the following:

- 1). If you are getting the BUFFER OVERRUN message from AUTOMON®-VTAM, you need to increase you input terminal I/O area sizes as described in *INSTALLATION*, section 02 under *MINIMUM TERMINAL INPUT SIZES*.
- 2). Try changing some of the customization options to try to isolate the problem. First, change the NDFSUP option to 'N' and try the application. If it still fails, change EXTATTR to LEAVE. You could also try turning off RBGEN. A final test would be to change EWIMG to 'F' if it is currently 'B', or vice-versa.

If one of these changes clears up the problem, report that to your support rep at Unicom Systems Inc.Systems.

- 3). There are three procedural conditions that can cause erroneous screen displays which can only be resolved by changing the way you operate the terminal.
  - The first of these occurs in text editors, including TSO. If the text lines contain nulls after the data to the end of the line, the following procedure can cause an error:

The operator moves the cursor past the end of the data using the cursor movement keys, rather than the space bar, then types some data. Then, instead of just pressing ENTER, some command such as DOWN or UP is issued which shifts the screen display. The result, sometimes, is that the data which was keyed at the end of the line stays on the screen, even through subsequent paging forward, etc.

The text member is correctly updated. If you refresh the screen by clearing or pressing PA2 (in TSO), the extra data disappears and the display will be correct.

This problem is caused by the nulls on the text lines. Because of the nature of 3270 terminals, where nulls are never sent on and inbound transmission, AUTOMON®-VTAM cannot recognize that the data was keyed at the end of the line and therefore does not know to clear the end of the line.

The problem will not occur if:

- (a) Nulls are not used.
  - (b) The update is made in place, rather than scrolling or paging at the same time.
  - (c) The space bar is used to move the cursor.
- The second procedural problem involves the ERASE INPUT key. If the operator presses ERASE INPUT at a screen display, then keys some data and presses ENTER or a PF key, the application program will sometimes respond incorrectly.

This problem occurs, again, because there is no indication given by the hardware that the ERASE INPUT key was pressed. When AUTOMON®-VTAM receives the input, it does not know that the MDT flags were cleared at the terminal via the ERASE INPUT. Therefore, it will insert into the datastream any preset MDT fields, just like any other input. If the application is not expecting this, it may respond incorrectly.

Note that there is no problem with the ERASE EOF key. This condition only involves the ERASE INPUT key, which is very seldom used in most environments.

This problem will not occur if:

- (a) ERASE INPUT is not used.
  - (b) Inbound compression is turned off.
- The third procedural problem involves the use of PA keys in conjunction with the RBGEN option.

Read-buffer commands often occur in help products, or session managers, in response to a PF or PA key used as a 'hot' key. If you specify the RBGEN option as 'Y' so that AUTOMON®-VTAM will generate the read-buffer datastream, and if you are using a PA key as your 'hot' key, it is not possible for AUTOMON®-VTAM to determine the cursor position when the PA key is pressed.

This occurs because the cursor position is not transmitted in when a PA key is pressed. It is transmitted when a read-buffer command is executed, which is the method used by the application doing the read-buffer to determine the cursor position. Since the RBGEN=Y parameter causes the read-buffer command to be suppressed, the cursor position is never transmitted back.

AUTOMON®-VTAM attempts to keep track of the cursor position of the last output screen and will use that as a cursor position on the generated read-buffer datastream. This is often incorrect, however.

This problem is particularly noticeable with help products, where cursor-sensitive fields have on-line help displays which are associated with them. In other cases, an incorrect cursor position is often not important.

This problem will not occur if:

- (a) RBGEN is set to 'N'.
- (b) PA keys are not used as 'hot' keys for applications which do a read-buffer..

**[Note]:** The three procedural problems listed here are common to all terminal datastream compression products. If you have an application for which the procedure cannot be changed and one of these problems is hampering operation, please discuss it with a support rep from Unicom Systems Inc. Systems. There may some installation-specific solution that can be derived.

If the problem cannot be resolved, the following information should be obtained:

- 1). A screen print of the display, both when it fails and when it works correctly (with EXPRESS inactive).
- 2). A AUTOMON@-VTAM diagnostic trace of the transaction. For MVS users who are running authorized path you can either bring up a test version of the application which does not use authorized path in order to run the trace, or you can use the SLIP trace, described later in this section. It may also be possible to run a local trace and obtain screen prints of the trace records.
- 3). A VTAM buffer trace, both when it fails and when it works correctly.

**[Note]:** Be sure that you carefully identify the output as to which is the one that fails and which is the one that works. It's rarely obvious to the support rep, who is not familiar with your display.

## TRANSACTION ABENDS

For the type 3 problem, an ABEND in a user transaction, the following information should be obtained:

- 1). A screen print of the display when it works correctly (with EXPRESS inactive).
- 2). A AUTOMON@-VTAM diagnostic trace of the transaction. For MVS users who are running authorized path you can either bring up a test version of the application which does not use authorized path in order to run the trace, or you can use the SLIP trace, described later in this section. It may also be possible to run a local trace and obtain screen prints of the trace records.
- 3). A VTAM buffer trace, both when it fails and when it works correctly.
- 4). The core dump produced by the transaction ABEND.

## SYSTEM ABENDS

For the type 4 problem, an ABEND which brings down the region or partition, if the ABEND is a SOC4, try coding KEY=8 on the VAPPL statement. If this does not resolve the problem or if it is a different type of ABEND, obtain the following information:

- 1). The core dump produced when the ABEND occurred.
- 2). In many cases, the dump produced when the ABEND occurs is inadequate. The support rep will need to see all of the storage in the partition or region, if possible. For CICS, this may involve setting the dump option to PARTITION, so that the entire partition is dumped rather than the formatted snap dump which is normally produced.
- 3). For MVS users, the preferred dump is an SVC dump. If that cannot be obtained, a SYSABEND dump is the next choice.

## POOR COMPRESSION RESULTS

If you are experiencing poor compression results in IMS, the first thing you should try is increasing the outbound RU size. Please read the discussion in the MVS installation section entitled *SNA RU SIZE CONSIDERATIONS* for more information about this.

If you are experiencing compression results of less than 40 percent in any application other than IMS and TSO (TSO will normally get from 15 to 25 percent) it probably means that your applications are doing something that is causing EXPRESS to not be able to perform its outbound image saving properly.

It is possible that your applications are very efficient in their native state, but this is rarely the case.

Try changing the compression method from field-level to byte-level or vice-versa to see if you get better results. This is accomplished with EWIMG=B for byte-level and EWIMG=F for field level.

If you think the application uses extended attributes, make sure that you have EXTATTR=CMPRS specified. For a test, you might try EXTATTR=STRIP, which removes extended attributes altogether. If that makes a significant improvement, you could weigh the value of keeping the extended attribute displays versus the improved compression performance without them.

Finally, if you are running an application such as a help package or electronic mail package which frequently does read-buffer commands, be sure that you specify RBGEN=Y. This will cause AUTOMON®-VTAM to generate the read-buffer datastream from the terminal image, saving the transmission of the full terminal buffer over the line.

The situation can probably be improved upon by producing a AUTOMON®-VTAM I/O trace with a typical sampling of your transaction mix and letting a support rep analyze the data. A buffer trace will not suffice for this since it does not dump the entire screen buffer.

Still another method is to capture a set of actual datastreams and send them to the support rep. This can be done with the AUTOMON®-VTAM global I/O trace if the application is not running authorized path. IMS has its own trace facility which makes this possible, or you may have some utility or system function that can capture datastreams (at least the outbound datastreams) and write them to a sequential file. If this can be done, the support rep can drive your datastreams through AUTOMON®-VTAM at his site and determine why better results are not happening.

One reason for poor compression results is frequent screen changes. That is, each time the operator responds to a screen, an entirely different screen display appears. In this scenario, image saving does not accomplish much, since there is little duplication of data sent from screen to screen. Byte level image compression (EWIMG=B) will generally work the best in this case. If you have this situation, it may be that there is little that can be done to improve your statistics. However, it still may be worth your time to capture a set of datastreams and send them to the support rep for analysis.

## BYPASSING THE PROBLEM

If you know what transaction is causing the problem to occur, which would be the case for problem types 2 and 3 and sometimes 4, you can usually bypass the problem by using the VSKIP statement to tell AUTOMON®-VTAM to exclude this particular datastream from compression.

You should first produce all requested documentation on the problem and send it to the support rep so that it can be resolved, then code a VSKIP statement in the customization table to identify the screen display and exclude it. This will allow you to continue operating with EXPRESS until a resolution is forthcoming.

For more information on VSKIP, see *VSKIP - IDENTIFY DATASTREAMS TO BE EXCLUDED* in section 03 - *CUSTOMIZATION*.

## RUNNING THE SLIP TRACE

MVS users whose application runs authorized path and cannot produce a AUTOMON®-VTAM diagnostic trace can use a GTF SLIP trace to produce similar diagnostic documentation.

The SLIP trace will capture the datastream, both before and after compression, and will dump some of the control blocks present at the time. It is usually enough information to resolve problem type 2 and is always helpful if you know what transaction is causing the failure to occur.

To activate the SLIP trace, you must set a SLIP trap, as follows:

- 1). First bring up GTF.
- 2). Enter the TRACE command for GTF as follows:

```
TRACE=SLIP
```

You must reply 'U' to GTF after this, to complete the GTF initialization.

- 3). Now enter the SLIP command to set the trap location. The SLIP command is entered at the system log as a stand-alone command. Enter as follows:

```
SL SET,IF,ID=VTEX,A=TRACE,J=jobname,
LPAMOD=(VTEXMAIN,10),TD=(STD,REGS,1R?,15R?),END
```

In the SLIP command, 'jobname' is the JOB name of the application system. If you are tracing in TSO, this would be your TSO user ID, as displayed in the JOBNAME column of the SDSF 'DA' display. The '10' in the LPAMOD operand is the program location in the VTEXMAIN program where the trap is to be set. This location could change with different releases but is correct for release 2.0.

- 4). Now you must supply a VTRACE command to AUTOMON®-VTAM for the LU to be traced. Code the VTRACE in the customization table as documented in section 03 - *CUSTOMIZATION*. Then reactivate compression in order to load the new table. You do not need a trace file, nor do the TRACDDN or TRACDSN keywords need to be present on the VAPPL.

Tracing will begin when both the SLIP command and the activation with VTRACE are completed. You may do these in either sequence. You should see the message "SLIP TRACE ID=VTEX MATCHED" appear on the system log when tracing begins.

- 5). Now the SLIP is set, so perform the failing transaction on the LU designated with the VTRACE statement, to the point where the erroneous screen display or abend occurs.
- 6). Now you must issue a 'P' command to stop GTF. Then you can print the slip trace. The following JCL can be used to print it:

```
//STEP1                EXEC  PGM=IKJEFT01,PARM=AMDPRDMP
//SYSTSIN              DD    DUMMY,DCB=(RECFM=F,
//                      //    LRECL=80,BLKSIZE=80)
//SYSTSPRT             DD    DUMMY
//SYSPRINT             DD    SYSOUT=*
//DISK                 DD    DSN=GTF.TRACE.FILE,           [Note 1]
//                      //    DISP=OLD,UNIT=SYSDA
//PRINTER              DD    SYSOUT=*
//SYSIN                DD    *
                       TITLE AUTOMON®-VTAM SLIP TRACE      [Note 2]
                       E DD=DISK,SLIP                      [Note 3]
                       END
/*
```

**[Notes]:**

1. This is the GTF file as specified in your GTF initialization PROC.
  2. It is helpful to use a meaningful title for the problem you are tracing.
  3. This command is required.
- 7). To delete the slip trap, enter the command:

```
SL DEL,ID=VTEX
```

[Note]: The SLIP command is documented in *MVS SYSTEM COMMANDS*.

## SUMMARY

If you experience any sort of problem with AUTOMON®-VTAM, it can be resolved if adequate documentation can be produced on the problem.

If you are unfamiliar with any of the diagnostic tools referenced in the section, such as SVC dumps, buffer traces, GTF, SLIP traps, etc, please feel free to ask your support rep for instructions. He can tell you the specific instructions to perform as well as direct you to the IBM documentation describing that function.

## USING THE MEMORY DISPLAY/ALTER UTILITY

The first step of the installation procedure loaded a utility program, STSCORE, which is a memory display/alter utility. You should have received a separate document which describes the operation of this utility, which runs in CICS.

There is also a version of STSCORE that runs in TSO. It is invoked by entering 'VTEXCORE' from the TSO command panel. This version operates somewhat differently than the CICS version as is explained in detail in the STSCORE manual.

For AUTOMON®-VTAM, there are special functions which can be performed with STSCORE that are not documented in the other manual. This is because they only pertain to AUTOMON®-VTAM users, whereas STSCORE is generally distributed with all products.

Generally speaking, you would have no need of these functions unless you are working with a Unicom Systems Inc. technician to analyze a problem situation. Nevertheless, they are documented here for your information.

## SPECIAL COMMANDS UNIQUE TO AUTOMON®-VTAM

There are a series of command keywords which allow quick access to the AUTOMON®-VTAM programs and control blocks. These are:

- VMAIN - The VMAIN command takes you directly to the beginning of the VTEXMAIN program in the LPA or SVA. You may browse forward from there or append the command with offsets as desired.
- VOPT - The VOPT command displays the beginning of the option table (customization table) control block. You may browse forward from there or append the command with offsets as desired.
- VECT - The VECT command displays the beginning of the VECTOR control block. You may browse forward from there or append the command with offsets as desired.
- VAPPL - The VAPPL command by itself displays the first application entry in the option table control block. You can also enter VAPPL=xxxxxxx, where the xxxxxxxx is one of the APPLIDs of your VAPPL statements. This will go directly to that entry. If you enter VAPPL=\*, the application entry of the CICS system where you are logged-on will be displayed.

Once positioned to a VAPPL entry, the NEXT and PREVIOUS commands (PF8 and PF7) will proceed from one VAPPL to another.

VLU - The VLU command by itself displays the first VLU entry in the option table control block. You can also enter VLU=xxxxxxx, where the xxxxxxx is one of the LU names of your VLU statements. This will go directly to that entry. If you enter VLU=\*, the VLU entry of your terminal (if present) will be displayed.

Once positioned to a VLU entry, the NEXT and PREVIOUS commands (PF8 and PF7) will proceed from one VLU to another.

LU - The LU command by itself displays the first terminal control block in this CICS application. You can also enter LU=xxxxxxx, where the xxxxxxx is any LU name active in this CICS system. This will go directly to that terminal control block. If you enter LU=\*, the terminal control block of your terminal will be displayed.

Once positioned to a terminal control block, the NEXT and PREVIOUS commands (PF8 and PF7) will proceed from one terminal control block to another.

## ALTERING MEMORY IN THE LPA OR SVA

The STSCORE utility allows memory in the LPA (MVS) or SVA (DOS) to be altered, with proper security authorization. The alter function works exactly as in altering memory in any other location, that is, you simply over-type the characters to be changed on either the hex or character side.

You must supply the proper password to be able to alter memory in the LPA/SVA.

For MVS users, the type-3 SVC that was installed with AUTOMON®-VTAM is used during the alter of LPA storage. If the SVC has not been installed, the operation will not be allowed.

It is sometimes very expedient to apply a PTF to VTEXMAIN by altering the memory directly in the LPA or SVA, rather than going through the process of reloading the module, which often requires an IPL. If the PTF is very small, affecting only a few bytes at one location, this is generally safe, as long as you are careful. If you choose to do this for a larger PTF, do not attempt it without the assistance of a support representative from Unicom Systems Inc. Systems. If a PTF is applied in the wrong sequence it could very well bring down your entire network.

## MESSAGES

All messages in AUTOMON®-VTAM are preceded by a 5-character code which designates the program module producing the message and a 3-digit message number. Messages will normally appear on the system log, although some can appear on the terminal when running VTEXPRES as a TSO command, or while in the VTEX transaction.

The messages are presented here in alphabetic order with an explanation of each message and the action that can be taken.

### VAPDS001. SYSTEM COMPRESSION STOPPED

This message indicates successful deactivation of compression for the system in response to a SYSTEM STOP command.

ACTION: This is an informational message only. No action is required.

### VAPDS002. MUST USE VTEXPRES TO START SYSTEM

This message indicates that compression has not yet been activated by an ACTIVATE command to VTEXPRES. Until that is done, you cannot use the START/STOP SYSTEM function of the application display.

ACTION: Activate compression with VTEXPRES. For more information see the *OPERATION* section for DOS/VSE or MVS.

### VAPDS003. SYSTEM COMPRESSION STARTED

This message indicates successful activation of compression for the system in response to a SYSTEM START command.

ACTION: This is an informational message only. No action is required.

### VAPDS004. COMPRESSION STOPPED FOR APPLICATION

This message indicates successful deactivation of compression for the application in response to an APPL STOP command.

ACTION: This is an informational message only. No action is required.

### VAPDS005. COMPRESSION STARTED FOR APPLICATION

This message indicates successful activation of compression for the application in response to an APPL START command.

ACTION: This is an informational message only. No action is required.

### VAPDS006. SYSTEM COMPRESSION IS CURRENTLY STOPPED

This message indicates that an APPL START or LU START command was issued; however, compression could not be activated for the application or LU because compression for the system is not active. For compression to be active on any level, compression must also be active on all hierarchical levels above that level.

ACTION: You must activate compression at a higher level, either the system or the application.

### VAPDS007. COMPRESSION STOPPED FOR xxxxxxxx IN yyyyyyyy

This message indicates successful deactivation of compression for the terminal in response to an LU STOP command. The xxxxxxxx in the message is the LU name, the yyyyyyyy is the application ID.

ACTION: This is an informational message only. No action is required.

VAPDS008. COMPRESSION STARTED FOR xxxxxxxx IN yyyyyyyy

This message indicates successful activation of compression for the terminal in response to an LU START command. The xxxxxxxx in the message is the LU name, the yyyyyyyy is the application ID.

ACTION: This is an informational message only. No action is required.

VAPDS009. APPLICATION COMPRESSION IS CURRENTLY STOPPED

This message indicates that an LU START command was issued; however, compression could not be activated for the LU because compression for the application is not active. For compression to be active on an LU, compression must also be active on all hierarchical levels above the LU.

ACTION: Activate compression for the application first; then, if necessary, activate compression for the LU.

VAPDS010. 'CICS' IS ONLY VALID FOR SCRSIZE

A request was issued to alter the SCRSIZE operand on the APPLICATION STATISTICS screen with an invalid entry. This operand may only be coded as 'CICS' or <spaces>. For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS011. INVALID SCREEN SIZE

A request was issued to alter the PSIZE or ASIZE operand on the APPLICATION STATISTICS screen with an invalid entry. The value entered was less than 1920. Any value from 1920 to 9999 may be entered. The standard screen sizes are:

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

For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS012. NON-NUMERIC VALUE FOR PSIZE, ASIZE OR DELAY

A request was issued to alter the PSIZE, ASIZE or DELAY operand on the APPLICATION STATISTICS screen with an entry that did not consist entirely of numbers.

For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS013. INCORRECT OPTION FOR EWIMG - Y, N, B OR F

A request was issued to alter the EWIMG operand on the APPLICATION STATISTICS screen with an invalid entry. Valid entries are: 'Y', 'N', 'B' and 'F'.

For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS014. INCORRECT OPTION FOR EXTATTR - C, S OR L

A request was issued to alter the EXTATTR operand on the APPLICATION STATISTICS screen with an invalid entry. Valid entries are: 'C', 'S' and 'L'.

For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS015. INVALID ENTRY, MUST BE Y OR N

A request was issued to alter an operand on the APPLICATION STATISTICS screen with an entry other than 'Y' or 'N' for a field that can only be coded with a 'Y' or 'N'.

For more information, refer to section 03 - *CUSTOMIZATION*.

ACTION: Re-enter with a valid entry.

VAPDS016. OPTION CHANGES SUCCESSFULLY COMPLETED

This message indicates a successful change in the user options and that no entry errors were detected.

ACTION: This is an informational message only. No action is required.

VAPDS017. CANNOT ACTIVATE, LU IS EXCLUDED

A request was issued to activate compression for an LU that has been excluded in the AUTOMON@-VTAM customization table.

**ACTION:** You cannot activate compression for this LU unless you remove the VLU statement which excludes this LU (or change the LU=EXCL on the VSTART to LU=INCL). Then the table must be reloaded into EXPRESS with the ACTIVATE command. Upon doing that, you can activate this LU by logging off and back on.

#### VAPDS018. SYNTAX ERROR IN COMMAND

The input to the VTEX transaction could not be recognized as a valid command.

**ACTION:** Re-enter with valid input.

#### VAPDS019. INVALID KEYWORD IN COMMAND

One of the local trace commands TRACE, TDISP or PURGE was entered and an accompanying keyword could not be identified. Valid keywords are LU=, CT=, OF=, and DIAG.

**ACTION:** Re-enter with valid input. For more information refer to section 07 - *USING THE VTAM TRACE FEATURE*.

#### VAPDS020. DATA FOR LU= OR CT= IS TOO LONG

One of the local trace commands TRACE, TDISP or PURGE was entered and an accompanying LU or CT keyword contained an operand that was too long. The maximum number of characters that can follow the equal sign is 8.

**ACTION:** Re-enter with valid input. For more information refer to section 07 - *USING THE VTAM TRACE FEATURE*.

#### VAPDS021. DATA FOR CT= IS NOT NUMERIC

One of the local trace commands TRACE or TDISP was entered and the data following the equals sign for the accompanying CT keyword was not all numbers. CT must be a value from 1 to 255.

**ACTION:** Re-enter with valid input. For more information refer to section 07 - *USING THE VTAM TRACE FEATURE*.

#### VAPDS022. DATA FOR CT= MUST BE FROM 1 TO 255

One of the local trace commands TRACE or TDISP was entered and the data following the equals sign for the accompanying CT keyword was either zero or exceeded 255. CT must be a value from 1 to 255.

**ACTION:** Re-enter with valid input. For more information refer to section 07 - *USING THE VTAM TRACE FEATURE*.

#### VAPDS023. LOCAL TRACE ACTIVATED FOR xxxxxxxx

This message indicates successful activation of the local trace function.

**ACTION:** No action is required. You can now press PF3 or CLEAR to exit the TRACE command function.

#### VAPDS024. TERMINAL xxxxxxxx IS CURRENTLY INACTIVE

A request was issued to activate local trace for a terminal which is currently not active in compression. The terminal may be quiesced or excluded. The xxxxxxxx in the message is the LU name.

**ACTION:** You must activate the terminal before you can start local trace. Try the LU START function of the application display.

VAPDS025. LOCAL TRACE PURGED FOR xxxxxxxx

This message indicates successful completion of a PURGE command to delete the local trace records for a terminal. The xxxxxxxx in the message is the LU name.

ACTION: No action is necessary. You can now press PF3 or CLEAR to exit the PURGE command function.

VAPDS026. SCAN MUST BE EVEN NUMBER OF HEX DIGITS

The SC= keyword was entered on a TDISP command followed by X', indicating that the scan data was in hex. When a hex scan is done, the number of hex digits entered must be an even number.

ACTION: Re-enter the SC= keyword with an even number of digits.

VAPDS027. SCAN DATA IS NOT VALID HEX DIGITS

The SC= keyword was entered on a TDISP command followed by X', indicating that the scan data was in hex. The data following the quote was not valid hexadecimal digits.

ACTION: Re-enter the SC= keyword with all valid hex digits.

VAPDS028. APPLICATION STATISTICS RESET TO ZERO

This message indicates that the application statistics were successfully reset to zero in response to the reset command.

ACTION: This is an informational message only. No action is required.

VAPDS029. CANNOT ACTIVATE, APPLICATION IS EXCLUDED

A request was issued to activate compression for an application that has been excluded in the AUTOMON®-VTAM customization table.

ACTION: You cannot activate this application without changing the customization table to include the application, then reloading the table with an ACTIVATE command. It will probably be necessary to recycle the application in order to start compression.

VCICS001. UNABLE TO LOCATE SVC IN LPA

When attempting to activate compression, the type-3 user SVC could not be located in the system SVC table. Either the SVC was not installed in this system or an IPL has not been done since installing it.

ACTION: Review the installation instructions and determine whether the SVC was properly installed. If necessary, perform an IPL to activate the SVC.

VCICS002. VECTOR PARTIAL PROCESS

This is a start-up message from the VTEXCICS program used with CICS when running with the HPO option.

ACTION: No action is required. This is an informational message.

VCICS003. VECTOR PROCESSED

This is a start-up message from the VTEXCICS program used with CICS when running with the HPO option.

ACTION: No action is required. This is an informational message.

#### VCICS004. VECTOR INITIAL SETUP

This is a start-up message from the VTEXCICS program used with CICS when running with the HPO option.

ACTION: No action is required. This is an informational message.

#### VEXDS001. AUTOMON@-VTAM IS NOT ACTIVE

A request was issued to display compression statistics, but AUTOMON@-VTAM has not yet been activated.

ACTION: You must first activate AUTOMON@-VTAM by running the VTEXPRES jobstream with PARM='ACT' (or for MVS you may use PARM='DSN'. For more information, refer to *VTEXPRES PARM STATEMENTS* in the *OPERATION* section for MVS or DOS/VSE.

#### VEXDS002. NO APPLICATION ENTRIES PRESENT

No application entries could be found in the customization table.

ACTION: This is a "should never occur" error. If you receive it, notify Unicom Systems Inc. Systems Technical Support.

#### VEXDS003. REQUESTED APPLICATION NOT FOUND

An attempt to access statistics for an application failed because a VAPPL statement for the requested application could not be found in the customization table.

ACTION: Ensure that the application name was spelled correctly. If so, then check to see if the application appears in the option table. If the application is participating in compression only by the use of the \$DEFAULT entry, in the option table, then the statistics for this application will be a combined part of the statistics of \$DEFAULT.

#### VEXDS007. FUNCTION AVAILABLE FOR LOCAL APPL ONLY

The requested function failed because that function may only be performed from within the application for which the function was issued. For example, you can only view LU statistics for the application where you are logged-on. To view LU statistics for another application, you must log-on to that application.

ACTION: Log-on to the proper application; then issue the command.

#### VEEXPR001. - INVALID OPTION

The PARM statement command to VTEXPRES when running as a batch program is either syntactically incorrect or specifies an unknown command.

ACTION: Correct the PARM statement and rerun the job. Valid commands are ACT, DEACT, BACKOUT and DSN. For DSN, the syntax is DSN(dataset(member)).

#### VEEXPR002. - CALLER NOT AUTHORIZED

The VTEXPRES program was not link-edited into an authorized library or APF authorization was not set at link-edit time. Execution can not continue.

ACTION: Move the VTEXPRES load module into an authorized library, then rerun the job.

#### VEXPR003. - COMPRESSION ACTIVATED

This message indicates successful completion of an activation request, using either the ACTIVATE or DSN command. The compression routines are now operating.

ACTION: No action is required. This is an informational message.

#### VEXPR004. - ACTIVATE FAILED

This message indicates that VTEXPRES was unable to activate compression. This message will be preceded by message VEXPR010, VEXPR011 or VEXPR012 which gives the reason for the failure.

ACTION: Refer to the explanation of the preceding message to determine the reason for the failure and the action to take.

#### VEXPR005. - OPTION FILE LOAD FAILURE - RC=xx

#### VEXPR005. - TABLE LOAD FAILURE - RC=xx

This message indicates that VTEXPRES was unable to build the customization table for some reason. The xx in the message is a return code which gives the reason for the failure. The return code will be one of the following:

- 04 - Allocation failure for customization table dataset
- 08 - PDS open failure
- 0C - Member not found in PDS
- 10 - Member open failure
- 14 - Subsystem table entry VEXP not defined

ACTION: Verify that the DSNAME and member name are specified correctly and the right library is being used. If the cause of the error is not obvious from the return code, notify Unicom Systems Inc. Systems technical support.

#### VEXPR006. - COMPRESSION DEACTIVATED

This message indicates successful completion of a deactivation request, using the DEACTIVATE command. The compression routines are now quiesced.

ACTION: No action is required. This is an informational message.

#### VEXPR007. - EXPRESS SVC NOT INSTALLED

When attempting to activate compression, the type-3 user SVC could not be located in the system SVC table. Either the SVC was not installed in this system or an IPL has not been done since installing it.

ACTION: Review the installation instructions and determine whether the SVC was properly installed. If necessary, perform an IPL to activate the SVC.

#### VEXPR008. - AUTOMON@-VTAM HAS EXPIRED ,CODE=x

This message indicates that the product has expired or cannot be used because of an installation error. The product is now inactive and cannot be used.

The "x" in the above message will be one of the following error codes:

- 1 -The current date is greater than the password expiration date.
- 2 -Module "STSPASS" could not be found in a library in the search string.
- 3 -The product ID could not be located in STSPASS.

- 4 -The password in STSPASS is invalid.
- 5 -Module "STS0100" could not be found in a library in the search string.
- 6 -STSPASS contains a CPU ID that is invalid for the machine that it is installed on.
- 7 -The password entered is not valid for the requested product.
- 9 -An undetermined error has been encountered. Please call Unicom Systems Inc.Systems Technical Support at(818) 838-0606 .

ACTION: Correct the error if possible, or call Unicom Systems Inc.Systems Technical Support at(818) 838-0606 .

VEXPR009. - AUTOMON@-VTAM WILL EXPIRE IN xx DAYS

This message indicates that the end of the trial period for AUTOMON@-VTAM is approaching. This message will start appearing within seven days of the expiration date.

ACTION: No action is required at this time; the product is still operational. If it is allowed to expire, however, it can no longer be used. Notify your sales representative at Unicom Systems Inc.Systems.

VEXPR010. - UNABLE TO LOCATE xxxxxxxx

During the processing of an activate request, one of the modules required to be in the LPA could not be located. The xxxxxxxx in the message indicates the name of the required LPA module.

This is probably an installation error of some kind.

**ACTION:** Review the installation steps. Verify that all modules are installed in correct libraries and that an IPL was done with the CLPA command to create the LPA.

#### VEXPR011. - VTEXPRES - UNABLE TO LOCATE ATCVT POINTER

During the processing of an activate request VTEXPRES was unable to locate the VTAM CVT pointer.

This error can occur if you reloaded the VTEXMAIN module into the LPA/SVA without first doing a BACKOUT, then recycling all applications that were currently compressing. If this is the first activation, it may mean that an unsupported version of VTAM is installed.

**ACTION:** This is a serious error which indicates that AUTOMON®-VTAM cannot operate in this environment as currently configured. If it was caused by the reload process described above, you must re-IPL in order to activate. (Refer to *RELOADING THE VTEXMAIN MODULE* in section 04 (MVS) or 05 (DOS) for more details). Otherwise, notify Unicom Systems Inc. Systems technical support for further analysis.

#### VEXPR012. - VTEXPRES - ATCVT POINTER ERROR

During the processing of an activate request the VTAM CVT is not properly constructed.

This error can occur if you reloaded the VTEXMAIN module into the LPA/SVA without first doing a BACKOUT, then recycling all applications that were currently compressing. If this is the first activation, it may mean that an unsupported version of VTAM is installed.

**ACTION:** This is a serious error which indicates that AUTOMON®-VTAM cannot operate in this environment as currently configured. If it was caused by the reload process described above, you must re-IPL in order to activate. (Refer to *RELOADING THE VTEXMAIN MODULE* in section 04 (MVS) or 05 (DOS) for more details). Otherwise, notify Unicom Systems Inc. Systems technical support for further analysis.

#### VEXPR013. - MODULE VTEXMAIN NOT LOADED IN SVA

The VTEXMAIN module could not be found in the SVA. Either the job to load the SVA module has not been run since AUTOMON®-VTAM was installed or an IPL has been performed and the VTEXMAIN entry has not been added to the SET SDL statement in the ASI procedure.

**ACTION:** Load VTEXMAIN in the SVA as described in section 02 - *INSTALLATION*, then activate compression.

#### VEXPR014. - JOB MUST RUN IN SAME ASID AS VTAM

An attempt is being made to activate or backout compression and the VTEXPRES program is running in a partition which is in a different address space than VTAM. VTEXPRES must run in the same address space as VTAM in order to activate.

**ACTION:** Use the MAP command to locate a partition in the same address space as VTAM. Run VTEXPRES in that partition.

#### VEXPR015. - COMPRESSION BACKED OUT

This message indicates successful completion of a deactivation request, using the BACKOUT command. The compression routines are now quiesced, and the interception hook is removed from VTAM.

**ACTION:** No action is required. This is an informational message.

VEXPR030. - LABEL TOO LONG

While interpreting the customization table statements, a label field was found which exceeds 8 characters. The statement in error displays with this message.

ACTION: The statement in error was bypassed. Correct the statement and re-activate.

VEXPR031. - ERROR OBTAINING SVA STORAGE

Storage could not be obtained in the SVA to build the option control table from the customization table.

ACTION: The allocation of storage for the SVA must be increased.

VEXPR033. - NO VERB PRESENT

While editing the customization table statements, a statement was found which had no recognizable verb. The valid verbs are VSTART, VAPPL, VLU, VSKIP, VTRACE and VFTRAN.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR034. - INVALID VERB

While editing the customization table statements, a statement was found which had no recognizable verb. The valid verbs are VSTART, VAPPL, VLU, VSKIP, VTRACE and VFTRAN.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR035. - NO OPERAND PRESENT

The verb of this customization statement requires at least one operand to be present. None could be found.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR036. - INVALID KEYWORD VALUE - xxxxxxxx

While editing the customization table statements, a statement was found with an invalid value for a keyword. The keyword in error is displayed by xxxxxxxx in the message.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR037. - INVALID OPERAND VALUE - xxxxxxxx

While editing the customization table statements, a statement was found with an unrecognizable keyword. The keyword in error is displayed by xxxxxxxx in the message.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR038. - DDNAME OMITTED

No DDNAME could be found for the TRACDDN or STATDDN keywords.

ACTION: If the keyword is present, there must be a DDNAME specified. Correct the statement and re-activate.

VEXPR039. - INVALID HEX CHARACTER - xxxxxxxx

The data for the keyword designated by xxxxxxxx is not valid hexadecimal data.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR040. - TEXT EXCEEDS MAX LENGTH - 40 - xxxxxxxx

The data for the keyword designated by xxxxxxxx exceeds the maximum allowable number of characters, which is 40.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR041. - MISSING LABEL FIELD

The verb of this customization statement requires that a label be present in position 1. None could be found.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR042. - INVALID ROW/COL SPECIFICATION - xxxxxxxx

The ROWCOL keyword of the VSKIP statement is either syntactically incorrect or contains an invalid row or column.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR043. - INVALID SCREEN SIZE - xxxxxxxx

The PSIZE or ASIZE keyword of the statement is either syntactically incorrect or contains an invalid screen size. Screen size can be any number from 1920 to 9999. The keyword in error is designated by xxxxxxxx in the message.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR044. - MAXIMUM VALUE EXCEEDED - xxxxxxxx

The data for the keyword designated by xxxxxxxx exceeds the maximum allowable value for that keyword.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR045. - ENTER PASSWORD FOR ACCESS  
VEXPR045. - ENTER PASSWORD TO ACTIVATE  
VEXPR045. - ENTER PASSWORD TO DEACTIVATE

A request for a function has been issued; however, this function is password-protected. If this function is to be completed, the correct password must be entered at this time. The password may be from 1 to 8 characters in length and will not display on the screen.

ACTION: Key the password and press ENTER.

VEXPR046. - DENIED - INVALID PASSWORD

The request has been denied because the password entered was incorrect.

ACTION: Re-attempt the request and supply the correct password.

VEXPR047. - HEX DIGITS NOT IN PAIRS - xxxxxxxx

The data for the keyword designated by xxxxxxxx is valid hexadecimal data but is not an even number of hex digits. You must enter hex data in pairs of digits where one pair = one byte.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR048. - INVALID HEX DIGITS - xxxxxxxx

The data for the keyword designated by xxxxxxxx is not valid hexadecimal data.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR049. - OFFSET SPECIFICATION NOT NUMERIC - xxxxxxxx

The data for the keyword designated by xxxxxxxx is not valid numeric data.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR050. - NO DATA PRESENT FOR OFFSET - xxxxxxxx

An offset has been indicated for the keyword designated by xxxxxxxx but no offset data could be found.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR051. - OFFSET SPECIFICATION TOO LARGE

An offset has been specified in a VFTRAN statement; however, the offset is too large to be a valid offset. This offset can not exceed 255 (x'FF').

ACTION: Correct the statement and re-activate.

VEXPR052. - NO START/STOP DATA PRESENT

The start or stop data, of a VFTRAN statement, could not be found. This is a required entry.

ACTION: Correct the statement and re-activate.

VEXPR053. - MISSING COMMA FOR OFFSET - xxxxxxxx

There needs to be a comma between the data and the offset on the VFTRAN statement. The xxxxxxxx in the message indicates the START or STOP keyword in question.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR054. - MISSING PARENTHESIS OR QUOTE - xxxxxxxx

The data for the keyword designated by xxxxxxxx is syntactically incorrect. There is an unbalanced parenthesis or quote in the statement.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR055. - COMPRESSION ENABLED

This is a confirmation message indicating successful execution of the VTEXPRES ON command.

ACTION: This is an informational message. No action is required.

VEXPR056. - COMPRESSION DISABLED

This is a confirmation message indicating successful execution of the VTEXPRES OFF command.

ACTION: This is an informational message. No action is required.

VEXPR057. - INVALID HEX/CHAR DELIMITER - xxxxxxxx

The data for the keyword designated by xxxxxxxx is syntactically incorrect. AUTOMON®-VTAM has found the quotes used to designate the enclosed characters as data; however, there is no 'X' or 'C' present to specify hex or character data.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR058. - SYNTAX ERROR IN INCL= OR EXCL=

A syntax error has been found in an 'INCL=' or 'EXCL=' statement. For proper syntax, please refer to section 03 - *CUSTOMIZATION*, under the discussion *VLU - IDENTIFY LOGICAL UNITS FOR COMPRESSION*.

ACTION: The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR059. - VAPPL NOT PRESENT FOR APPLID

The APPLID referenced in the 'VLU INCL=' or 'VLU EXCL=' statement has not been previously defined in a VAPPL statement.

ACTION: The APPLID in the VLU statement must be a valid application ID and must be defined in a VAPPL statement before the VLU statement occurs. The statement in error is displayed with this message. Correct the statement and re-activate.

VEXPR061. - TYPE NEW OPTION DSN OR HIT ENTER

A request to activate EXPRESS (or reactivate with a different customization table) has been issued, but has not yet been executed. This message is requesting the name and location of the customization table to be used for option settings.

ACTION: If this is the first activation (or the first activation after a previous backout) you will need to specify the User Option Table that is to be used by EXPRESS. If this is a subsequent activation, pressing ENTER will reload the same customization table, or you may key a different table for EXPRESS to use. For specifying a User Option Table, the proper syntax would be:

library.name(member name)

For more information, please refer to *CONSIDERATIONS FOR ACTIVATION OF COMPRESSION* or *ENTERING COMMANDS AT THE TSO STATISTICS DISPLAY* in the section entitled *OPERATION - MVS SYSTEMS*.

VLUDS001. STATISTICS NOT AVAILABLE, MEMORY SWAPPED-OUT

This message can occur when attempting to view LU statistics for an application other than the one to which you are currently logged-on. The memory in that region has been swapped-out. The LU name shown is actually the ACB name when this message is present. This occurs most often in TSO.

ACTION: This is an informational message. No action is required.

VLUDS002. NO LU ENTRIES FOUND FOR REQUEST

Either there are no LU entries active for this application or the LU name that was entered to begin the display could not be found.

ACTION: If an LU name was entered, try starting with no LU name and browsing through the display. If there are no active entries it may be that the application is not active.

VLUDS003. ERROR RETURN FROM AXSET OR AXRES

During cross-memory access for the LU display from a different application, an error return was posted by MVS for one of the macros listed above.

ACTION: This is a "should-never-occur" error. Notify Unicom Systems Inc. Systems Technical Support.

VMAIN001. VTEXMAIN ACTIVE - VECTOR xxxxxxxx

This message indicates that compression has been activated for this application. The xxxxxxxx in the message is the storage address of the vector control block for this application. This address is sometimes needed when debugging a problem situation.

ACTION: No action is required. This is an informational message.

VMAIN002. VTEXMAIN OPEN FAILURE - xxxxxxxx

The file designated by the DDNAME in the message could not be successfully opened for output. This probably indicates that the file is not defined properly.

ACTION: Make sure that the file has been defined, either through JCL or with TSO. If the problem cannot be resolved, notify Unicom Systems Inc. Systems Technical Support.

VMAIN003. VTEXMAIN OUTPUT HALTED - xxxxxxxx

The output for a trace or statistics recording file has been stopped because the file has filled up and the WRAP option is set to NO.

ACTION: No action is required. You may need to enlarge this file to allow for more output, or set the WRAP option to YES.

VMAIN004. VTEXMAIN OUTPUT WRAPPING - xxxxxxxx

The output for a trace or statistics recording file has filled up and restarted at the beginning of the file. This action is dictated by the WRAP option of the VAPPL statement.

ACTION: No action is required. You may need to enlarge this file to allow for more output.

VMAIN005. AUTOMON@-VTAM FILE xxxxxxxx OPENED

The statistics recording or global trace file designated by the DDNAME in the message has been successfully opened.

ACTION: No action is required. This is an informational message.

VMAIN006. UNABLE TO ALLOCATE FILE xxxxxxxx, CODE=xxxx

The file designated by the DDNAME in the message could not be successfully allocated to this application region. This probably indicates that the file is not defined properly or it is in use with DISP=OLD in another region. The xxxx code value is the return code from the SVC 99 allocation processing.

ACTION: Make sure that the file has been defined, either through JCL or with TSO and is not in use. If the problem cannot be resolved, notify Unicom Systems Inc. Systems Technical Support.

VMAIN007. AUTOMON@-VTAM TRACE FILE OPENED - xxxxxxxx

The global trace file designated by the DDNAME in the message has been successfully opened.

ACTION: No action is required. This is an informational message.

VMAIN007. AUTOMON@-VTAM STATISTICS FILE OPENED - xxxxxxxx

The statistics recording file designated by the DDNAME in the message has been successfully opened.

ACTION: No action is required. This is an informational message.

VMAIN008. GETVIS UNAVAILABLE, COMPRESSION STOPPED

VTEXMAIN was unable to obtain any more storage from the GETVIS area of the application partition. This causes compression to be automatically quiesced for the application. The storage cushion has been released in order to output this message and quiesce the application compression.

**ACTION:** You need to allocate more GETVIS to this partition or reduce the amount of storage needed by VTEXMAIN. This can be done by using the VIO buffer pool or by using compressed images (CPRIMG).

**Warning:** Do not restart compression for this partition without bringing the application down and back up. The GETVIS area of the partition is near exhaustion and the AUTOMON®-VTAM storage cushion has already been released. It will not be able to recover again, which would cause the application to ABEND.

VMAIN009. AUTOMON®-VTAM TRACE FILE CLOSED - xxxxxxxx

The global trace file designated by the DDNAME in the message has been successfully closed. You can now run the trace print program.

**ACTION:** No action is required. This is an informational message.

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

This message can occur during normal transaction operation. It will happen if the terminal input area sizes in the VTAM application are too small to contain the input TIOA after inserting the MDT fields.

**ACTION:** The area sizes of the application must be increased. Refer to *MINIMUM TERMINAL INPUT SIZES* in section 02, *INSTALLATION* for more information.

VSTAT001. OPEN ERROR ON FILE xxxxxxxx.

An open error has occurred while attempting to open the input statistics file. This is probably a JCL error or the file does not exist.

ACTION: Verify the JCL for the statistics print operation. Verify that the file is present and has data in it. If the problem cannot be resolved, notify Unicom Systems Inc.Systems Technical Support.

VSTAT003. FILE I/O ERROR HAS OCCURRED

An I/O error has occurred while processing the input statistics file.

ACTION: Verify the JCL for the statistics print operation. Verify that the file is present and has data in it. Verify that there is a valid end-of-file indicator in the file. If the problem cannot be resolved, notify Unicom Systems Inc.Systems Technical Support.

VSTAT004. END OF STATISTICS

This is an informational message indicating that the end of input has been reached for the statistics print program.

ACTION: No action is required.

VTRAC001. OPEN ERROR ON TRACE FILE.

An open error has occurred while attempting to open the input global trace file. This is probably a JCL error or the file does not exist.

ACTION: Verify the JCL for the statistics print operation. Verify that the file is present and has data in it. If the problem cannot be resolved, notify Unicom Systems Inc.Systems Technical Support.

VTRAC002. END OF TRACE

This is an informational message indicating that the end of input has been reached for the global trace print program.

ACTION: No action is required.

## APPENDIX A - STORAGE UTILIZATION

### STORAGE UTILIZATION IN MVS/XA

<b>Customization table</b>	- CSA below 16M line.  128 bytes of header information. 224 bytes per VAPPL statement. 9 bytes per VTRACE statement. 14 bytes per VLU statement. 13 bytes + text length per VSKIP statement 3 bytes + text length per VFTRAN statement
<b>Vector table</b>	- Private area below 16M line.  768 bytes per opened ACB in TSO. 8800 bytes in CICS and IMS 8800 bytes per ACB in other applications
<b>SMF record buffer</b>	- Private area below 16M line.  64 bytes of SMF recording is being used.
<b>Terminal entries</b>	- Private area above 16M line, one entry per logged-on terminal.  228 bytes per terminal control block.  Image buffer which is the larger of the default or alternate screen size. If CPRIMG=Y, the average image size is approximately 35% or the full screen size.  If EWIMG=N the image buffer is not present.  Certain VTAM requests require working storage for the life of the request. The amount varies but falls in the range of 408 to 1000 bytes. For SEND requests, add the size of the application send buffer.
<b>VTEXMAIN</b>	- LPA module - 72K
<b>SVC routine</b>	- CSA below 16M line - 312 bytes.

## STORAGE UTILIZATION IN DOS/VSE

<b>Customization table</b>	- SVA storage  128 bytes of header information. 224 bytes per VAPPL statement. 9 bytes per VTRACE statement. 14 bytes per VLU statement. 13 bytes + text length per VSKIP statement 3 bytes + text length per VFTRAN statement
<b>Vector table</b>	- Partition GETVIS area 8800 bytes in CICS 8800 bytes per ACB in other applications
<b>Terminal entries</b>	- Partition GETVIS area. One entry per logged-on terminal.  228 bytes per terminal control block.  Image buffer which is the larger of the default or alternate screen size. If CPRIMG=Y, the average image size is approximately 35% of the full screen size.  If the VIO buffer pool is in use, Storage usage in GETVIS is dramatically reduced.  If EWIMG=N the image buffer is not present.  Certain VTAM requests require working storage for the life of the request. The amount varies but falls in the range of 408 to 1000 bytes. For SEND requests, add the size of the application send buffer.
<b>VTEXMAIN</b>	- SVA module - 40K.

## APPENDIX B - INSTALLATION DOCUMENTATION

Following is the installation documentation that is printed in the second installation step.

### INSTALLATION STEPS - MVS

```
*****
*      PREPARATION OF THE AUTOMON@-VTAM INSTALLATION JCL      *
*****
```

The JCL has been created in the member name of the PDS that you supplied for the SYSPCH DDNAME in step 2 of the basic product install.

If for some reason you cannot or do not want to use the installation JCL from step 2 of the installation instructions, use the printed output from step 2 to create the JCL in any fashion desired.

After creating the JCL, by whatever means, modify it to fit your system's specific requirements. The optional entries are noted and filled with question marks. When the modified JCL is ready, submit it along with the necessary tape mounts.

This JCL accomplishes the following three steps:

- 1). Extract AUTOMON@-VTAM from the product master install tape and create a deblocked installation tape.
- 2). Load all of the AUTOMON@-VTAM object modules to their appropriate libraries.
- 3). Load the skeleton customization table to a PDS member.

```
*****
*      THE INSTALLATION TAPE      *
*****
```

The deblocked installation tape contains five files, as follows:

- 1). VTEXMAIN object module. This member handles all of the datastream compression functions. This member must be link-edited into SYS1.LPALIB or any authorized library where LPA-eligible members are kept. If desired, a new linklib can be defined and added to the MLPA list in SYS1.PARMLIB as described later in this document under 'CREATING A MODIFIED LPA LIST'.
- 2). VTEXPRES object module. This member controls the activation and deactivation of AUTOMON@-VTAM and is used in TSO to display statistics and other control functions. This member must be link-edited to an authorized LINK library such as SYS1.CMDLIB.  
  
VTEXCORE object module. This member can be used to display/alter core from the TSO COMMAND facility. This member must be link-edited to an authorized link library such as SYS1.CMDLIB.
- 3). The AUTOMON@-VTAM SVC routine. This module can be link-edited into SYS1.LPALIB and supplied with the name 'IGC00xxx', where xxx is the SVC number to be used. This will be described in more detail later.
- 4). Four support program modules, VTEXDISP, VTEXPRNT, VTEXTRAC, and VTEXSTAT. The first two modules, VTEXDISP and VTEXPRNT are CICS programs, the remaining two are batch programs. If you are running CICS, you should link-edit these programs into a CICS eligible library. You may then move the batch programs to another library if desired.
- 5). A sample customization table for use in tailoring your own table. These are source command statements that can be loaded into SYS1.PARMLIB or any PDS with an 80-byte record size.

```

*****
* THE INSTALLATION JCL *
*****

//STSINST JOB 1,SOFTOUCH-SYSTEMS,MSGCLASS=X,CLASS=A
//*****
/* STEP1 - CREATE THE AUTOMON@-VTAM PRODUCT INSTALL TAPE *
//*****
//STEP1 EXEC PGM=STSINST
//SYSPRINT DD SYSOUT=*
//STEPLIB DD DSN=??????.DISP=SHR * NOTE 1 *
//MPRDIN DD DSN=MASTER.TAPE,UNIT=TAPE, * NOTE 2 *
// VOL=SER=MASTER,LABEL=(2,NL),
// DCB=BLKSIZE=32000
//MMAINOT DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 3 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(1,NL)
//MPRESOT DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 3 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(2,NL)
//MSVCOT DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 3 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(3,NL)
//MCICSOT DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 3 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(4,NL)
//MSKELOT DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 3 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(5,NL)
//SYSIN DD *
PRODUCT=EXPRESS PRODUCT=EXPRESS
MODE=CREATE (CREATE A PRODUCT INSTALL TAPE)
OPSYS=MVS (OPERATING SYSTEM)
LINES=56 (CAN BE 1 TO 99)
/*
//*****
/* STEP 2 - LINK EDIT THE VTEXMAIN MODULE *
//*****
//STEP2 EXEC PGM=IEWL,PARAM='LIST,LET,XREF'
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 4 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(1,NL),
// DCB=BLKSIZE=3200,DISP=(SHR,PASS)
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(20,20))
//SYSLMOD DD DSN=SYS1.LPALIB(VTEXMAIN),DISP=SHR * NOTE 5 *
/*
//*****
/* STEP 3 - LINK EDIT THE VTEXPRES MODULE *
//*****
//STEP3 EXEC PGM=IEWL,PARAM='LIST,LET,XREF,AC=1'
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 4 *
// VOL=(,RETAIN,,,SER=VTEX),LABEL=(2,NL),
// DCB=BLKSIZE=3200,DISP=(SHR,PASS)
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(20,20))
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR * NOTE 6 *
//*****
/* STEP 4 - LINK EDIT THE USER SVC MODULE (OPTIONAL) *
//*****
//STEP4 EXEC PGM=IEWL,PARAM='LIST,LET,XREF'
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=VTEX.OBJ,UNIT=TAPE, * NOTE 4 *

```

```

//                                VOL=(,RETAIN,,,SER=VTEX),LABEL=(3,NL),
//                                DCB=BLKSIZE=3200,DISP=(SHR,PASS)
//SYSUT1      DD      UNIT=SYSDA,SPACE=(1024,(20,20))
//SYSLMOD     DD      DSN=SYS1.LPALIB(IGC00???),DISP=SHR      * NOTE 7 *
//*****
//*      STEP 5 - LINK EDIT THE SUPPORT PROGRAMS      *
//*****
//STEP5       EXEC     PGM=IEWL,PARM='LIST,LET,XREF'
//SYSPRINT    DD      SYSOUT=*
//SYSLIB      DD      DSN=CICS???.LOADLIB,DISP=SHR      * NOTE 8 *
//SYSLIN      DD      DSN=VTEX.OBJ,UNIT=TAPE,           * NOTE 4 *
//                                VOL=(,RETAIN,,,SER=VTEX),LABEL=(4,NL),
//                                DCB=BLKSIZE=3200,DISP=(SHR,PASS)
//SYSUT1      DD      UNIT=SYSDA,SPACE=(1024,(20,20))
//SYSLMOD     DD      DSN=CICS???.LOADLIB,DISP=SHR,      * NOTE 4 *
//*****
//*      STEP 6 - LOAD THE SKELETON CUSTOMIZATION TABLE      *
//*****
//STEP6       EXEC     PGM=IEBUPDTE
//SYSPRINT    DD      SYSOUT=*
//SYSUT1      DD      DSN=SYS1.PARMLIB,DISP=SHR      * NOTE 10 *
//SYSUT2      DD      DSN=SYS1.PARMLIB,DISP=SHR      * NOTE 10 *
//SYSIN       DD      DSN=VTEX.OBJ,UNIT=TAPE,           * NOTE 4 *
//                                VOL=(,RETAIN,,,SER=VTEX),LABEL=(5,NL),
//                                DCB=BLKSIZE=3200,DISP=(SHR,PASS)

```

\*  
 \*\*\*\*\* Notes ...  
 \*

1. Must be the library that "STSINST" phase was link-edited to in the initial JCL process from the Installation guide.
2. This tape drive should have the mounted product master tape received from Unicom Systems Inc. Systems.
3. This tape drive should be a scratch tape. The EXPRESS installation files will be written to this drive. It then becomes the input for the remainder of the install process. All files will be written to the same volume.
4. This tape drive should have the mounted EXPRESS installation tape. (Output from step 1).
5. This is the link library which will be used to load from when the system is IPL'd. It must be an authorized library for LPA-eligible programs. If you prefer to load VTEXMAIN into a modified LPA, refer to 'CREATING A MODIFIED LPA LIST', following. You must code the member name as 'VTEXMAIN'.
6. If SYS1.CMDLIB is not used, this library must be an authorized LINK library, eligible for TSO command processor programs.
7. This step is not required but may be performed.

Since this is a type 3 SVC routine, the load library must be SYS1.LPALIB. For the member name, the first five characters must be 'IGC00'. For the last three characters, pick an unused type 3 SVC number. Replace the first two ?? with the first two digits of that number. The last character must be the alphabetic character which is produced by adding a 'C' zone to the last digit of the SVC number.

Thus, if the last digit of the SVC number is '1', adding a 'C' zone makes it x'C1', or the character 'A'. The following table lists the possibilities.

If the last digit of the SVC number is ...

- '1' - The character would be 'A'
- '2' - The character would be 'B'
- '3' - The character would be 'C'
- '4' - The character would be 'D'
- '5' - The character would be 'E'
- '6' - The character would be 'F'
- '8' - The character would be 'H'
- '9' - The character would be 'I'
- '0' - The character would be '{' (x'C0' = Left bracket)

This SVC routine must be defined to the system via the IEASVC00 member in SYS1.PARMLIB. The procedure for this is described below in step 7.

8. The library for the SYSLIB statement must contain the CICS command-level stub program (DFHEAL) used by command-level assembler programs.
9. This can be any load library. If you are running CICS, it must be a CICS-eligible library. You may want to move the two batch programs, VTEXSTAT and VTEXTRAC to a batch loadlib.

If you run the high-performance option of CICS, you must install the the VTEXCICS program and define it to CICS as a PLTSI member. Refer to step 8, below for this procedure. For information on the high performance option, see 'SELECTING THE HIGH PERFORMANCE OPTION' in the 'CICS INSTALLATION AND OPERATIONS' manual.

10. If AUTOMON®-VTAM has been previously installed using this same PDS, there is no real need to run step 6, since you will have prepared your own customization table.

If the VTEXSKEL member already exists in this PDS, the version from the tape will not load unless you delete it first. If you are using this VTEXSKEL member as a production customization table, do not delete it as this step will simply bypass the load.

The library must be a PDS defined with 80-byte records. It should be a library such as SYS1.PARMLIB where command streams are kept. The customization table is not assembled. The commands in the table are accessed directly by VTEXPRES and interpreted at execution time.

```
*****
*          STEP 7. INSTALL THE TYPE 3 SVC ROUTINE:          *
*****
```

If you want AUTOMON®-VTAM to always dynamically allocate the user SVC, you may skip this step. If you prefer to keep a permanent record in IEASVC00 of all SVCs in use, follow this procedure.

The SVC number assigned to the SVC routine used with AUTOMON®-VTAM must be defined to the system. MVS/SP users must perform a Type 3 SVC generation using normal procedures. MVS/XA users can perform the following procedure:

Edit member IEASYS00 in SYS1.PARMLIB and locate the SVC= keyword. It will usually say SVC=00. Whatever the number is becomes the nn suffix to the IEASVCnn member to be edited next.

Edit member IEASVCnn in SYS1.PARMLIB and add the following statement:

```
SVCPARM xxx,REPLACE,TYPE(3),EPNAME(IGC00xxy)
```

where xxx is an unused type 3 SVC number to be used. This number must agree with the last three digits of the member name of the IGC00xxy member loaded in step 4, above, after converting the 'C' zone of the last digit back to an 'F' zone. The EPNAME parameter specifies the same member name that was assigned to the SVC routine in step number 4.

To illustrate, if SVC number 255 were selected, the statement would read as follows:

```
SVCPARM 255,REPLACE,TYPE(3),EPNAME(IGC0025E)
```

\* Note \* Be sure and code SVC= on the VSTART statement of the customization table designating the same SVC number.

```
*****  
* STEP 8. INSTALL THE CICS STATISTICS DISPLAY PROGRAM, VTEXDISP. *  
*****
```

The VTEXDISP program is used to display on-line statistics for each application being compressed. It must be installed in every CICS system where it will be used. Note that this program will display statistics for applications, no matter which CICS system you invoke it from. In other words, it does not have to be installed in a particular CICS system in order to display statistics for that system.

To install VTEXDISP in CICS, do the following:

- 1). Add a PPT entry using RDO or macro assembly as follows:  

```
DFHPPT TYPE=ENTRY,PROGRAM=VTEXDISP,PGMLANG=ASSEMBLER
```
- 2). Add a PCT entry using RDO or macro assembly as follows:  

```
DFHPCTTYPE=ENTRY,TRANSID=VTEX,PROGRAM=VTEXDISP,TWASIZE=0
```
- 3). Now, install the group containing these entries (if RDO was used) or recycle CICS (if macro assembly was used) and the program will be available.

```
*****  
* STEP 9. INSTALL THE CICS LOCAL TRACE PRINT PROGRAM, VTEXPRNT.. *  
*****
```

The VTEXPRNT program is used to print the local trace entries to a CICS Printer. It must be installed in every CICS system where it will be used. After activating a local trace (by use of VTEX,TRACE) from a terminal, the local trace may be printed to a CICS printer using the VTEX,TDISP command.

To install VTEXPRNT in CICS, do the following:

- 1). Add a PPT entry using RDO or macro assembly as follows:  

```
DFHPPT TYPE=ENTRY,PROGRAM=VTEXPRNT,PGMLANG=ASSEMBLER
```
- 2). Add a PCT entry using RDO or macro assembly as follows:  

```
DFHPCT TYPE=ENTRY,TRANSID=VPRT,PROGRAM=VTEXPRNT,TWASIZE=0
```
- 3). Now, install the group containing these entries (if RDO was used) or recycle CICS (if Macro Assembly was used) and the program will be available.

```
*****  
* STEP 10. CHANGE APPLICATION SYSTEMS TERMINAL DEFINITIONS *  
*****
```

Changes to application system terminal definitions may be required. If CICS is in use, there are almost always changes required to the CICS TCT.

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:

```
'VMMAIN015.    BUFFER OVERRUN HAS OCCURRED, INPUT TIOA AREA SIZE MUST BE
                INCREASED'
```

```
*****
*                STEP 11. INSTALL THE STSCORE PROGRAM                *
*****
```

You should have received an accompanying document with AUTOMON®-VTAM that describes the memory display and alter program, STSCORE. While this utility is not a part of AUTOMON®-VTAM, it is very useful for analyzing control blocks and can be used to apply in-core PTFs.

We recommend that you install the STSCORE program. If you contact Unicom Systems Inc. Technical Support about any problem with AUTOMON®-VTAM, the support rep may ask you to use it to assist in diagnosing the problem.

STSCORE runs in CICS, the installation procedure is described in the reference guide for STSCORE. It was loaded in step 1 of the installation from the AUTOMON®-VTAM reference guide. You will need to move the STSCORE, STSPASS and STS0100 load modules to a CICS load library.

This ends the required steps to install AUTOMON®-VTAM. The following discussion deals with the procedure for creating a modified LPA list if you cannot use SYS1.LPALIB for the VTEXMAIN program module. Return now to the discussion entitled 'CREATION OF A CUSTOMIZATION TABLE' found in the 'INSTALLATION' section of the reference manual for further instructions.

\*\*\*\*\*

\*            CREATING A MODIFIED LPA LIST            \*

\*\*\*\*\*

If VTEXMAIN was not link-edited into SYS1.LPALIB, it must run in the modified LPA area. In order to load VTEXMAIN into a modified LPA, it must be included in the system modified LPA list. There may or may not be an existing modified LPA list established. In either case, use the following procedure to add VTEXMAIN to the list:

Edit member IEASYS00 in SYS1.PARMLIB. There must be a keyword in the form MLPA=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 IEALPA member. The IEASYS00 member contains many keywords such as:

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

Coding MLPA=00, for instance, would mean that there is another member in SYS1.PARMLIB with a member name of IEALPA00 which describes the MLPA definitions.

The IEALPAXX member contains one or more library/member name statements. These are the program identifiers for the programs to be loaded into the MLPA. The library and member name are separated by a space. Thus, to add VTEXMAIN to the list, add the following statement:

SYS1.LINKLIB VTEXMAIN

If you link-edited the VTEXMAIN member into some link library other than SYS1.LINKLIB, specify this link library name.

\* Note \* The procedure for adding a MLPA member is documented in the MVS INITIALIZATION AND TUNING manual.

INSTALLATION STEPS - DOS/VSE

\*\*\*\*\*  
\* PREPARATION OF THE AUTOMON®-VTAM INSTALLATION JCL \*

After running step 2 of the basic product install, the JCL has been created and written to the SYSPCH power queue. Use the appropriate command of your text editor to retrieve the JCL and create a member in your editor. If you are using ICCF, the 'GETP' command can be used for this purpose.

If for some reason you cannot or do not want to use the installation JCL from step 2 of the installation instruction, use the printed output from step 2 to create the JCL in any fashion desired.

After creating the JCL, by whatever means, modify it to fit your system's specific requirements. The optional entries are noted and filled with question marks. When the modified JCL is ready, submit it along with the necessary tape mounts.

This JCL accomplishes the three following steps:

- 1). Extract AUTOMON®-VTAM from the product master install tape and create a deblocked installation tape.
2). Link-edit all of the AUTOMON®-VTAM object modules to a common library.
3). Punch the skeleton customization table to the power punch queue.

\*\*\*\*\*  
\* THE INSTALLATION TAPE \*

The deblocked installation tape contains two files, as follows:

- 1). Object file containing all programs in AUTOMON®-VTAM. This file must be link-edited into any sub-library or core-image library. The programs contained in this file are:

- VTEXMAIN. This member handles all of the datastream compression functions.
VTEXPRES. This member controls the activation and deactivation of AUTOMON®-VTAM, as well as some other control function.
VTEXDISP. This is a command level CICS program which is used to display on-line compression statistics and perform other control functions.
VTEXPRNT. This is a command level CICS program which is used to print local trace entries to a CICS printer.
VTEXSORT AND VTEXDTFS. These are utility programs used in statistics capture and print.
VTEXSTAT AND VTEXTRAC. These are batch programs used to print statistics and trace entries, respectively.

- 2). A sample customization table for use in tailoring your own table. These are source command statements that can be loaded in your text editor for use.

\*\*\*\*\*  
\* THE INSTALLATION JCL \*

\* \$\$ JOB JNM=STSINST,CLASS=C
\* \$\$ PRT,HQ

```

// JOB STSINST                PRINT INSTRUCTIONS AND PUNCH JCL
* *****
*   STEP 1. -   CREATE EXPRESS INSTALL TAPE                               *
* *****
// ASSGN SYS011,???          INPUT MASTER PRODUCT TAPE
// ASSGN SYS012,???          (SCRATCH TAPE FOR OUTPUT)
// LIBDEF *,SEARCH=L.S      * NOTE 1 *
// EXEC STSINST
PRODUCT=EXPRESS
MODE=CREATE                (CREATE A PRODUCT INSTALL TAPE)
OPSYS=DOS                  (DOS, MVS, MVS/SP)
LINES=56                   (CAN BE FROM 1 TO 99)
/*
* *****
*   STEP 2. -   LNKEDT THE EXPRESS PHASES                               *
* *****
// ASSGN SYSIPT,???          * NOTE 2 *
// MTC REW,SYSIPT
// LIBDEF PHASE,CATALOG=L.S * NOTE 1 *
// OPTION CATAL
INCLUDE
// EXEC LNKEDT
/*
* *****
*   STEP 3. -   LOAD THE SAMPLE CUSTOMIZATION TABLE                   *
* *****
// RESET SYSIPT
// ASSGN SYS004,???          * NOTE 2 *
// MTC REW,SYS004
// MTC FSF,SYS004,1
// UPSI 1
// EXEC DITTO,SIZE=200K      * NOTE 3 *
$$DITTO TC INPUT=SYS004
$$DITTO EOJ
/*
/ &
* $$ EOJ

```

Now, move the member from the punch queue into your text editor using the appropriate command. If your text editor is ICCF, this would be the GETP command, in the form:

```
GETP STSINST A MEM=VTEXSKEL DELETE * NOTE 4 *
```

```

*****
*   JCL NOTE EXPLANATIONS                                             *
*****

```

- 1). Depending on your VSE system :  
L.S = LIBRARY.SUBLIBRARY to contain the phases.  
CL = CORE IMAGE LIBRARY to contain the phases.
- 2). SYSIPT should be assigned to the tape drive where the product install tape is mounted. (output from the 'CREATE' run).
- 3). There are several ways to load this file. This is just one example. You could catalog it into a source library, or load it directly into your text editor. It is a sequential file with 80-byte records. There are no control statements present on the front.

- 4). The first operand of the GETP command must correspond to the POWER JOB NAME. This step loads one source member. It should be loaded into your text editor under the name VTEXSKEL.

```
*****  
*          STEP 4 - INSTALL THE CICS STATISTICS DISPLAY PROGRAM          *  
*****
```

The VTEXDISP program is used to display on-line statistics for each application being compressed. It must be installed on every system where it will be used. Note that this program will display statistics for applications, no matter which CICS system you invoke it from. In other words, it does not have to be installed in a particular CICS system in order to display statistics for that system.

To install VTEXDISP in CICS, do the following:

- 1). Add a PPT entry using RDO or Macro Assembly as follows:

```
DFHPPT TYPE=ENTRY,PROGRAM=VTEXDISP,PGMLANG=ASSEMBLER
```

- 2). Add a PCT entry using RDO or Macro Assembly as follows:

```
DFHPCT TYPE=ENTRY,  
        TRANSID=VTEX,  
        PROGRAM=VTEXDISP,  
        TWASIZE=0
```

- 3). Now, install the group containing these entries (if RDO was used) or recycle CICS (if macro assembly as used) and the program will be available.

```
*****  
*          STEP 5 - INSTALL THE CICS LOCAL TRACE PRINT PROGRAM.          *  
*****
```

The VTEXPRNT program is used to print local trace entries for a terminal to a CICS printer after activating the local trace function by issuing the (VTEX,TRACE) command. VTEXPRNT must be installed in every CICS system where it will be used.

To install VTEXPRNT in CICS, do the following:

- 1). Add a PPT entry using RDO or Macro Assembly as follows:

```
DFHPPT TYPE=ENTRY,PROGRAM=VTEXPRNT,PGMLANG=ASSEMBLER
```

- 2). Add a PCT entry using RDO or Macro Assembly as follows:

```
DFHPCT TYPE=ENTRY,  
      TRANSID=VPRT,  
      PROGRAM=VTEXPRNT,  
      TWASIZE=0
```

- 3). Now, install the group containing these entries (if RDO was used) or recycle CICS (if Macro Assembly was used) and the program will be available.

```
*****  
*           STEP 6 - CHANGE APPLICATION SYSTEM TERMINAL DEFINITIONS           *  
*****
```

Changes to application system terminal definitions may be required if CICS is in use, there are almost always changes required to the CICS TCT.

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 VTAM receive requests in order to accept an input datastream.

With AUTOMON@-VTAM, in order for outbound compression to work properly, the minimum TIOA size 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 coded in DFHSIT. If this is the case, the default value is 256, which is always too small for AUTOMON@-VTAM.

Note that for CICS RELEASE 1.6, the 'RAMAX' keyword is coded at the beginning of the DFHTCT, rather than in DFHSIT. In release 1.7, it can occur in either place.

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
```

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 generation the datastream is executed:

```
'VTEX0100.    BUFFER OVERRUN HAS OCCURRED, INPUT RU SIZE OR TIOA SIZE MUST BE  
              INCREASED'
```

```
*****  
*           STEP 7. INSTALL THE STSCORE PROGRAM                               *  
*****
```

\*\*\*\*\*

You should have received an accompanying document with AUTOMON®-VTAM that describes the memory display and alter program, STSCORE. While this utility is not a part of AUTOMON®-VTAM, it is very useful for analyzing control blocks and can be used to apply in-core PTFs.

We recommend that you install the STSCORE program. If you contact Unicom Systems Inc. Technical Support about any problems with AUTOMON®-VTAM, the support rep may need to use it to diagnose the problem.

STSCORE runs in CICS, the installation procedure is described in the Reference Guide for STSCORE.

This ends the required steps to install AUTOMON®-VTAM. Return now to the 'INSTALLATION' section of the Reference Manual for further instruction.

## APPENDIX C - PROBLEM REPORTING

If a problem is encountered with **AUTOMON®-VTAM** that can not be resolved by the information in this document, you may call Unicom Systems Inc. Systems at (818) 838-0606 . You will be referred to a technical support representative who will take the information about the problem and resolve it then and there on the phone, if possible. Should a support person not be available when you call, you will be called back as soon as possible, usually the same day.

Prior to calling the support rep, be sure to review the section entitled *PROBLEM ANALYSIS AND DIAGNOSIS*. This section gives several hints to resolving problems, plus the types of diagnostic information that will be required for different problems.

If the problem can not be resolved on the phone, the support representative will require that you run certain diagnostic reports and send them to him. This will usually include a VTAM trace of the transaction or series of transactions involved, a core dump if one is available and a listing of your customization able.

When you call, please have the following information available:

- 1). Your Operating System.
- 2). The release of VTAM where AUTOMON®-VTAM runs.
- 3). The release and PTF level number of the AUTOMON®-VTAM that you have. This is displayed on the TSO terminal statistics screen for MVS systems and on the CICS statistics screen (VTEX transaction) for DOS/VSE systems. It is also typed on the label of the installation tape.
- 4). The release numbers of the application systems involved.
- 5). Any message numbers involved in the problem, whether produced by an application system, AUTOMON®-VTAM or the Operating System.

Unicom Systems Inc. Systems is committed to effective product support, and we will make every effort to resolve your problem as quickly as possible.

## APPENDIX D - THE PRODUCT LINE

This appendix briefly describes all of the products currently available from Unicom Systems Inc. Systems, Inc. These products are all available on the master installation tape and may be installed and evaluated if desired.

### **AUTOMON®-VTAM**

AUTOMON®-VTAM is our top-of-the-line CICS session manager, providing full transaction session management within CICS. In addition, interactive windowing allows up to four windows on one screen, each capable of operating transactions as if it were in a full screen.

AUTOMON®-VTAM allows the terminal operator to define their terminal as two, three, up to nine "logical", or "virtual" terminals. This means that a single terminal can operate more than one task at a time. A PF or PA key is used to "toggle" from one virtual terminal to another. When the operators toggles out of a transaction, the transaction screen and all pertinent information to restart the task is saved in off-line storage. When the operator returns, the screen is restored intact and the transaction may be continued as if it were not interrupted.

In addition, it is capable of "interactive windowing". This means that the terminal operator can divide the terminal screen into multiple parts, or windows, displaying a portion of each application task in each window. Operation of each task can continue while in window mode. Windows can be sized for best fit of the application screens, or the screen display can be "panned" left, right, up and down to view additional information. This feature makes it easy for the operator to work in one application while viewing another, thereby eliminating the need for external hard-copy reports or other media.

Two additional features of AUTOMON®-VTAM are outbound datastream compression, which can provide savings of up to 40 percent for all outbound terminal traffic, whether the terminal is using AUTOMON®-VTAM or not, and the "session view" feature, which allows an operator at one terminal to view the current display of another terminal in the network.

AUTOMON®-VTAM offers significant savings in operator time and productivity, due to the elimination of time-consuming transactions which are required to move from one application to another. Most users experience savings on the average of 30 minutes per day, per operator. This translates to thousands of dollars in time savings per day in many installations, where hundreds or even thousands of terminal operators are present.

### **CICS-JUGGLER**

CICS-JUGGLER is our mid-level CICS session manager providing full transaction session management within CICS. It contains all of the major features of AUTOMON®-VTAM except for interactive windowing and outbound data compression.

For users who want session management in CICS but do not care about interactive windowing, this product can be a good fit. It is priced in the mid-range between AUTOMON®-VTAM and CICS-JUGGLER/SVT, yet still offers support for all CICS environments and provides everything you need to take advantage of transaction session management.

### **CICS-JUGGLER/SVT**

CICS-JUGGLER/SVT is our lowest-level CICS session manager providing full transaction session management within CICS. This is a modified version of CICS-JUGGLER which is designed to offer sessions management to small computer installations. It performs the same basic functions of CICS-JUGGLER but does not have many of the extra features and "frills" that are present in CICS-JUGGLER and AUTOMON®-VTAM.

CICS-JUGGLER/SVT contains very few features beyond basic session management in a non-MRO environment. It's most attractive feature for many users is its low cost, with the ability to upgrade to the greater functionality of CICS-JUGGLER or AUTOMON®-VTAM paying only a cost differential.

## **CICS-FILESERV**

CICS-FILESERV is a software tool for the systems programmer in a CICS installation. It enhances some of the basic file-handling features of CICS. It provides the ability to define files to CICS dynamically. That is, a programmer can add an on-line file to the system without bringing the system down, which would normally be required. This means that the operation of all terminal operators can continue undisturbed, whereas it would normally be interrupted for 30 minutes or so each time a new file is defined. In addition, it allows on-line files to be opened and closed from another region, thereby alleviating scheduling bottlenecks in operations. It also provides significant savings in storage utilization within the CICS system.

CICS-FILESERV also provides the ability to dynamically access any file in the CICS environment for on-line display or maintenance purposes. This means that a programmer can view or change data records without writing a program, which can result in hours or days of saved time. In addition it offers a powerful file search facility, allowing the formulation of complex boolean queries to extract selected data from a file.

CICS-FILESERV provides significant advantage over standard CICS file control services by offering the following features:

- Resource Definition On-line (RDO) for the FCT
- Batch interface to open and close CICS files from a batch region.
- Dynamic allocation and deallocation (both DOS/VSE and MVS).
- Automatic (first-access) file opens for all releases of CICS.
  - Automatic file closes (after specified period of inactivity).
  - Automatic file closes (at specified time of day).
  - Automatic disable/enable of transactions and programs associated with a file.
  - Grouping of file names.
  - Display, update, add or delete records of any file.
  - Powerful file queries to locate and optionally extract, modify or reformat data records.

## **HELP-WINDOWS**

This product addresses the area of on-line documentation. It handles the two most-needed functions in this area, which are:

- a). Full on-line text maintenance. Complete manuals can be created and maintained on-line. Entire manuals or selected subjects may be printed on the system printer or displayed on the terminal in "list" form (as it appears when printed). Automatic page numbering, section titles, running headers and print extraction are supported. In addition, text from other systems can be imported into the HELP-WINDOWS text file.
- b). On-line retrieval of help text from application transactions. Selected sections of a manual, or stand-alone documents may be "keyed" to any application transaction in the on-line environment. A user-defined PF or PA key may be used to retrieve the associated text while operating the transaction. The associated text can be displayed either in full-screen mode or in a "window" on the screen, the position and size of which is defined by the text writer. No modifications to user programs are required to provide on-line help access. Help text can be created for any application - user-written transactions, vendor packages, fourth-generation languages, conversational programs, regardless of the method used by the application for building screens.

HELP-WINDOWS offers a "word-wrap" feature which allows text of any length to fully display in a window of any size. In addition, you can create help text with a user transaction by invoking the transaction screen, then using the cursor to "point" to the fields where help is to be made available. Help text can be associated at the transaction level, the screen level or the field level.

## VTAM-WINDOWS

VTAM-WINDOWS is a multiple sessions manager operating at the VTAM level. It has full support for interactive windowing, just like AUTOMON®-VTAM. With VTAM-WINDOWS, you can toggle and window different VTAM applications, such as CICS, TSO, IMS, etc. A strategic feature of VTAM-WINDOWS is the use of an asynchronous VSAM file for saving screen buffers, which results in a significant reduction in storage overhead over most other VTAM session managers. Because of this feature, VTAM-WINDOWS can run very effectively in the DOS/VSE and MVS/SP environments, in addition to MVS/XA.

In addition to basic VTAM session management and interactive windowing, VTAM-WINDOWS offers the following features and benefits.

- Gate-way processing, provides a single entry-point to the system, with pass-through connection to VTAM applications.
- AUTOMON®-VTAM interface, automatic activation of AUTOMON®-VTAM using gate-way.
- Automatic connection to an application in any or all sessions.
- Automatic sign-on to applications.
- Command stream processing, allows a series of transactions to be automatically sent to a VTAM application.
- Cut and paste, provides ability to copy data from one application screen into another.
- Session viewing, lets an operator at one terminal view the current display of any session of another terminal using VTAM-WINDOWS.
- On-line profile changes, allows the operator to alter the choices of PF/PA key usage, window configuration and number of sessions that were set in the customization table.

## AUTOMON®-VTAM

AUTOMON®-VTAM is a terminal datastream compression product which runs at the VTAM level. Its function is to eliminate, as much as possible, superfluous characters in datastreams that are transmitted to and from the host CPU and the terminals. Since the number of characters in a datastream is the major factor in determining the response time for an on-line transaction, the elimination of extra characters can significantly reduce that time. Most users experience a 55 to 80% reduction in terminal traffic with AUTOMON®-VTAM, which results in dramatic savings both in time and cost of the computer installation usage.

With AUTOMON®-VTAM you can have datastream compression for any VTAM application. CICS, IMS, TSO, any application system which communicates with terminals using VTAM can benefit. It does not require its own region or partition to operate.

AUTOMON®-VTAM performs all three major types of datastream compression, that is:

- Elimination of repetitive duplicate characters.
- Outbound image saving, sending only the data fields which changed on this display.
- Inbound mirroring, removing MDT fields so that they do not re-transmit.

Customization options are available in AUTOMON®-VTAM to exclude selected applications or terminals from compression, or control the various types of compression being performed. Statistics can be viewed from an on-line display and/or logged to a disk file for batch reporting.

An additional feature of AUTOMON®-VTAM is an enhanced buffer trace, which will capture and print the entire buffer contents of VTAM send and receive requests for a designated terminal. IBM's standard VTAM buffer trace will only show you the first 256 bytes of data.

# INDEX

- ACTIVATE command
  - description of.....58, 87
  - functions of.....59
  - relation to DSN.....52
- activation
  - of compression ....67, 77, 87, 96
  - of global trace...107, 109
- altering memory utility ... (see STSCORE)
- application name
  - in VAPPL statement...31
- APTYPE keyword
  - description of.....32
- ASIZE keyword
  - description of.....32, 44
  - in application display 62, 71, 91
- attributes
  - extended ..34, 44, 63, 72, 92
- authorized path
  - limitations with...31, 106
- backout
  - of compression53, 60, 87
- BACKOUT command
  - description of..53, 60, 87
- batch processing
  - SMF extract.....101
  - SORT.....102, 103
  - VTEXPRES program 51, 85
  - VTEXSTAT program .....103, 104
  - VTEXTRAC program .....108, 110
- CICS
  - displaying statistics ...70, 79, 89, 97
  - minimum TIOA size..14, 20, 37
  - required table changes .....14, 20
  - screen size control .....37
  - using local trace.....111
  - using MRO and ISC ..14, 19
- commands
  - ACTIVATE.....58, 87
  - BACKOUT ....53, 60, 87
  - DEACTIVATE....53, 59, 87
  - DSN.....52
  - NEWCOPY .....53
  - of STSCORE
    - LU .....137
    - VAPPL .....136
    - VECT .....136
    - VLU .....137
    - VMAIN .....136
    - VOPT .....136
  - OFF .....60
  - ON .....60
  - PURGE.....114, 115
  - RESET.....60
  - TDISP.....112, 113, 115
  - TRACE.....111, 114
- compression
  - activation of...52, 59, 67, 77, 87, 96
  - backout of.....52, 87
  - considerations of
    - activation .....54, 88
    - considerations of
      - deactivation ....54, 88, 89
      - deactivation of .....52, 53, 60, 67, 77, 87, 96
      - exclusion of terminals 43
      - for printers .....44
      - inbound.....6, 35
      - outbound.....5, 6
      - packages available.....4
      - percentage57, 66, 76, 95, 105
      - poor results with .....132
      - restarting for application .....67, 78, 96
      - restarting for LU ..67, 78, 96
      - restarting for system ..67, 77, 96
      - stopping for application .....67, 78, 96
      - stopping for LU ...67, 78, 96
      - stopping for system ...67, 77, 96
      - storage utilization -
        - DOS/VSE .....156
        - storage utilization - MVS .....155
    - deactivation .....54, 88, 89
    - deactivation of .....52, 53, 60, 67, 77, 87, 96
    - exclusion of terminals 43
    - for printers .....44
    - inbound.....6, 35
    - outbound.....5, 6
    - packages available.....4
    - percentage57, 66, 76, 95, 105
    - poor results with .....132
    - restarting for application .....67, 78, 96
    - restarting for LU ..67, 78, 96
    - restarting for system ..67, 77, 96
    - stopping for application .....67, 78, 96
    - stopping for LU ...67, 78, 96
    - stopping for system ...67, 77, 96
    - storage utilization -
      - DOS/VSE .....156
      - storage utilization - MVS .....155
  - techniques used..... 5
- COUNT keyword
  - description of ..... 47
- CPRIMG keyword
  - description of ..... 32
- in application display 63, 72, 92
- cross-domain
  - in VTAM.....14, 19
- CT keyword
  - description of ..... 111
  - use of .....112, 114, 115
- CUSTOMIZATION ..... 11
- customization options
  - activate statistics ..... 38
  - alternate screen size .. 32, 44
  - application exclusion . 32
  - assigning password.... 30
  - changing on-line ..68, 78, 96
  - CICS screen size control ..... 37
  - control erase/write imaging ..... 34
  - dataset name of LPA . 29
  - default action for
    - applications..... 40
    - default action for LUs 29
  - excluding transaction
    - datastreams ..... 44
    - excluding/including LUs ..... 42
  - extended attribute
    - compression.....34, 44
    - file wrap option..... 39
    - identify application type ..... 32
    - identifying applications ..... 31
    - image compression .... 32
    - inbound compression. 35
    - log-on delay ..... 33
    - maximum storage control..... 29
    - non-display field suppression .....36, 63, 72
    - null character control . 36
    - outbound user exit
      - specification of .. 39, 44

primary screen size.....36, 43	in application display.63, 73, 93	in VTAM .52, 53, 59, 60, 87, 88
read-buffer generation 37	DIAG keyword	IEAPAK
RU consolidation.....37	description of....112, 114	use of ..... 10
SMF recording.....30	use of .....114	image
statistics file DDNAME .....38	diagnostics ..... (see problem handling)	in inbound compression6
statistics file DSNNAME .....38	global trace ..... 106, 108	in outbound compression ..... 6
statistics reset.....37	local trace .....111	image saving
storage protect key .....35	SLIP tracing.....133	controlling ..... 34
trace file DDNAME ...38	display/alter utility..... (see STSCORE)	image-saving
trace file DSNNAME....39	DSN command .....59	description of ..... 5
VIO buffer pool usage 39	description of .....52	IMS
customization statements	erase command	considerations for ..... 15
VAPPL .....31	removal of .....6	minimum TIOA size.. 15
VFTRAN.....48	use of .....4, 5	performance
VLU.....42	erase input key	considerations ..... 15
VSKIP .....44	problems with.....131	INBND keyword
VSTART .....29	EWIMG keyword	description of ..... 35
VTRACE.....47	description of .....34	in application display 62, 71, 92
customization table	in application display.62, 72, 92	inbound mirroring
allocation of.....52	use of .....132	description of ..... 6
contents of skeleton...11, 17	EXCL keyword	specification of ..... 35
defining .....27	description of .....32, 43	INCL keyword
display of name .....56	exclusion	description of .....32, 43
format of.....27	of applications .....32	installation
generation of defaults .40	of logical units .....29, 42	MVS ..... 9
loading new version ..59, 81, 99	of transaction	of other products..... 8
name of..... 11, 17, 27	datastreams .....46	tape contents ..... 8
order of .....28	EXTATTR keyword	installation documentation
purpose of.....27	description of .....34, 44	MVS .....157, 164
statements of.....28	in application display.63, 72, 92	IPL
datastreams	use of .....133	creation of SVA..... 20
built with mapping .....4	extended attributes 34, 44, 63, 72, 92	JCL
exclusion of .....46	F keyword	example
inbound.....6	description of .....113	VTEXSTAT .....102, 103
inefficiency factors.....4	failures ..... (see problem handling)	link-edit object members .....21, 24
outbound.....5, 6	file transfers	load install program
DEACTIVATE command	coding examples .....49	DOS/VSE ..... 16
description of..53, 59, 87	IBM standard .....50	MVS ..... 9
functions of.....59	method of identifying .48	load source members 21, 24
deactivation	support for .....83, 99	product password install ..... 24
of compression ....53, 67, 77, 87, 96	FTRAN keyword	running the install program
of global trace...108, 110	description of .....35	DOS/VSE ..... 16
default action	generic application ID	MVS ..... 9
application statistics ..61, 71, 90	specification of .....31	KEY keyword
for unknown	help	description of ..... 35
applications .....40	at application display .67, 77, 96	keyword
DELAY keyword	at LU display ..69, 80, 98	CT.....114, 115
description of.....33	hook	DIAG ..... 114

OF.....	115	STATDSN.....	38	OFF command	
keywords		STATMON.....	31	description of.....	60
APPLID.....	31	STATS.....	38	OFS keyword	
APTYPE.....	32	STOP.....	48	description of.....	47
ASIZE.....	32, 44	SVC.....	31	ON command	
COUNT.....	47	TEXT.....	46	description of.....	60
CPRIMG.....	32	TO.....	113	PA keys	
CT.....	111, 112	TRACDDN.....	38	problems with.....	131
DELAY.....	33	TRACDSN.....	39	PAK.....	10
DIAG.....	112, 114	TYPE=INPUT.....	47	PARM statements	
EWIMG.....	34, 132	UEXIT.....	39, 44	of VTEXPRES.....	52, 87
EXCL.....	32, 43	VIO.....	39	password	
EXTATTR.....	34, 44, 133	WRAP 39, 108, 112, 114		entry of.....	59, 60
F.....	113	logical unit name		for product	
FTRAN.....	35	in VLU statement.....	42	description of.....	24
INBND.....	35	in VTRACE statement.....	47	fields of.....	25
INCL.....	32, 43	LOGON keyword		initial value of.....	8
KEY.....	35	description of.....	49	installation of	
LOGON.....	49	LPA.....	10	DOS/VSE.....	24
LPADSN.....	29	altering memory in.....	137	MVS.....	25
LU.....	29, 111, 112, 113, 114	listing directory of.....	10	PASSWORD keyword	
LUname.....	42, 47	refreshing VTEXMAIN.....	84	description of.....	30
MAXBUF.....	29	LPADSN keyword		functions required for.....	30
NDFSUP.....	36, 63, 72	description of.....	29	PC transfers.....(see file transfers)	
NULLS.....	36	LU command		performance	
OF.....	112	description of.....	137	considerations for.....	15, 21
of installer program		LU keyword		personal computers...(see file transfers)	
CICS.....	22	description of.....	29, 111	PF keys	
DIRECTORY.....	21	use of.....	112, 113, 114	use in statistics display.....	58, 67, 69, 77, 96
FSRVFILE.....	22	mapping		use of.....	4
HELPSCRN.....	22	use of.....	4	MAXBUF keyword	
HELPTXT.....	22	description of.....	29	description of.....	29
JUGLFILE.....	22	MDT.(see modified data tag)		messages	
LINES.....	22	appearance of.....	138	description of.....	138
MODE.....	22	modified-data-tag		removal of.....	6
OPSYS.....	23	removal of.....	6	savings by removing.....	57, 65, 75, 94, 95, 105
PRODUCT.....	23	use of.....	4, 6	use of.....	4, 6
VTWOFILE.....	23	NDFSUP keyword		description of.....	36, 63, 72
WNOFILE.....	23	description of.....	36, 63, 72	newcopy	
of STSINST program.....	21	of VTEXMAIN.....	53	NEWCOPY command	
OFS.....	47	description of.....	53, 58, 84	description of.....	53, 58, 84
PASSWORD.....	30	null characters		problems with.....	130
PRT.....	113	problems with.....	130	NULLS keyword	
PSIZE.....	36, 43	NULLS keyword		description of.....	36
RBGEN.....	37, 133	description of.....	36	OF keyword	
RESET.....	37	OF keyword		description of.....	112
ROWCOL.....	47	description of.....	112	use of.....	115
RUCONS.....	37	STATDDN.....	38		
SC.....	113, 118				
SCRSIZE.....	37				
SMFID.....	30				
SMFSVC.....	30				
SMFUSER.....	30				
START.....	48				
STATDDN.....	38				
				PRT keyword	
				use of.....	113

PSIZE keyword  
description of.....36, 43  
in application display 62,  
71, 91

PURGE command  
description of....114, 115

RA sequence  
use of.....5

RBGEN keyword  
description of.....37  
in application display 64,  
73, 93  
problems with.....131  
use of.....133

reading requirements  
for installation .....3  
for operation .....3  
for problems .....3

repetitive characters  
elimination of .....5, 6

request units  
in VTAM.....15, 21  
recommendation for ..15,  
21

RESET command  
description of.....60

RESET keyword  
description of.....37

response time  
factors of.....4  
problem of .....4

right-justified names  
in VLU statements.....42

ROWCOL keyword  
description of.....47

RU size .....(see request units)

RUCONS keyword  
description of.....37  
in application display 64,  
73, 93

savings  
by inbound compression  
.....105  
by outbound  
compression...57, 66, 75,  
95, 105

SBA sequence  
use of.....5

SC keyword  
description of.....113  
use of.....118

screen  
application statistics ..61,  
70, 90  
CICS statistics by LU 79,  
97

TSO terminal statistics56  
VLU table.....68

screen size  
alternate .....32, 44  
primary .....36, 43  
selection of32, 36, 43, 44  
setting default .....40

SCRSIZE keyword  
description of.....37  
in application display.63,  
72, 92

session managers  
considerations for .....41

Shared Virtual Area.....(see  
SVA)

SLIP trace  
activation of.....133

SMF recording  
definition of .....101  
printing of.....101  
specification of .....38  
specifying .....30  
with authorized path...31

SMFID keyword  
description of.....30

SMF SVC keyword  
description of.....30

SMFUSER keyword  
description of.....30

START keyword  
description of.....48

start/stop compression ....(see  
compression)

STATDDN keyword  
description of.....38  
in application display.64,  
73

STATDSN keyword  
description of.....38  
in application display.64,  
73

statistics  
capture and reporting 101  
defining.....38  
display of.....56  
excluded applications 62,  
71, 90  
printing of.....102, 103  
report description.....104  
sequential file attributes  
.....102  
sequential file recording  
.....101  
setting default .....40  
SMF extract .....101  
SMF recording.....101

SMF specification..... 30  
sorting of.....102, 103  
unknown applications61,  
71, 90

statistics display  
APPLICATION field 62,  
71, 90  
ASIZE field ....62, 71, 91  
COMPRESSION field  
.....62, 71, 90  
CPRIMG field 63, 72, 92  
CPU time usage...66, 76,  
95  
DELAY field ..63, 73, 93  
EWIMG field..62, 72, 92  
EXTATTR field...63, 72,  
92  
INBND field...62, 71, 92  
percent compressed  
fields .....57, 66, 76, 95  
PSIZE field .....62, 71, 91  
RBGEN field ..64, 73, 93  
RUCONS field64, 73, 93  
SCRSIZE field63, 72, 92  
send/receive counts.... 65  
STATDDN field...64, 73  
STATDSN field...64, 73  
STATS field ...63, 72, 93  
storage usage ..66, 76, 95  
TRACDDN field ..64, 73  
TRACDSN field...64, 74  
VIO field..... 93

statistics displays  
by application, from  
CICS .....70, 89  
by application, from  
TSO ..... 61  
by terminal, from CICS  
.....79, 97  
by TSO user..... 56  
TSO user..... 56

STATMON keyword  
description of..... 31

STATS keyword  
description of ..... 38  
in application display 63,  
72, 93

STOP keyword  
description of ..... 48

storage usage  
amount of - DOS/VSE  
..... 156  
amount of - MVS..... 155  
controlling ..... 29  
display of .....66, 76, 95

STS0100

description of.....8	use of .....107	user exit
installation of.....9, 16	trace	specifying .....39, 44
link-edit of	activation of....38, 39, 48	VAPPL command
DOS/VSE.....16	global	description of..... 136
STSCORE	activation of.....107,	VAPPL statement
description of.....8	109	description of ..... 31
installation of.....9, 16	conditions for ...107,	keywords of ..... 31
special uses of..... 136	109	relation to VLU..... 43
ST SINST	description of.....106	setting defaults..... 40
description of.....8	information	VECT command
installation of.....9, 16	captured .....107, 109	description of..... 136
keywords of .....21	limitations of .....106	VFTRAN statement
link-edit of	printing of..108, 110	description of ..... 48
DOS/VSE.....16	sequential file	examples of.....49
MVS.....9	attributes.....106	IBM standard..... 50
STSPASS	sequential file	VIO keyword
description of.....8	definition ...106, 108	description of ..... 39
installation of.....9, 16	terminating 108, 110	in application display. 93
link-edit of	types of.....110	VLU command
DOS/VSE.....16	use of .....107, 109	description of ..... 137
MVS.....9	with authorized path	VLU statement.....42, 43
suffix code	.....106	description of ..... 42
of customization table11,	local	keywords of ..... 42
17	activation of.....111,	relation to VAPPL 32, 36
SVA	114	wild-card coding of.... 42
adding VTEXMAIN...20	description of....106,	VMAIN command
altering memory in ...137	111	description of ..... 136
creation of at IPL.....20	display functions116	VOPT command
refreshing VTEXMAIN	display of...112, 116	description of ..... 136
.....100	limitations of .....111	VSKIP statement
SVC keyword	printing of.....113	description of ..... 44
description of.....31	record types of...116	format of ..... 46
SVC number	use of in CICS ...111	keywords of ..... 46
selection of .....30	use of in TSO ....114	methods of ..... 45
SVC routine	purpose of.....106	VSTART statement
for SMF records .....30	specification of38, 39, 48	keywords of; .i.VSTART
TDISP command	TRACE command	statement
description of... 112, 113,	description of....111, 114	description of..... 29
115	transactions	VTAM
terminal input area	exclusion from	application IDs ..... 31
minimum size required	compression.....46	hooking.....52, 59, 87
..... 14, 20	TSO	session managers ..... (see
TEXT keyword	coding rules for VAPPL	session managers)
description of.....46	.....32	using cross-domain... 14,
TO keyword	command screen .....55	19
description of.....113	displaying statistics....56,	VTEXDISP
TRACDDN keyword	61	functions of.....51, 85
description of.....38	executing VTEXPRES55	VTEXMAIN
in application display 64,	subsequent log-ons to .49	newcopy of ..... 53
73	use of authorized path31,	reloading into LPA .... 84
relation to VTRACE...47	106	reloading into SVA.. 100
use of .....107	using local trace.....114	VTEXPRES
TRACDSN keyword	TYPE=INPUT keyword	as TSO command ..... 55
description of.....39	description of .....47	functions of.....50, 55, 85
in application display 64,	UEXIT keyword	operation of..... 51
74	description of .....39, 44	

to activate compression	20
VTEXSKEL	
contents of .....	11, 17
explanation of .....	13, 19
modification of .....	27
VTEXSTAT	
JCL example.....	103, 104
use of .....	101
VTEXTRAC	
JCL example.....	108
use of .....	108, 110
VTRACE statement	
description of.....	47
relation to TRACDDN	
.....	38, 39
wild-cards	
in VLU statements.....	42
WRAP keyword	
description of.....	39, 112,
114	
use of .....	108

