

Virtual Software Systems, Inc.

VPARS and VTAPE
Product Installation Guide

Version 4 Release 2 modification 00



Virtual Software
Systems, Inc.

Preface

Virtual Software Systems, Inc. product installation guide 4.2 at service level 00 and above on 08/04/03.

Copywrite Virtual Software Systems, Inc., 1985-2003.

Copies of this publication may be printed for the internal use of licensed or trial users of products offered by
Virtual Software Systems, Inc.
7715 Browns Bridge Road
Gainesville, Georgia 30506
(770) 781-3200
www.vsoftsys.com

This notice must appear in all copies.

z/VM and IBM are trademarks of International Business Machines Corporation.

Table of Contents

	Chapter 1. Year 2000 compliance	1
	Description.	1
	VM releases supported.	1
	Date format supported.	1
	Installation default.	1
	VSSI Date commands.	2
	Principle of operation.	2
	Online code.	2
	CMS code.	2
	Chapter 2. VSSI Products	3
	VPARS	3
	VTAPE	3
	Chapter 3. Installation Facilities	5
	CNTRL, AUX, and UPDATE Files	5
	Updates to CP Control Files	5
	Updates to CMS Control Files	6
	Naming Conventions for Auxfiles	6
	Naming Conventions for Update Files	6
#	Naming Conventions for Text Files	7
	Assembler Considerations	7
	Chapter 4. The VSSINSTL DEFAULTS File	9
	Chapter 5. The VSSI \$PPF Override File	11
	Chapter 6. Installation Execs	13
	VSAPPEND	15
	VSASMALL	16
#	VSBLDAUX	17
#	VSBLDCTL	18
	VSBLDMAC	19
#	VSBLDNUC	20
#	VSBLDVVT	21
#	VSBLVVTP	22
	VSCMSBLD	23
	VSCOPY	24
#	VSCPBLs	25
	VSEXTRAC	26
	VSGENEXC	27
	VSGENMSG	28
	VSHASM	29
	VSSASM	30
	VSSETUP	31
	VSSIASM	32
	VSSIPREP	33
	VSTAPLOD	34

VSUASM	35
VSUPDT	36
VS1MAC	38
Chapter 7. Distribution Tape Format	39
Chapter 8. VSSI Manuals	41
Manuals on-line	41
Chapter 9. MVS Guest Support	43
MVS UIMs (Unit Information Modules)	43
MVS Utilities	44
Chapter 10. Installation Procedure	45
File Customization	45
VSSI Install (Local) and Config (PARM) Disks	45
Installation Steps	47

#

Chapter 1. Year 2000 compliance

Description.

All VSSI software is YEAR2000 compliant. This means that all VSSI components support a 4 digit year format. The actual format used is defined at installation time and can be modified via CP commands or by configuration file parameters at IPL time.

VM releases supported.

VSSI YEAR2000 compliance is supported in all z/VM releases

Date format supported.

The following date formats are supported by VSSI YEAR2000 compliance. When applicable the corresponding IBM format name is specified in parentheses.

MM/DD/CCYY - (FULLDATE)
DD/MM/CCYY
CCYY/MM/DD - (ISODATE)
CCYY/DD/MM
MM/DD/YY - (SHORTDATE) This is the default at installation.
DD/MM/YY
YY/MM/DD
YY/DD/MM

Installation default.

The installation default is SHORTDATE.

VSSI Date commands.

The following CP commands have been implemented to support the VSSI YEAR2000 compliance: (See the VPARS or VTAPE User's guide for the full command syntax)

- VSSet with the following arguments:
 - VSSet DATEFormat { Anyone of the 8 formats defined above }
 - VSSet { Anyone of the 3 predefined IBM format names }
 - VSSet DELimiter { any character }
- VSQuery with the following argument:
 - VSQuery DATEFormat

Principle of operation.

The default date format is specified during installation by modifying the VSOPTNS Macro.

Online code.

After installation the default date format can be modified by:

- Including the VSI_DATEFormat statement in the VSSI system configuration file. (See the installation guide)
- Issuing the VSSset command with the appropriate parameters.

CMS code.

The default date format, from the VSOPTNS macro, is included in all CMS modules created at installation time. When a CMS module executes, it requests the current date format from the CP using the VSQuery command. If the online date format is available, the CMS module will use it. If the request fails, the CMS module will use the default format.

Chapter 2. VSSI Products

VPARS

VPARS (Virtual Private Active Record Shadowing) for the IBM Transaction Processing Facility (TPF) allows multiple TPF guest virtual machines to share a common TPF system base. Records modified by each guest are maintained on a Private Modified Record (PMR) database for that guest.

VTAPE

VTAPE provides virtual 3420, 3480 and 3490 tape drive simulation and tape data libraries to z/VM guests. Virtual tape drives appear to the guest as real tape drives.

Chapter 3. Installation Facilities

To install VSSI software products, you should understand the facilities that IBM provides for VM product installation and maintenance. These facilities include VM/SES, EXEC files, CNTRL files, macro libraries, the CP nucleus load list, and the CP directory update program.

VSSI provides several front end execs to VM/SES to simplify product installation and to provide capabilities that are not yet available in VM/SES for CP product installations. Execs are provided to:

- Build auxfiles for updates to IBM CP modules.
- Build and update macro libraries.
- Assemble a list of modules.
- Build Version Vector Table entries from an auxfile.

IBM provides the following files that are used and/or updated by the installation process:

- Product Parameter File (PPF)
- CMS control file
- CP control file
- CP load list

IBM files that are updated during installation are placed on a minidisk you allocate for the VSSI installation. This disk is called the VSSI installation or local disk and is accessed when the VSSI PPF is used with VSSETUP or VMFSETUP to access disks. Most of the installation procedure is common to all VSSI products.

CNTRL, AUX, and UPDATE Files

The VSBLDCTL step (called by VSSIPREP) builds new control files and places them on the VSSI local disk. Several entries are added to the control files. The names of the input control files are extracted from your current PPF file. The new control file names are extracted from the VSSI PPF file.

Updates to CP Control Files

The install process will add an entry to the CP control file for each of the following types of updates, depending on the products being installed.

- IBM file updates for VPARS.
- IBM file updates for VTAPE
- IBM file updates that are common to VPARS and VTAPE.
- VPARS file updates.
- VTAPE file updates.
- VPARS and VTAPE common file updates.

Updates to CMS Control Files

The install process will add an entry to the CMS control file for each of the following types of updates, depending on the products being installed.

- VPARS file updates.
- VTAPE file updates.
- VPARS and VTAPE common file updates.

Naming Conventions for Auxfiles

The filetype for VSSI Auxfiles is *AUXcppvr*, where:

c is C for updates to IBM files and A for updates to VSSI files.

pp is a VSSI product identifier (VS, VP, or VT).

vr is the VSSI product version and release number.

For VSSI V4 software the VTAPE AUX filetype for VSSI module RVTSYS is AUXAVT40. The AUX filetype for IBM module HCPIOV is AUXCVT40.

Naming Conventions for Update Files

The filetype for VSSI updates to IBM files is *xvrnnnpp*, where:

x is a prefix letter that matches the first letter of IBM updates to IBM modules. If you are familiar with VMSES, this is the first letter of the :SLVI. value.

vr is the VSSI product version and release number.

nnnn
is an update number.

pp is a VSSI product identifier (VS, VP, or VT).

For VSSI V4 software a sample filetype for a z/VM 4.4.0 VTAPE update to an IBM module is C40001VT.

The filetype for updates to VSSI files is *ppvrnnnn*, where:

pp is a VSSI product identifier (VS, VP, or VT).

vr is the VSSI product version and release number.

nnnn
is an update number.

For VSSI V4 software a sample filetype for an update to a VTAPE module is VT400001.

Naming Conventions for Text Files

CP text files created using the VSSI control file have a file type of *Lcpvrnnn*, where:

c is:

- # • A for 32 bit VSSI modules
- # • C for 32 bit IBM modules

p is the VSSI product identifier (P, S, or T).

vr is the version and release number of the product.

nnn is the update number of the last update applied to the file.

For VSSI V4 software a sample text filetype for VTAPE is LAT40045. The filename corresponds to the VSSI module, such as RVTSYS.

Assembler Considerations

All VSSI source files are distributed in packed format. The VSSI assemble execs automatically unpack the source file to the A-disk for assembly. When the assembly completes, the text file is moved to the VSSI local disk and the source file is erased from the A-disk or repacked if the assemble file was found on the A-disk.

Note: When you assemble HCPDIR, a DIRECTXA module is created. The new DIRECTXA module is moved to the VSSI local disk.

Chapter 4. The VSSINSTL DEFAULTS File

The VSSINSTL DEFAULTS file is used by the installation execs with the VSSI \$PPF override file to control the installation of VSSI products. The VSSINSTL DEFAULTS file on the VSSI distribution tape has the names and locations of the IBM distributed files. If you use different names or move the files, you must update the VSSINSTL DEFAULTS file. This is covered in “Chapter 10. Installation Procedure” on page 45. The :VSPROD keyword identifies the VSSI \$PPF file name.

The installation process uses the :LOCALVSS keyword in the \$PPF override file to identify the VSSI installation disk address. If you want to use a name other than :LOCALVSS for the install, the VSSI local disk keyword can be changed; however, it must be changed in the VSSINSTL DEFAULTS file and the VSSI \$PPF override file.

#

:KEYWORD	ITEM	* COMMENTS
:BASEPROD	ZVM	* IBM or local product id (\$PPFalias)
:DEFCOMP	CP	* Default \$PPF component
:CPREL	310	* CP release z/VM 3.1.0
:VSPROD	VSZVM40	* VSSI product id (\$PPF override filename)
:VSREL	40	* VSSI version and release number
:VPREL	40	* VPARS version and release number
:VTREL	40	* VTAPE version and release number
:VSASM	HASM	* System assembler ASM, HASM or HLASM
:VSLOCAL	LOCALVSS	* VSSIPPF local disk identifier
:VSMACLIB	VSSI40	* VSSI installation maclib filename
:VSMLST	40MAC	* VSSI maclib list suffix
:NUCTARG	MODULE	* CP nucleus destination target
:CPNUCMAP	493	* CPNUCMAP disk address
:VSCPBL	DUAL NODUAL	* Gen 32 and 64 32 only Loadlist
:VSYSMODE	DUAL NODUAL	* Assemble for 32 & 64 32 only

Figure 1. z/VM Release 310 Sample Entries.

Chapter 5. The VSSI \$PPF Override File

The VSSI \$PPF override file is used to override the IBM \$PPF for the installation of VSSI products. The file name of the VSSI \$PPF file is identified by the :VSPROD keyword in the VSSINSTL DEFAULTS file.

z/VM allows multiple \$PPF overrides; the IBM \$PPF override has a filename of ZVM. If you have local \$PPF override files you must change the :CP. and :CMS. tags in the VSZVM40 \$PPF file from ZVM to the filename of your highest level \$PPF override file. You must also change the :BASEPROD tag in the VSSINSTL DEFAULTS file.

You can use any address for the VSSI local disk, that is not used for a different local or IBM required disk. To change the address, change the value after the LOCALVSS keyword in the CP and CMS sections of the VSZVM40 \$PPF file .

If your system procedures require that you change the LOCALVSS \$PPF keyword tag, you must change the value after the :VSLOCAL keyword in the VSSINSTL DEFAULTS file, and the LOCALVSS tag keyword in the VSZVM40 \$PPF override file.

The file name of the VSSI \$PPF file can be changed to any file name. If you do this, you must change the value after the :VSPROD keyword in the VSSINSTL DEFAULTS file.

The following is a sample VSSI \$PPF file for z/VM.

```

# :OVERLST.    CP CMS
*
* CP override section
:CP.          CP ZVM
:CNTRLOP.
:CNTRL.       VSPVM          * VSSI CP control file name
:ALTCNTRL.    VSPVME        * Alternate cntrl. z/VM 3.1.0 only
:DCL.
./INSERT &DELTZ BEFORE
&VSLOC LINK MAINT A310 A310 MR          * VSSI install disk
./END
:EDCL.
:MDA.         UPDATE
./INSERT      LOCALSAM AFTER
LOCALVSS     &VSLOC          * VSSI install disk
./END
:EMDA.
:BLD.
CPLOAD64    VMFBDNUC BUILD7 TXE (NUCTARG MODULE RLDSAVE CNTRL VSPVME
:EBLD.
:DABBV.
LCP TXT LCP00000
LCT TXT LCT00000
LCS TXT LCS00000
LAP TXT LAP00000
LAT TXT LAT00000
LAS TXT LAS00000
:EDABBV
:END.
*
* CMS override section
:CMS.         CMS ZVM
:CNTRLOP.
:CNTRL.       VSSVM          * VSSI CMS control file name
:DCL.
./INSERT &DELTZ BEFORE
&VSLOC LINK MAINT A310 A310 MR          * VSSI install disk
./END
:EDCL.
:MDA.         UPDATE
./INSERT      LOCALSAM AFTER
LOCALVSS     &VSLOC          * VSSI install disk
./END
:EMDA.
:DABBV.
LCP TEXT LCP00000
LCS TEXT LCS00000
LCT TEXT LCT00000
:EDABBV
:END.

```

Figure 2. z/VM Sample \$PPF Override File

Chapter 6. Installation Execs

The execs described in this section are provided to support the installation and maintenance of VSSI software products. The information in this section is presented for reference. For a normal installation, the installation steps in “Chapter 10. Installation Procedure” on page 45 will suffice. Some of these execs are subroutines to other execs, and are listed below, but are not intended to be called directly by the user.

The following execs are provided:

VS\$PPF	Extract values from a compiled PPF.
VSAPPEND	Build a multi-file file for distribution.
VSASMALL	Do all VSSI install assemblies.
VSBLDAUX	Build AUX files for VSSI updates to IBM CP modules.
VSBLDCTL	Update CP and CMS CNTRL files to support VSSI updates.
VSBLDMAC	Build or combine macro libraries.
VSBLDNUC	Issue VMFBLD to build CP nucleus.
# VSNUC64B	Called by VSBLDNUC for z/VM
VSBLDVVT	Build version vector table entries for all VSSI products.
VSBLVVTP	Build version vector table entries for a product or single update.
VSCMSBLD	Build VSSI CMS module files.
VSCOPY	Copy VSSI files to a common user or system disks.
VSCPBLs	Create the CPLOAD list.
VSDFLTS	Extract values from the VSSINSTL DEFAULTS file.
VSEXTRAC	Split the files in a VSAPPEND file.
VSFDISK	Find a disk access mode.
VSFMTBK	Format a block of storage using the definition in a CBMAP file.
VSGENEXC	Update the VSSLTAPE exec.
VSGENMSG	Generate a VSSUME message repository.
VSGETTAG	Extract and stack all records between a :TAG. and an :END.
VSHASM	Update and assemble VSSI CP modules.
VSPRDUCT	Identify the products on the VSSI local disk.

VSSASM	Assemble VSSI CMS programs and build module files.
VSSSETUP	Access disks in CP maintenance order.
VSSIASM	Assemble files without displaying the updates to the terminal.
VSSIDISK	Extract and validate VSSI install disk address from DEFAULTS.
VSSIMSG	Issue messages with console spooled NOTERM during VSSIASM.
VSSIPREP	Execute the initial VSSI install execs.
VSSLTAPE	Provide an interface to VM tape management systems.
VSTAPLOD	Load the VSSI installation files to the VSSI local disk.
VSTXTCPY	Copy text files to the VSSI local disk.
VSUASM	Update and assemble IBM CP modules modified by VSSI.
VSUBLD	Build CPLOAD list and DIRECTXA module.
VSUPDT	Xedit a CP or CMS assemble, macro, or copy file with updates.
VSVMFASM	VSSI interface to VMFHASM and VMFHLASM.
VSVMFLDS	Extracts the filetype of the highest level textfile.
VS1MAC	Update and add a macro to a macro library.
VS40MAC	List of macro and copy files common to VPARS and VTAPE.
VP40MAC	List of VPARS macro and copy files.
VT40MAC	List of VTAPE macro and copy files.

The execs used during installation and execs used for maintenance and error diagnosis are described on the following pages.

VSAPPEND

The VSAPPEND exec will build a file containing other files for sending to an IBMMAIL address or distribution on a floppy disk. The VSEXTRAC exec will split this file into the original files with the correct file types.

VSAPPEND	<i>ifile ofile</i> ([<i>width</i>]
----------	---------------------------------------

ifile is the file specification of the file to be appended (fname ftype fmode).

ofile is the file specification of the append file (fname ftype fmode).

width

is an optional width for the append file. The default is 80 bytes.

Usage Notes:

1. If the append file exists the input file will be appended, if not, the append file will be created.

VSASMALL

The VSASMALL EXEC will use VSSIASM to do all of the assemblies required by the VSSI install.

VSRECMAP	
----------	--

```
/* Execute all VSSI assemblies */  
'VMFCLEAR'  
'VSSIASM CMS'  
'VMFCLEAR'  
'VSSIASM CP '  
'VMFCLEAR'  
'VSSIASM IBM'  
exit
```

VSBLDAUX

The VSBLDAUX exec will build auxfiles for VSSI updates to IBM CP modules.

VSBLDAUX	[<i>product</i> [<i>comp</i>]]
----------	------------------------------------

product

is an optional product identifier (VPARS or VTAPE) that is being installed. *product* defaults to all products that have been loaded to the VSSI install disk.

comp

specifies the component (CP), for which auxfiles should be built. The default is the component named on the :DEFCOMP. keyword in the VSSINSTL DEFAULTS file.

Usage Notes:

1. If the VSSINSTL DEFAULTS file is set up properly, no arguments are required.
 2. If z/VM 3.1.0 or up, the keyword :VSYSMODE. in the VSSINSTL DEFAULT file, will determine whether AUX files for 64 bit modules are created.
 - DUAL will create the AUX files for 64 bit modules.
 - NODUAL will create the AUX files for 32 bit modules only.
- #
- #
- #
- #

VSBLDCTL

The VSBLDCTL EXEC will build an updated CP or CMS control file using the control file name in the previous PPF file. All parameters are optional; an asterisk can be used to bypass a parameter.

VSBLDCTL	[<i>comp</i> [<i>maclib</i> [<i>inctl</i> [<i>outctl</i> [(TEST]]]]]]
----------	--

comp

specifies the component (CP or CMS) for which the control file is being built. The default is the component named on the :DEFCOMP. keyword in the VSSINSTL DEFAULTS file.

maclib

is an optional macro library name. The default is the maclib named in the VSSINSTL DEFAULTS file. This macro library name is added to the other entries on the MACS card in the updated control file.

inctl

is an optional input control file name. The default is the control file named in the VSSINSTL DEFAULTS file. This control file is used as the base for the updates.

outctl

is an optional output control file name. The default is the CP control file named in the VSSI \$PPF override file. This is the name given to the updated control file.

TEST

is an optional keyword that causes an UPDTEST card to be added to the new control file for the creation of new update files.

Usage Notes:

- #
- #
1. If z/VM 3.1.0 or up, the keyword :VSYSMODE. in the VSSINSTL DEFAULT file, will determine whether 1 or 2 CP CTL files are created.
- DUAL will create both the VSPVM and VSPVME control files.
 - NODUAL will only create the VSPVM control file.
- #

VSBLDMAC

The VSBLDMAC exec will add or replace a list of macro and copy files to an existing or new maclib. Updates are applied to the macros before they are added or replaced in the maclib. If no parameters are entered, the default macro library named in the VSSINSTL DEFAULTS file will be built. An asterisk can be entered to bypass any parameter.

VSBLDMAC	[<i>maclib</i> [<i>macexec</i> [<i>cntrl</i> [(LIST [UPDLIST]]]]]
----------	--

maclib

is the name of the maclib to be built or updated. *maclib* defaults to the maclib named in the VSSINSTL DEFAULTS file.

macexec

is the name of an exec file that contains the list of macro and copy files to be added or replaced in the requested maclib. The entries in this exec file must be in the same format as required by the VMFMAC exec (or created by the LISTFILE command with the EXEC option), which is '&1 &2 filename filetype [filemode]'. *macexec* defaults to the lists for the VSSI products installed.

cntrl is the name of a control file to be used to apply updates to the macro and copy files. *cntrl* defaults to the CP control file named in the VSSI \$PPF file.

LIST

causes the names of the macros to be listed on the terminal as they are added to the library. The default is not to list the macro names.

UPDLIST

causes the names of updates to be listed on the terminal as they are applied to macros. The default is not to list the updates.

VSBLDNUC

The VSBLDNUC EXEC will generate a CP nucleus. It spools the virtual reader, printer, and punch devices for a CP nucleus load, and executes the VMFBLD exec using entries from the VSSI \$PPF override file.

VSBLDNUC	[<i>vsprod</i> [<i>comp</i> [<i>loadlist</i>]]] (<i>bldots</i>
----------	---

vsprod

is the VSSI Program Product File (PPF) name, such as VSZVM40. The default is taken from the :VSPROD keyword in the VSSINSTL DEFAULTS file. This is used as the name of the \$PPF override file.

comp

specifies the component (CP) for which the nucleus should be built. The default is the component named on the :DEFCOMP keyword in the VSSINSTL DEFAULTS file.

loadlist

specifies the name of the loadlist to be used to build the nucleus. The default is taken from the VSSI PPF.

The default loadlist(s) used depends on the value of the :VSCPBLs.
keyword in the VSINSTL DEFAULT file.

- # • NODUAL will only use the CPLOAD32 list.
- # • DUAL will use both the CPLOAD32 and CPLOAD64 lists. (This is the
default.)

bldots

are options to be passed to VMFBLD such as; SETUP, NOSETUP, FASTPATH, DUAL or NODUAL

The options "DUAL" or "NODUAL" will override the keyword :VSCPBLs.
in the VSSINSTL DEFAULTS file.

&nucopts.

is the nucleus load deck destination (DISK, PUNCH, TAPE, or MODULE). MODULE is supported in VM/ESA release 2.0 and above. If TAPE is specified, the load deck will be written to tape 182. If you have the Virtual Tape product already installed, this can be a real or a virtual tape. PUNCH is the default. Since this is an option, you must place a left parenthesis and a space before the destination parameter.

This options is not valid for z/VM 3.1.0 and above.

VSBLDVVT

The VSBLDVVT exec will build VVT entries for all VSSI products installed.

VSBLDVVT	<i>systype</i>
----------	----------------

systype

is ALL, CP, CMS or IBM. The default is ALL.

Usage Notes:

1. The VSBLDVVT will erase the VSSI VVT for the requested type before building the new VVT entries.
2. For z/VM 3.1.0 and above, the VVT created will support 32 & 64 bit nucleus modules.

#

VSBLVVTP

The VSBLVVTP exec will build or update VVT entries for a requested product and component or a VVT entry for a requested file.

VSBLVVTP	<i>vsprod comp</i> [ERASE] [(<i>fname</i>)
----------	---

vsprod

is VP, VS or VT. This is a required parameter.

comp

is CP or CMS for VSSI files or IBM for IBM CP files. This is a required parameter.

ERASE

will cause the VVT for the requested component to be erased.

fname

is the name of a file to build an updated VVT entry for.

Usage Notes:

- #
1. For z/VM 3.1.0 and above, the VVT created will support 32 & 64 bit nucleus modules.
- #

VSCMSBLD

The VSCMSBLD exec will build all VSSI CMS modules that contain a specified VSSI CMS textfile.

VSCMSBLD	<i>textfile</i> [<i>product</i>] [MAP]
----------	--

textfile

is the name of a CMS program (textfile) that has been assembled.

product

is the name of a product (VTAPE or VPARS). If a VSSI product name is entered, only module files for that textfile and product are built. The default is to build modules that include the given textfile for all installed products.

MAP

If MAP is entered, a loadmod map is created for the modules built. The default is no map.

Usage Notes:

1. All CMS modules are built using RMODE ANY for 31-bit addressing and are relocatable. The VSCMSBLD exec temporarily sets the LOADAREA to RESPECT while building the modules. If you halt execution of the VSCMSBLD exec, LOADAREA may still be set to RESPECT.

VSCOPY

The VSCOPY exec will copy files from the installation disk to any other CMS disk. If a product name is not given, files of the specified type for all installed products will be copied.

VSCOPY	{ CMS DIR HELP PARM SAMPLES } <i>vdev</i> (<i>prod</i>
--------	---

CMS

will copy the CMS modules and execs intended for public use to the specified disk, which should be accessible to general users (such as the 19E or 319 disk).

DIR or DIRECT or DIRECTXA

will copy the updated DIRECTXA module to the specified disk. If a DIRECTXA MODULE exists on the receiving disk, VSCOPY will rename it to IBMDIREC MODULE. If IBMDIREC MODULE exists on the receiving disk VSCOPY will rename DIRECTXA MODULE to DIRECTXA MODOLD. If DIRECTXA MODOLD exists on the receiving disk, it will be erased.

HELP

will copy the VSSI help files to the specified help disk. You should then rebuild the shared help disk directory.

PARM

will copy the VSSI files which need to reside on the parm disk.

SAMPLES

will copy the products sample configuration files to a user selected disk.

vdev is the virtual device number (minidisk address) that the exec will copy files to. You should resave the CMS nucleus if you copy files to the 19E disk.

prod is VPARS, VTAPE, or ALL. The default is ALL, to copy the specified files for all installed products.

VSCPBL

The VSCPBLS exec creates the CP nucleus loadlist. It will scan HCPMDLAT as well as the VSSI private MDLAT. Once VSSI products are installed, this exec must be used to recreate the CPLOAD list. Assembling HCPLDL will not pick up the VSSI modules.

VSCPBLS	<i>loadlist (options</i>
---------	---------------------------

loadlist
Specifies the name to be given to the loadlist. For release prior to z/VM
3.1.0 the default is CPLOAD.
For z/VM 3.1.0 and above, the default is determined by the keyword
:VSCPBL in the VSSINSTL DEFAULT file.

- DUAL specifies the creation of both the 32 bit (CPLOAD32) and 64 bit (CPLOAD64) load lists.
- NODUAL generates only the 32 bit loadlist (CPLOAD32). This is the current default.

options
Is either 'CP' or 'ALL'. Default is ALL to include private Mdlats.

VSEXTRAC

The VSEXTRAC exec will split a file that was built by the VSAPPEND exec into the original appended files with the correct file attributes.

VSEXTRACT	<i>ifile fmode</i>
-----------	--------------------

ifile is the file specification of the file to be extracted (fname ftype fmode).

fmode

is the file mode of the disk to receive the extracted files.

Usage Notes:

1. If the extracted file exists on the receiving disk, it will be replaced.

VSGENEXC

The VSGENEXC EXEC will build an updated VSSLTAPE exec.

VSGENEXC	
----------	--

Usage Notes:

1. This command updates the VSSLTAPE exec using EXECUPDT, and copies the updated exec to the VSSI install disk.

VSGENMSG

The VSGENMSG EXEC will the VSSI CMS message repository (VSSUME) text file.

VSGENMSG	<i>repos</i> <i>cntrl</i> (UPDLIST
----------	-------------------------------------

repos

is the filename of the CMS message repository to be updated and generated. The default is VSSUME.

cntrl is the filename of the CNTRL file to be used for updates. The default is the CNTRL file in the VSSI PPF.

UPDLIST

will cause the names of the update files applied to the repository to be listed on the terminal.

VSHASM

The VSHASM EXEC will assemble VSSI CP modules. The EXEC can assemble the CP modules for all products installed, the CP module for a single product, or a single CP module. This exec is normally invoked by the VSSIASM exec.

VSHASM	[<i>product</i>] [<i>options</i>]
--------	---------------------------------------

product

is the name of a VSSI product (VPARS or VTAPE) or the name of a single VSSI CP module to be assembled, such as RVPCCW. The default is to assemble all modules for all installed products.

options

are optional parameters to be passed to VMFHASM.

VSSASM

The VSSASM EXEC will assemble and build module files for the CMS programs of the products that are being installed. It can also be used to assemble and build a single module, or all modules that include a CMS text file. This exec is normally invoked by the VSSIASM exec.

VSSASM	[<i>vsprod</i> <i>textfile</i>] <i>asmopts</i> (<i>loadopts</i>)
--------	--

vsprod

is the name of a VSSI product (VPARS or VTAPE) for which all CMS modules should be assembled and built. The default is to assemble all textfiles for all installed products.

textfile

is the name of a single file which should be assembled. All modules that include the requested textfile will be built.

asmopts

are optional parameters to be passed to VMFASM or VMFHASM , with one exception. The option 'OUTMode m' will store the resulting text file(s) and module(s) on the Mdisk accessed as file mode m, overriding the PPF definition for the install disk.

loadopts

are options to be passed to the CMS loader.

Usage Notes:

1. If no parameters are entered, all textfiles for all installed products will be assembled, and all product modules will be built.
2. The assembler to be used, VMFASM, VMFHASM or VMFHLASM, will be taken from the value after the :VSASM tag in the VSSINSTL DEFAULTS file.
3. The CMS modules are built using RMODE ANY for 31-bit addressing and are relocatable. The VSSASM exec temporarily sets the LOADAREA to RESPECT while building the modules. If you halt execution of the VSSASM exec, LOADAREA may still be set to RESPECT.

VSSETUP

The VSSETUP EXEC will execute the VMFSETUP exec using the VSSI PPF file, to access the disks for the installation.

VSSETUP	<i>vsprod comp</i> (NOPPF)
---------	----------------------------

vsprod

is the name of a PPF to be used for the execution of VMFSETUP. The default is the name on the :VSPROD tag in the VSSINSTL DEFAULTS file.

comp

is the name of a component to be used for the execution of VMFSETUP. The default is the component specified on the :DEFCOMP tag in the VSSINSTL DEFAULTS file.

NOPPF

requests that the VMFPPF exec should not be executed to build a compiled VSSI PPF. The default is to build a new PPF.

VSSIASM

The VSSIASM EXEC will assemble CP, CMS, or updated IBM assemble files.

VSSIASM	<i>option</i> [<i>vsprod</i> <i>module</i>] <i>asmopts</i>
---------	--

option

is CP, CMS, or IBM, to assemble CP, CMS, or updated IBM assembler file(s).

vsprod

is the name of a product (VPARS or VTAPE) for which all CP, CMS, or updated IBM assembler files should be assembled.

module

is the name of a single module which should be assembled.

asmopts

are optional parameters to be passed to VMFASM or VMFHASM.

Usage Notes:

1. If only CP, CMS, or IBM is entered, all modules of that type for all installed products will be assembled. The IBM modules which have VSSI updates are determined from the presence of auxfiles.
2. The assembler to be used, VMFASM or VMFHASM, will be taken from the value after the :VSASM keyword in the VSSINSTL DEFAULTS file.
3. The VSSIASM exec will start console spooling if it is not started. Prior to calling the requested assemble exec the console spool file is closed and the console is spooled NOTERM. As each module is assembled, its name is displayed on the terminal. If any errors occur during the update or assembly, the error return code is displayed. When all files are assembled, the console is spooled TERM and the console spool file is closed. If there were any errors during the assemblies, this console file can be reviewed to evaluate and correct the error.
4. The VSSI assemble execs use the IBM VMFHASM exec to do the update and assembly of each module. VMFHASM sets EMSG OFF during the update and assembly. If you halt execution during the assembly process, you should 'SPOOL CONSOLE TERM' and 'SET EMSG ON'.

VSSIPREP

The VSSIPREP exec does the initial preparation for the installation of VPARS and/or VTAPE.

VSSIPREP	
----------	--

Usage Notes:

1. VSSIPREP executes the VSBLDCTL, VSBLDAUX, VSBLDEXC, VSGENMSG, VSBLDMAC execs and for VM/ESA 2.0 and above, the VSBLDVVT exec.
2. The VSSETUP or VMFSETUP exec must be run, to access the disks for the install, before the VSSIPREP exec is run.

VSTAPLOD

The VSTAPLOD exec loads the VSSI files from the distribution tape to the VSSI install disk.

VSTAPLOD	<i>product</i> <i>vdev</i>
----------	----------------------------

product

is the product code for your release of VM:

VM Release:	Product code:
z/VM 3.1.0	ZVM310
z/VM 4.2.0	ZVM420
z/VM 4.3.0	ZVM430
z/VM 4.4.0	ZVM440

Additional VM releases may be announced before this manual is reprinted. If so, the product codes should be predictable. If a cover letter accompanying the tape lists new product codes, use the product codes from the cover letter.

vdev is the virtual device number (address) of the disk you have defined and formatted for this installation of VPARS and/or VTAPE.

Usage Notes:

1. The disk to which you load the products should not contain any files. All VSSI installation tapes are refresh tapes that contain all files for the product(s) ordered.

VSUASM

The VSUASM EXEC will assemble CP assemble files that are updated by VPARS or VTAPE. This exec is normally invoked by the VSSIASM exec.

VSUASM	[<i>vsprod</i> <i>module</i>] <i>asmopts</i>
--------	--

vsprod

is the name of a VSSI product (VPARS or VTAPE) for which all updated IBM CP modules should be assembled. The default is to assemble all updated IBM CP modules for all installed products.

module

is the name of a single module which should be assembled.

asmopts

are optional parameters to be passed to VMFASM or VMFHASM.

Usage Notes:

1. If no parameters are entered, all updated IBM modules for all installed VSSI products will be assembled. The IBM modules which have VSSI updates are determined from the presence of auxfiles.
2. The assembler to be used, VMFASM, VMFHASM or VMFHLASM will be taken from the value after the :VSASM keyword in the VSSINSTL DEFAULTS file.

VSUPDT

The VSUPDT exec will XEDIT a CP or CMS assemble, macro, or copy file in update mode. The VSSI CP or CMS control file controls the application of updates. See the usage notes for an explanation of the parameter parsing for this exec.

VSUPDT	<i>fname</i> [<i>ftype</i>] [<i>fmode</i>] [<i>sid</i>] [<i>until</i>] (<i>vsprod comp</i>
--------	---

fname

is the name of the file to update.

ftype is the filetype of the file to update.

fmode

is the filemode of the file to update.

sid is the sidcode identifier to be used for the update. XEDIT will place the sidcode in columns 64-71 of the update.

until is an optional "until" update identifier. All updates through the "until" update are applied; updates listed in the auxfile after the "until" update are not applied.

vsprod

The VSSI product name (VSESA40, etc.). This is used as the name of the \$PPF override file in which to look for the component.

comp

The component name (CP, CMS). The *vsprod* and *comp* are used to determine the name of the control file to control the application of updates to the base file.

Usage Notes:

1. VSUPDT is a convenient tool to allow you to look at VSSI or IBM assembler source files, in XEDIT, with updates included. If you make changes to a file while in XEDIT, they will be saved in an update file. You do not need to use the VSUPDT exec for normal installation activities.
2. The parameters are partly positional and partly keyword-based. If the second parameter is not a recognized filetype, it is taken as a sidcode, and the filetype is determined by searching for a file with the given name and an updatable filetype (ASSEMBLE, COPY, XEDIT, EXEC, etc.). If the third parameter is one or two characters, it is taken as a filemode; if it is three or more characters, it is taken as an "until" identifier. The filemode cannot

be specified without the filetype. Most of the parameters can be omitted in the majority of cases, and the exec will determine what is intended.

3. You can use VSBLDCTL with the TEST option to create control files with an UPDTEST entry. VSUPDT will then use UPDTEST as the identifier for any new update applied to a module.

VS1MAC

The VS1MAC EXEC will update a macro or copy file, and add it to or replace it in a maclib.

VS1MAC	[<i>fname</i> [<i>maclib</i> [<i>cntrl</i>]]]
--------	---

fname

is the file name of a macro or copy file.

maclib

is the name of a macro library to be updated. It defaults to the macro named on the :VSMACLIB entry in the VSSINSTL DEFAULTS file.

cntrl

is the optional name of a control file to be used for the update. It defaults to the control file named in the VSSI \$PPF file for the default component.

Chapter 7. Distribution Tape Format

The VSSI distribution tape has 30 tape files per system release, plus a leading file containing the VSTAPLOD exec. For each release, the first 10 tape files contain files that are common across all products. The next 10 tape files are for VPARS files. The last 10 tape files are for VTAPE files. If you are not installing both products, the files for the product not installed will contain a dummy file.

For each release:

- Files 1 and 2 contain installation files.
- File 3 contains VSSI CP update files.
- File 4 contains VSSI assemble files.
- File 5 contains VSSI macro and copy files.
- File 6 contains VSSI AUX and UPDATE files.
- File 7 contains VSSI manuals in PDF and listing formats.
- File 8 contains CMS message help files.
- File 9 contains additional installation files.
- File 10 is reserved for future common files.
- File 11 contains VPARS exec and sample files.
- File 12 contains VPARS updates for CP segments.
- File 13 contains VPARS assemble files.
- File 14 contains VPARS macro and copy files.
- File 15 contains AUX and UPDATE files for VPARS and PTV.
- File 16 contains VPARS manuals in Script and listing formats.
- File 17 contains VPARS help files.
- Files 18, 19 and 20 are reserved for additional VPARS files.
- File 21 contains VTAPE exec and sample files.
- File 22 contains VTAPE updates for CP segments.
- File 23 contains VTAPE assemble files.
- File 24 contains VTAPE macro and copy files.
- File 25 contains AUX and UPDATE files for VTAPE.
- File 26 contains VTAPE manuals in Script and listing formats.
- File 27 contains VTAPE help files.
- Files 28, 29 and 30 are reserved for additional VTAPE files.

Chapter 8. VSSI Manuals

The VSSI manuals were created by IBM DCF/Script, using the PostScript¹ option 'DEVICE(PSA)', and the output files were then converted to PDF format using Adobe Acrobat. They are formatted to be printed two sided but can also be printed on a single sided printer

Manuals on-line

The VSSI documentation is available on-line, in two formats:

- HTML format suitable for browser viewing.
- PDF format, suitable for downloading and subsequent printing using the free ACROBAT READER program from Adobe Systems, Inc.

To access the on-line documentation, point your browser to <http://www.vsoftsys.com/doc/vssidoc.htm>

The documentation is also included with the distribution files in two formats:

- LISTING files formatted for a 1403 or 3211-type line printer. (The listing files can also be printed on a 3800 printer)
- PDF files. These files must be downloaded to a PC and can then be viewed or printed, using the free ACROBAT READER from Adobe Systems, Inc.

¹ PostScript and Adobe are registered trademarks of Adobe Systems, Inc.

Chapter 9. MVS Guest Support

MVS UIMs (Unit Information Modules)

To fully utilize VTAPE with MVS guests and have coexistence of real and virtual drives, it is necessary that virtual tape drives allocation be unique. If this is not done, a virtual tape mount could be requested on a real tape drive and a real tape mount could be requested on a virtual tape drive.

VSSI supplies UIMs for MVSCP (MVS Configuration Program) and MVS/HCD (MVS Hardware Configuration Definition). These UIMs provide unique esoteric device names for virtual tape drives to make them allocation and catalog unique. The device class and type bytes in the UCBTYP field are the same as the corresponding real devices. A bit is set in the UCBTYP field option byte to make virtual drives allocation and catalog unique. The following esoteric device names and their corresponding virtual tape define types are supported.

•	Esoteric	Define	Description
•	342V	V3420	3420 Model 8 dual density
•	348V	V3480	3480 without IDRC
•	348X	V3480	3480 with IDRC support
•	349V	V3490	3490 with IDRC support

The type of tape drive used does not affect the recording of data in the tape library. A tape written on any drive type can be read by any other drive type. The different drive types do provide different responses and support different functions. The 3480/3490 load display function, used with autolib mode, will cause tapes to be automatically mounted on MVS virtual tape drives.

The files to install the MVSCP and HCD UIMs are loaded to the VSSI local disk during installation. The files for MVSCP are:

- CBPUC005 ASSEMBLE
- CBPUC005 ASMJCL
- CBPUC005 LINKJCL

The files for HCD are:

- CBDUC005 ASSEMBLE
- CBDUC005 ASMJCL
- CBDUC005 LINKJCL
- CBDEC005 ASSEMBLE
- CBDEC005 ASMJCL
- CBDEC005 LINKJCL

To install the UIMs, you must send these files to your MVS system. The SYSLMOD statement in the link edit JCL is set to a temporary dataset. A SYSLMOD statement with the correct system library DSNAMES is in the

link edit JCL as a comment. After the UIMs have been assembled and link edited, an I/O generation is required for MVSCP, for HCD the virtual tape drives can be defined using the HCD panels.

MVS Utilities

VSSI provides an MVS program and CMS exec that can be used to uncatalog virtual tapes that are about to be scratched, because they have reached their expiration date. The MVS program is VTUNCTLG and the CMS exec is VTUNCAT. The files for the MVS program are:

- VTUNCTLG ASSEMBLE
- VTUNCTLG ASMJCL
- VTUNCTLG LINKJCL

Daily, after the VTRPTS exec is run, the VTUNCAT exec can be run to build an MVS uncatalog job. The VTUNCAT exec reads the scratch file created by VTRPT1 and verifies that the scratch file is current and matches the tapes in the library. The exec then builds an MVS job to execute the VTUNCTLG program. The VTUNCTLG program uses CAMLST to retrieve the catalog entry for each tape to be scratched. If the dataset is cataloged to the tape about to be scratched, CAMLST is used to uncatalog the dataset. The uncatalog job must be run after the daily scratch file is created and before the VTSCR1 exec is run to scratch the tapes. If VTSCR1 is run before the uncatalog job, a job could be started that requests the tape after it has been scratched.

Chapter 10. Installation Procedure

Before starting this installation, you should be familiar with the material in the User's Guide and Reference for the products being installed.

File Customization

See “Chapter 4. The VSSINSTL DEFAULTS File” on page 9 for information about the VSSINSTL DEFAULTS file, and “Chapter 5. The VSSI \$PPF Override File” on page 11 for information about the VSSI \$PPF override file. You may want to customize these two files after loading them to the VSSI installation disk.

VSSI Install (Local) and Config (PARM) Disks

You must define a minidisk to contain the VSSI product files for installation and can define a minidisk to contain the VSSI product configuration files. These minidisks can be defined in Maint's directory or a new directory entry such as VSALLOC and linked with write access by MAINT. They are not required to be allocated on CP Residence volumes, they can be allocated on any CP volume containing minidisks that is available during IPL.

The advantage of using a separate directory entry for the VSSI disks is that multiple versions of the VSSI product and configuration files can be maintained and linked as needed.

The size of the installation disk will depend on the products being installed. Each product (VPARS and VTAPE) will require approximately 20 cylinders on a 3390 formatted in 4K blocks. If you are installing both VPARS and VTAPE, you will need 40 cylinders of 3390 space or equivalent.

If you plan to download the VSSI install VMARC file to the Installation disk and keep it there, you should add 10 cylinders to the size of the disk.

You can move the VMARC file to a virtual or real tape and VMARC UNPACK the file directly to the installation disk from the tape

The virtual device number (address) of the VSSI LOCAL disk is identified in the VSSI \$PPF override using the :LOCALVSS keyword tag.

If you do not use the default address shown below, you will need to update all occurrences of :LOCALVSS in the VSSI \$PPF file after you have loaded the files to the VSSI LOCAL disk.

The default minidisk addresses in the VSSI \$PPF override file are:

VM Release:	minidisk address:
z/VM 3.1.0	A310
z/VM 4.2.0	A420
z/VM 4.3.0	A430
z/VM 4.4.0	A440

You can define a minidisk in Maint's or another directory such as VSALLOC to contain the VTAPE and/or VPARS configuration files. If you use a separate minidisk for the VSSI configuration files the system parm disk does not have to be released for updates to the VSSI files. This also makes it easier to migrate to future releases of VM because normally the VSSI user configuration files do not change from one VM release to the next.

The virtual device number and owning userid of the VSSI PARM disk will be specified in the VSSI CONFIG file, on the system IPL PARM disk. (The default is the system IPL PARM disk)

VTAPE only users should only need a 1 cylinder PARM disk to contain the VTSYSTEM defaults file. VPARS/VTAPE users may need more space depending on the number of VPARS configuration files maintained. You can use any device number for the VSSI PARM disk, VSSI uses device numbers such as ACF3, BCF3, etc. for different levels of configuration files.

Installation Steps

1. Logon to MAINT.
2. If you defined the VSSI LOCAL and PARM disks in a directory entry other than MAINT and did not place link statements in MAINT's PROFILE EXEC or directory entry then link the VSSI LOCAL and PARM disks. Replace *vdev1* with the device number assigned in the VSALLOC directory entry and *vdev2* with the device number required by MAINT.

vdev1 for the installation (LOCAL) disk might be A440, B440, etc. and *vdev2* would be A440 for Z/VM 4.4.0.

vdev3 is device number for the VSSI PARM disk that will contain the VSSI configuration files and be accessed by the system.

```
LINK VSALLOC vdev1 vdev2 MR
LINK VSALLOC vdev3 vdev3 MR
```

3. Format the installation minidisk.

```
FORMAT vdev2 E
```

Reply 1 to the format request and *VSvdev* to the disk label request. Where *vdev* is the virtual device number of the VSSI LOCAL disk.

4. Format the PARM minidisk.

```
FORMAT vdev3 W
```

Reply 1 to the format request and *VSvdev* to the disk label request. Where *vdev* is the virtual device number of the VSSI PARM disk.

5. If you downloaded the install VMARC file from our FTP site, use VMARC to expand the installation files to the VSSI local disk.

```
VMARC UNPACK ifn VMARC ifm = = E ( NOTRace
```

If you moved the VMARC file to tape you can use.

```
VMARC UNPACK = = E ( TAP1 NOTRace
```

6. If you are loading the install files from tape continue with step 7. otherwise skip to step 9. on page 48

7. Attach a tape drive at virtual address 181 and mount the distribution tape. Load the first file of the distribution tape to the VSSI disk:

```
ACCESS vdev E
VMFPLC2 LOAD * * E
```

The VSTAPLOD EXEC will be loaded.

8. The VSTAPLOD exec with the *VM product code* and the device number of the VSSI local disk will load the installation files to the VSSI local disk. The product codes are:

VM Release:	Product code:
z/VM 3.1.0	ZVM310
z/VM 4.2.0	ZVM420
z/VM 4.3.0	ZVM430
z/VM 4.4.0	ZVM440

Additional VM releases may be announced before this manual is reprinted. If so, the product codes should be predictable.

Execute the VSTAPLOD exec to load the installation files. The tape will be positioned to the files for your release of VM, and the installation files will be loaded to the VSSI local disk.

VSTAPLOD *product vdev*

9. Edit the VSSINSTL DEFAULTS file and check the name after the :BASEPROD keyword tag. This should be the filename of the PPF override file that you currently use to build the CP NUCLEUS. If you do not have a local PPF override file or a PPF override file for another installed product, the default \$PPF override filename for z/VM is ZVM.

If the \$PPF name does not match the name used at your installation, change the name in the VSSINSTL DEFAULTS file.

Check the value on the :CPREL keyword tag, it should be the VM release that you are installing for.

Other entries in the VSSINSTL DEFAULTS file are described in “Chapter 4. The VSSINSTL DEFAULTS File” on page 9.

10. Make a note of the name after the :VSPROD keyword tag. This is the filename of the VSSI \$PPF override file used during the installation. The filetype of the override file is \$PPF.
11. If you did not use the default device number for your VSSI local disk, edit the VSSI \$PPF override file and change the device number on all occurrences of :LOCALVSS to match the device number used.

Other entries in the \$PPF override file are described in “Chapter 5. The VSSI \$PPF Override File” on page 11 and in IBM's product service manual for your release of VM.

12. Access the disk that contains IBM's installation and service tools (VMSES). For most installations, this will be MAINT's 5E5 disk.

```
ACCESS 5E5 B
```

13. Access the 51D VMSES disk as your D-disk.

```
ACCESS 51D D
```

14. If the High Level Assembler is not available on your 190, 19E or 51D disks, Access the disk containing the Assembler as your C-disk.

```
ACCESS vdev C
```

15. If the VSSI local disk is not already accessed, it should be accessed as your E-disk.

```
ACCESS vdev E
```

16. Use VSSETUP to access the disks for the installation. The VSSETUP exec will build the VSSI compiled PPF files and use VMFSETUP to access the disks for the installation.

```
VSSETUP
```

After the initial VSSETUP, if no changes have been made to any \$PPF override files applied before the VSSI override, you can use VMFSETUP using the VSSI PPF file name or VSSETUP with the NOPPF option to access the disks.

17. If you want to change the default dateformat, you must edit the VSOPTNS MACRO on the VSSI local disk. (See “Chapter 1. Year 2000 compliance” on page 1 for valid formats)

```
XEDIT VSOPTNS MACRO E
Replace the default value of MM/DD/YY.
FILE
```

18. Skip this step if you are not installing VPARS. VPARS uses a VMDUSER field as an anchor for a VPARS control block. The default VPARS user field is VMDUSER8. If the VPARS default conflicts with an other product, you can edit the VPOPTNS MACRO on the VSSI local disk.

```
XEDIT VPOPTNS MACRO E
Change the default value of 8 with a non-conflicting
number from 1 to 7
FILE
```

19. Execute the VSSIPREP exec to build the CNTRL, AUX and VVT files and build the VSSI macro library. This step will take approximately 1 minute.

```
VSSIPREP
```

20. This step will create the CP nucleus loadlist. You can refer to the VSCPBLs EXEC description to determine what loadlist(s) are created.

```
#
#
```

```
VSCPBLs
```

21. The next steps will assemble the VSSI CP and CMS modules and the IBM CP modules that are updated by VSSI. A log of the results of each assemble is maintained on the VSSI install disk in VSSIASM LOG.

A description of the update for each IBM CP module modified by VSSI is contained in the VPARS VMMODS or VTAPE VMMODS files on the installation disk.

There are three ways to execute the assemblies.

- a. You can use the VSASMALL exec. VSASMALL will use the VSSIASM exec to do all of the required assemblies. If there are any errors in the update and assembly process, an error code will be displayed on the console following the name of the module that failed. At the completion of VSASMALL you can edit the VSSIASM LOG file to determine if any of the assemblies failed. If an assembly fails an asterisk is added to the log record following the return code (RC=xx). A simple locate for * will identify any failed assemblies.

If any assemblies failed you can peek the console file in your reader to determine the problem. If no updates or assemblies failed continue with Step 26. on page 51 If any updates or assemblies failed, go to step 25. on page 51

- b. You can use the VSSIASM exec. When you use VSSIASM your console will be spooled NOTERM. The name of each module will be displayed when it is assembled. If there are any errors in the update or assembly process, an error code will be displayed following the name of the module that failed. When VSSIASM completes, you can edit the VSSIASM LOG file to determine if any errors occurred.
- c. Using the individual assembly execs (VSHASM, VSUASM and VSSASM). If you use these execs, your console will not be spooled NOTERM and all update and assembly messages will be displayed. All update and assembly messages are always written to the console log file.

22. Use VSSIASM or VSHASM to assemble the VSSI modules that will be included in the CP nucleus. This step will take approximately 10 minutes.

VSSIASM CP *or* VSHASM

23. Use VSSIASM or VSSASM to assemble the VSSI CMS modules. This step will take approximately 7 minutes.

VSSIASM CMS *or* VSSASM

24. Use VSSIASM or VSUASM to assemble the IBM CP modules that are modified by VSSI. If you are installing VTAPE, HCPDIR is assembled and is used to build a new DIRECTXA module. This step will take approximately 10 minutes.

```
VSSIASM IBM or VSUASM
```

25. If there are any errors in the updates or assemblies, correct the error and reassemble the failing module, or contact Virtual Software Systems. You may reassemble any failing module with VSSIASM:

```
VSSIASM CP fname or VSHASM fname  
VSSIASM CMS fname or VSSASM fname  
VSSIASM IBM fname or VSUASM fname
```

where *fname* is the name of the file to be assembled.

26. Before continuing with the next step, make sure that you have a good backup of your CP nucleus and a tested procedure for restoring it.
27. Execute the VSBLDNUC exec to build a CP nucleus on Maint's 493.

```
# VSBLDNUC
```

- ```
#
```
- Which modules (CPLOAD32 and/or CPLOAD64) are created is based on the value of the keyword :VSCPBLs. in the VSSINSTL DEFAULT file.
  - DUAL will create both modules. (This is the default)
- ```
#
```
- When Dual is used the CPLOAD MODULE created can be IPLed on 32 or 64 bit processors.
- ```
#
```
- NODUAL will only create CPLOAD32 module and a CPLOAD MODULE that can be IPLed on a 32 bit processor only.
- ```
#
```

28. If you did not receive any errors on the nucleus build, edit the CPLOAD64 and CPLOAD32 MAP files on Maint's 394 and do a locate for Undefined Symbols and Unresolved references.

```
XEDIT CPLOAD32 MAP  
LOCate /UNRES  
LOCate /UNDEF  
QUIT
```

29. If any Unresolved References or Undefined Symbols are found, correct the problem and rebuild the nucleus.

30. To place the new nucleus into production you must copy the CPLOAD MODULE to your system or alternate PARM disk, normally Maint's CF1 or CF2.

Use CP Query CPDisks to determine if CP has the parm disk accessed. Normally CF1 is accessed as CPs A-disk.

```
Query CPDisks
```

Use CPRELeas to release CPs access of the disk

```
CPREL A
```

Use CP Query Virtual *vdev* to determine if Maint has Write access to the parm disk.

```
Query Virtual CF1
```

If maint does not have write access, use the CP LINK command to acquire write access.

```
LINK * CF1 as CF1 MR
```

Access the parm disk as W, save the current CPLOAD MODULE as CPLOLD for recovery and copy the CPLOAD MODULE to the parm disk.

```
ACCESS CF1 W
ERASE CPLOLD MODULE W
REName CPLOAD MODULE W CPLOLD = =
COPY CPLOAD MODULE L = = W (OLDDate
```

The CP nucleus portion of the installation process is complete. After completing the CMS steps you will be instructed to shutdown and re-ipl CP.

31. This step will perform the following:
- a. Copy the VSSI CMS modules to a common disk accessible to the users. (Usually MAINT 19E)
 - b. Copy the VSSI HELP files to the help disk. (Usually MAINT 19D)
 - c. Copy the VSSI config files to the IPL parm disk (Usually MAINT CF1)
 - d. Copy the new DIRECTXA module to the CMS RES disk. (Usually MAINT 190) (*VTAPE only*)
 - e. Copy the VSSI sample configuration files to a user defined or system PARM disk.

Note1: The minidisk addresses specified for user MAINT are the usual definitions. If your installation uses different disks, you will need to modify the VSCOPY commands.

Note2: If you copy the files to the system disks, you must rebuild the CMS save system after the copies. You should also rebuild the help saved segment information on saving CMS and the HELP segments is in the z/VM Service Guide.

Use VSCOPY to copy the CMS module files to a common disk:

```
VSCOPY CMS 19E
```

Use VSCOPY to copy the HELP files to a common disk:

```
VSCOPY HELP 19D
```

Use VSCOPY to copy the VSSI config file to the CP parameter disk.

```
VSCOPY PARM CF1
```

If you are installing Virtual Tape, use VSCOPY to copy the DIRECTXA module to a system disk:

```
VSCOPY DIR 190 (VTAPE only)
```

When you copy the directory update module to Maint's 190, the IBM version will be saved as IBMDIREC MODULE. The most recent VSSI or IBM version will also be renamed to DIRECTXA MODOLD.

Optionally copy the sample configuration files to a user defined or the system PARM disk. You will review and update these files during Post-Installation Customization of VTAPE and/or VPARS.

```
VSCOPY SAMPLES vdev
```

32. Edit and review the VSSI CONFIG file which was copied to the system PARM disk by a previous VSCOPY step. Pay particular attention to the VSI_DISK statement. It defines the CP accessed minidisk where you will later store the VSSI products (VPARS/VTAPE) configuration files, such as VTSYSTEM DEFAULTS, VPSYSTEM DEFAULTS, and VPARS configuration files.

Note: You can place all of the VPARS/VTAPE configuration files on the system PARM disk, however, VSSI recommends that separate user disk be used.

If the system PARM disk is still accessed as W edit the VSSI CONFIG file. If not, reaccess the PARM disk as Maint's W-disk.

```
XEDIT VSSI CONFIG W
```

Review the following statements:

```
VSI_DISK MAINT CF1 (Change if using a different owner/mdisk)
/*CP_ACCESS disk_owner disk_add fmode*/(Uncomment as needed)
VSI_DATEFormat (Use only if different from VSOPTNS macro)
VSI_DELIMiter (Use only if different from VSOPTNS macro)
```

33. Edit the SYSTEM CONFIG file the system PARM disk and add the following statements, following the Logo_Config statement.

```
XEDIT SYSTEM CONFIG W
LOCate /Logo_Config
=
I /****/
I /* VSSI_Config */
I /****/
I IMBED VSSI CONFIG
FILE
```

34. You should have a minimum of 4 VSSI files on your system PARM disk. You can do a FILEL ** W to ensure these files are present.

- VSSI CONFIG
- VSSI COMMANDS
- VPARS COMMANDS
- VTAPE COMMANDS

35. Resave the CMS and/or HELP saved segments if you copied the VSSI files to the system disks.

36. Schedule a VM SHUTDOWN and Re-IPL to load the new CP nucleus.

```
SHUTDOWN REIPL
```

37. The base components of the VSSI product(s) are now installed on your system. You will now prepare the products for use by general users.

Note: Once VPARS and/or VTAPE are installed on your system, you must use VSBLDNUC or VMFBLD with the VSSI PPF file each time you rebuild your CP nucleus. This ensures that the updated CP loadlist is used to include VPARS and/or VTAPE in the system.

If you apply corrective service to VM you should run the VSSI install procedure starting with step 16. on page 49 This will ensure that new service is included in the nucleus.

Note: To customize the VSSI product(s) being installed, refer to the Post-Installation Customization section of the VPARS and/or VTAPE User's Guide and Reference.

Reader's Comment Form

We welcome your comments on this manual. Is there anything you especially like or dislike about the organization or presentation? Possible topics for comments include clarity, accuracy, completeness, and specific errors and omissions.

If you would like a reply, please include your name and address:

VSSI Product Installation Guide, release 4.2, printed on 08/04/03.

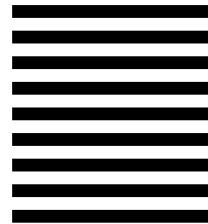
Thank you for your comments. A postage stamp is not necessary if you are mailing this from the U.S.A. You may also fax this page to (770) 781-3210.

Cut along line

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT No. 107 Gainesville, GA

POSTAGE WILL BE PAID BY ADDRESSEE

Virtual Software Systems, Inc.
7715 Browns Bridge Road
Gainesville, GA 30506-9867



Fold and tape

Reader's Comment Form

We welcome your comments on this manual. Is there anything you especially like or dislike about the organization or presentation? Possible topics for comments include clarity, accuracy, completeness, and specific errors and omissions.

If you would like a reply, please include your name and address:

VSSI Product Installation Guide, release 4.2, printed on 08/04/03.

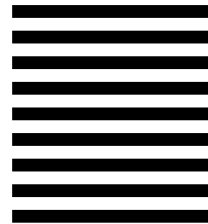
Thank you for your comments. A postage stamp is not necessary if you are mailing this from the U.S.A. You may also fax this page to (770) 781-3210.

Cut along line

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



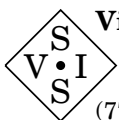
BUSINESS REPLY MAIL
FIRST CLASS PERMIT No. 107 Gainesville, GA

POSTAGE WILL BE PAID BY ADDRESSEE

Virtual Software Systems, Inc.
7715 Browns Bridge Road
Gainesville, GA 30506-9867



Fold and tape



Virtual Software Systems, Inc.

7715 Browns Bridge Road
Gainesville, Georgia 30506

(770) 781-3200 Fax (770) 781-3210