Program Directory for

IBM XL C/C++ for z/VM

version 1 release 2.0

Program Number 5654-A22

for Use with
z/VM V5R4.0

Document Date: September 2008

GI11-2899-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 47.

This program directory, dated September 2008, applies to the IBM® XL C/C++ for z/VM™ Compiler version 1 release 2.0, Program Number 5654-A22.

A form for reader's comments appears at the back of this publication. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

1.0 Introduction

This program directory is intended for the system programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of the XL C/C++ for z/VM Compiler. You should read all of this program directory before installing the program and then keep it for future reference.

The program directory contains the following sections:

- 2.0, “Program Materials” on page 2 identifies the basic and optional program materials and documentation for XL C/C++ for z/VM Compiler.
- 3.0, “Program Support” on page 6 describes the IBM support available for XL C/C++ for z/VM Compiler.
- 4.0, “Program and Service Level Information” on page 7 lists the APARs (program level) and PTFs (service level) incorporated into this release of XL C/C++ for z/VM Compiler.
- 5.0, “Installation Requirements and Considerations” on page 8 identifies the resources and considerations for installing and using the XL C/C++ for z/VM Compiler.
- 6.0, “Installation Instructions” on page 11 provides detailed installation instructions for the XL C/C++ for z/VM Compiler.
- 7.0, “Service Instructions” on page 26 provides detailed service instructions for XL C/C++ for z/VM Compiler.
- 8.0, “Define and Build The XL C/C++ for z/VM Compiler Saved Segment” on page 37 provides instructions on how to install XL C/C++ for z/VM Compiler in Saved Segments.
- Appendix A, “Overriding the Product Parameter File (PPF)” on page 44 describes how to override the Product Parameter File for XL C/C++ for z/VM Compiler.
- Appendix B, “Segment Build Lists (CCNBLSGL)” on page 46 describes the build list for Saved Segments for XL C/C++ for z/VM Compiler.

2.0 Program Materials

An IBM program is identified by a program number. The program number for IBM XL C/C++ for z/VM version 1 release 2.0 is 5654-A22.

The program announcement materials describe the features supported by XL C/C++ for z/VM Compiler. Ask your IBM marketing representative for this information if you have not already received a copy.

The following sections identify:

- basic and optional program materials available with this program.
- publications useful during installation

2.1 Basic Machine-Readable Material

The distribution medium for this program is 3590 or 3592 tape cartridge. You can also receive this program electronically by ordering it through the z/VM SDO (System Delivery Offering) using IBM ShopzSeries. For more information about IBM ShopzSeries go to **www**.ibm.com/software/ShopzSeries.

The cartridge or electronic envelope contains all the programs and data needed for installation. See 6.0, “Installation Instructions” on page 11 for more information about how to install the program. Figure 1 describes the physical tape cartridge. Figure 2 describes the file content of the program tape cartridge or electronic envelope.

Figure 1. Basic Material: Program Tape

Feature Number	Medium	Physical Volume	Tape Content	External Tape Label
6016	3590 cart.	1	XL C/C++ for z/VM Compiler, V1R2	XL C/C++ for z/VM
6017	3592 cart.	1	XL C/C++ for z/VM Compiler, V1R2	XL C/C++ for z/VM

Figure 2 (Page 1 of 2). Program Tape: File Content

Tape File	Content
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists

Figure 2 (Page 2 of 2). Program Tape: File Content

Tape File	Content
6	PTFPARTs
7	XL C/C++ for z/VM Compiler Service
8	XL C/C++ for z/VM Compiler Service
9	XL C/C++ for z/VM Compiler Base Code
10	XL C/C++ for z/VM Compiler Samples
11	XL C/C++ for z/VM Compiler Build Code
12	XL C/C++ for z/VM Compiler Build Code (English)
13	XL C/C++ for z/VM Compiler Build Code (Kanji)

2.2 Optional Machine-Readable Material

There are no optional machine-readable materials for XL C/C++ for z/VM Compiler.

2.3 Program Publications

The following sections identify the basic and other publications for XL C/C++ for z/VM Compiler.

These publications are supplied softcopy as part of the *IBM Online Library: z/VM Collection* DVD (SK5T-7054) or CD-ROM (SK2T-2067) in BookManager® and Adobe® Portable Document Format (PDF). One copy of the z/VM collection on DVD was included when you ordered the basic materials for z/VM.

In addition, the XL C/C++ for z/VM Compiler softcopy publications are available in Adobe Portable Document Format from the z/VM Collection link off of the z/VM library home page on the World Wide Web; the URL for this home page is:

http://www.**ibm.com**/servers/eserver/zseries/zvm/library

The XL C/C++ for z/VM Compiler publications can be ordered separately, for a fee, using the specific publication number through the IBM Publication Center at:

http://www.**ibm.com**/shop/publications/order/

The Publications Center is a world wide central repository for IBM product publications and marketing material. Furthermore, a large number of publications are available online in various file formats (such as Adobe PDF), which can currently be downloaded free of charge.

2.3.1 Basic Program Publications

One copy of the following publication is included when you order the basic materials for XL C/C++ for z/VM Compiler.

Figure 3. Basic Material: Unlicensed Publications

Publication Title	Form Number
XL C/C++ for z/VM: User's Guide	SC09-7625

2.3.2 Other Program Publications

The following publications may be needed to support XL C/C++ for z/VM Compiler.

Figure 4. Other Material: Unlicensed Publications

Publication Title	Form Number
XL C/C++ for z/VM: Runtime Library Reference	SC09-7624
z/OS® XL C/C++ Language Reference	SC09-4815
z/OS XL C/C++ Messages	GC09-4819
z/OS XL C/C++ User's Guide	SC09-4767
z/OS XL C/C++ Compiler and Run-Time Migration Guide	GC09-4913
z/OS XL C/C++ Programming Guide	SC09-4765
z/OS C Curses	SA22-7820
z/OS XL C/C++ Run-Time Library Reference	SA22-7821
z/OS Language Environment® Debugging Guide	GA22-7560
z/OS Language Environment Run-Time Messages	SA22-7566

2.4 Program Source Materials

No program source materials or viewable program listings are provided for XL C/C++ for z/VM Compiler.

2.5 Publications Useful During Installation

The publications listed in Figure 5 may be useful during the installation of XL C/C++ for z/VM Compiler. To order copies, contact your IBM representative.

Figure 5 (Page 1 of 2). Publications Useful During Installation / Service on z/VM V5R4.0

Publication Title	Form Number
z/VM: VMSES/E Introduction and Reference	GC24-6130

Figure 5 (Page 2 of 2). Publications Useful During Installation / Service on z/VM V5R4.0

Publication Title	Form Number
z/VM: Service Guide	GC24-6117
z/VM: CMS Commands and Utilities Reference	SC24-6073
z/VM: CMS File Pool Planning, Administration, and Operation	SC24-6074
z/VM: Other Components Messages and Codes	GC24-6120
z/VM: CMS and REXX/VM Messages and Codes	GC24-6118
z/VM: CP Messages and Codes	GC24-6119

3.0 Program Support

This section describes the IBM support available for XL C/C++ for z/VM Compiler.

3.1 Preventive Service Planning

Before installing XL C/C++ for z/VM Compiler, check with your IBM Support Center or IBMLink™ (ServiceLink) to see whether there is additional Preventive Service Planning (PSP) information. To obtain this information, specify the following UPGRADE and SUBSET values:

Figure 6. PSP Upgrade and Subset ID

RETAIN®			
COMPID	**Release**	**Upgrade**	**Subset**
5654A2200	120	CCZVM120	CMS120

3.2 Statement of Support Procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Figure 7 identifies the component ID (COMPID), Retain Release and Field Engineering Service Number (FESN) for XL C/C++ for z/VM Compiler.

Figure 7. Component IDs

RETAIN			
COMPID	**Release**	**Component Name**	**FESN**
5654A2200	120	XL C/C++ for z/VM Compiler, V1R2	0400000

4.0 Program and Service Level Information

This section identifies the program and any relevant service levels of XL C/C++ for z/VM Compiler. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs shipped with this product. Information about the cumulative service tape is also provided.

4.1 Program Level Information

The APAR fixes against previous releases of XL C/C++ for z/VM Compiler that have been incorporated into this release are as follows:

```
PQ80009   PK01039   PK07442   PK25540   PK35093   PK44672
```

4.2 Service Level Information

This is the initial release of XL C/C++ for z/VM Compiler. There are no PTFs on the installation media.

4.3 Cumulative Service Tape

Cumulative service for XL C/C++ for z/VM Compiler is available through a monthly corrective service tape, Expanded Service Option (ESO).

The XL C/C++ for z/VM Compiler PRODID, which is needed to order an ESO, is 5654A22B.

5.0 Installation Requirements and Considerations

The following sections identify the system requirements for installing and activating the XL C/C++ for z/VM Compiler.

5.1 Hardware Requirements

There are no special hardware requirements for the XL C/C++ for z/VM Compiler.

5.2 Program Considerations

The following sections list the programming considerations for installing the XL C/C++ for z/VM Compiler and activating its functions.

5.2.1 Operating System Requirements

The XL C/C++ for z/VM Compiler operates under the following system environment (or subsequent releases):

- z/VM V5R4.0

5.2.2 Other Program Product Requirements

Language Environment for z/VM, which is included in the base of z/VM, includes the C runtime library.

The High Level Assembler is required if you wish to customize the XL C/C++ for z/VM Compiler compile options (see 6.4.1, "Customization of Compile-time Options (Optional)" on page 22).

5.2.3 Program Installation Considerations

This section describes items that should be considered before you install the XL C/C++ for z/VM Compiler.

- VMSES/E is required to install this product.
- If multiple users install and maintain licensed products on your system, there might be a problem getting the necessary access to MAINT's 51D disk (software inventory). If you find that there is contention for write access to the 51D disk, you can eliminate it by converting the Software Inventory from minidisk to Shared File System (SFS). See the *VMSES/E Introduction and Reference* manual, section 'Changing the Software Inventory to an SFS Directory', for information on how to make this change.
- Customers no longer install the XL C/C++ for z/VM Compiler strictly using the MAINT user ID, but will use a new user ID, 5654A22B. This is the IBM suggested user ID name. Customers are free to change this to any user ID name they wish, however, a PPF override must be created.

Note: It may be easier to make changes to the PPF during the installation procedure 6.2.1, “Plan Your Installation For XL C/C++ for z/VM Compiler” step 6 on page 14, rather than after you have installed this product.

5.3 DASD Storage and User ID Requirements

Figure 8 lists the user IDs and minidisks that are used to install the XL C/C++ for z/VM Compiler.

Important Installation Notes:

- User ID(s) and minidisks will be defined in 6.2.1, “Plan Your Installation For XL C/C++ for z/VM Compiler” on page 12 and are listed here so that you can get an idea of the resources that you will need prior to allocating them.
- 5654A22B is a default user ID and can be changed. If you choose to change the name of this user ID you need to create a Product Parameter File (PPF) Override with the new user ID. This can be done in 6.2.1, “Plan Your Installation For XL C/C++ for z/VM Compiler” step 6 on page 14.

  Note: If you choose to install the XL C/C++ for z/VM Compiler on a common user ID the default minidisk addresses for the XL C/C++ for z/VM Compiler may already be in use. If any of the default minidisk addresses required by the XL C/C++ for z/VM Compiler are already in use you will need to define different minidisk addresses for the XL C/C++ for z/VM Compiler minidisks and create a PPF override to change the default minidisk addresses to the new addresses you have defined.

Figure 8 (Page 1 of 2). DASD Storage Requirements for Target Minidisks

Minidisk owner (user ID)	**Default Address**	**Storage in Cylinders**		**FB-512 Blocks**	**SFS 4K Blocks**	**Usage**
		DASD	**CYLS**			**Default SFS Directory Name**
5654A22B	2B2	3390	250	360000	45000	Contains all the base code shipped with XL C/C++ for z/VM Compiler **VMSYS:5654A22B.CCXX.OBJECT**
5654A22B	2C2	3390	5	7200	900	Contains customization files. This disk may also be used for local customer modification. **VMSYS:5654A22B.CCXX.SAMPLE**
5654A22B	2D2	3390	450	648000	81000	Contains serviced files **VMSYS:5654A22B.CCXX.DELTA**

Note: Cylinder values defined in this table are based on a 4K block size. A total of 1290 cyls are needed for minidisk install. FB-512 block and SFS values are derived from the 3390 cylinder values in this table. 232,200 4K blocks are needed for SFS install.

Figure 8 (Page 2 of 2). DASD Storage Requirements for Target Minidisks

Minidisk owner (user ID)	**Default Address**	**Storage in Cylinders**		**FB-512 Blocks**	**SFS 4K Blocks**	**Usage**
		DASD	**CYLS**			**Default SFS Directory Name**
5654A22B	2A6	3390	5	7200	900	Contains AUX files and software inventory tables that represent the test service level of XL C/C++ for z/VM Compiler **VMSYS:5654A22B.CCXX.APPLYALT**
5654A22B	2A2	3390	5	7200	900	Contains AUX files and software inventory tables that represent the service level of XL C/C++ for z/VM Compiler that is currently in production. **VMSYS:5654A22B.CCXX.APPLYPROD**
5654A22B	29E	3390	450	648000	81000	Test build disk. This code will be copied to a production disk, (such as MAINT 19E) so the production disk will also require this amount of free space. **VMSYS:5654A22B.CCXX.TBUILD**
5654A22B	191	3390	125	180000	22500	5654A22B user ID's 191 minidisk **VMSYS:5654A22B.**

Note: Cylinder values defined in this table are based on a 4K block size. A total of 1290 cyls are needed for minidisk install. FB-512 block and SFS values are derived from the 3390 cylinder values in this table. 232,200 4K blocks are needed for SFS install.

6.0 Installation Instructions

This chapter describes the installation methods and the step-by-step procedures to install and activate the XL C/C++ for z/VM Compiler.

The step-by-step procedures are in two column format. The steps to be performed are in bold large numbers. Commands for these steps are on the left hand side of the page in bold print. Additional information for a command may exist to the right of the command.

Each step of the installation instructions must be followed. Do not skip any step unless directed otherwise.

Throughout these instructions, the use of IBM-supplied default minidisk addresses and user IDs is assumed. If you use different user IDs or minidisk addresses to install the XL C/C++ for z/VM Compiler, adapt these instructions as needed for your environment.

Note!

The sample console output presented throughout these instructions was produced on a z/VM V5R4.0 system. If you are installing XL C/C++ for z/VM Compiler on a different z/VM system, the results obtained for some commands might differ from those depicted here.

6.1 VMSES/E Installation Process Overview

The following is a brief description of the main steps in installing the XL C/C++ for z/VM Compiler using VMSES/E.

- Plan Your Installation

  The VMFINS command is used to load several VMSES/E files from the product tape and to obtain the XL C/C++ for z/VM Compiler resource requirements.

- Allocate Resources

  The information obtained from the previous step is used to define the appropriate user IDs and minidisks (or SFS directories) needed to install and use the XL C/C++ for z/VM Compiler.

- Install the XL C/C++ for z/VM Compiler Product

  The VMFINS command is used to load the XL C/C++ for z/VM Compiler product files from tape to the test BUILD and BASE minidisks/directories. VMFINS is then used to update the VM SYSBLDS file used by VMSES/E for software inventory management.

- Place the XL C/C++ for z/VM Compiler Files into Production

  Once the operation of the XL C/C++ for z/VM Compiler is satisfactory, the product files are copied from the test BUILD disk(s) to production BUILD.

- Perform Post Installation Tasks

Information about file tailoring is presented in 6.6, “Post-Installation Considerations” on page 25

For a complete description of all VMSES/E installation options refer to:

- *z/VM: VMSES/E Introduction and Reference*

6.2 Installation of the XL C/C++ for z/VM Compiler with VMSES/E (VMFINS)

VMFINS will be used to install the XL C/C++ for z/VM Compiler. VMFINS is the installation aid supplied as part of VMSES/E to make installation of Licensed Program Products (LPs) consistent.

For a complete description of all VMFINS installation options refer to *VMSES/E Introduction and Reference*.

6.2.1 Plan Your Installation For XL C/C++ for z/VM Compiler

The VMFINS command will be used to plan the installation. This is a two step process that will:

- load the first tape file, containing installation files
- generate a 'PLANINFO' file listing
  - all user ID/minidisk requirements
  - required products

To obtain planning information for your environment:

1 Log on to a user ID, such as MAINT, to plan your XL C/C++ for z/VM Compiler installation.

This user ID can be any ID that has read access to MAINT's 5E5 minidisk and write access to MAINT's 51D minidisk. If you plan to install into SFS this user ID must have SFS administration authority.

2 Mount the XL C/C++ for z/VM Compiler installation tape and attach it to the user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be attached at virtual address 181. If you have a product envelope SERVLINK file make sure it is available on the A-disk or any work disk that is accessed as file mode C.

3 Establish read access to the VMSES/E code.

link MAINT 5e5 5e5 rr
access 5e5 b

The 5E5 disk is where VMSES/E resides.

4 Establish write access to the Software Inventory disk.

link MAINT 51d 51d mr
access 51d d

The MAINT 51D disk is where the VMSES/E system-level Software Inventory and other dependent files reside.

Note: If another user already has the MAINT 51D minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you need to have the other user re-link the 51D in read-only mode (RR). You will then need to re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

5 Load the XL C/C++ for z/VM Compiler specific files to the 51D minidisk. The VMFINS INFO command will perform the following:

- load Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

a If installing from **tape**

vmfins install info (nomemo

The NOMEMO option will load the memos from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

b If installing from a product **envelope** file

vmfins install info (nomemo env *envfilename*

envfilename is the file name of the product envelope file. The file type must be SERVLINK.

The NOMEMO option will load the memos from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

```
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

6 Obtain resource planning information for the XL C/C++ for z/VM Compiler. The product will **not** be loaded by the VMFINS command at this time.

The PLAN option indicates that you want VMFINS to perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

Note:

- If you change the PPF name, a default user ID, or other parameters using a PPF override, you'll need to use your changed values instead of those indicated (when appropriate) throughout the rest of the installation instructions. For example, you'll need to specify your PPF override file name instead of 5654A22B for VMSES/E commands that use the PPF keyword.
- For more information about overriding the PPF, for example to change the VMSYS file pool name at initial install time, refer to Chapter 3 in the *z/VM: VMSES/E Introduction and Reference*.

  If you need override the PPF at a later time, refer to Appendix A, “Overriding the Product Parameter File (PPF)” on page 44.
- Use ONE of the following install components:
  - **CCXX** if installing on minidisks.
  - **CCXXSFS** if installing in SFS directories.
  - **CCXXK** if installing Kanji on minidisks.
  - **CCXXKSFS** if installing Kanji in SFS directories.

a If installing from **tape**

vmfins install ppf 5654A22B *{CCXX* | *CCXXSFS* | *CCXXK* | *CCXXKSFS}* **(plan nomemo**

You can override any of the following by typing in a *1* followed by a *0*:

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

If no override is needed just type in a *0* to continue.

b If installing from a product **envelope** file

vmfins install ppf 5654A22B *{CCXX* | *CCXXSFS* | *CCXXK* | *CCXXKSFS}* **(plan nomemo env** *envfilename*

envfilename is the file name of the product envelope file. The file type must be SERVLINK.

You can override any of the following by typing in a *1* followed by a *0*:

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

If no override is needed just type in a *0* to continue.

```
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5654A22B CCXX :PRODID
            5654A22B%CCXX?
            Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX
VMFREQ2805I Product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX has passed
            requisite checking
VMFINT2603I Planning for the installation of product :PPF 5654A22B CCXX :PRODID
            5654A22B%CCXX
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFINS2760I VMFINS processing completed successfully
Ready;
```

7 Review the install message log ($VMFINS $MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see the appropriate *z/VM: Messages and Codes*, or use on-line HELP.

vmfview install

6.3 Allocate Resources for Installing the XL C/C++ for z/VM Compiler

Note

Ensure that MAINT's 19E is large enough to hold the complete product should you decide to put the product there.

Use the planning information in the 5654A22B PLANINFO file, created in the **PLAN** step, to:

- Create the 5654A22B user directory entry for minidisk install

  OR

- Create the 5654A22B user directory entry for SFS install

6.3.1 Preparing to install XL C/C++ for z/VM Compiler on Minidisk

1 Obtain the user directory information from the 5654A22B PLANINFO file.

Note: The user directory entry is located at the bottom of the PLANINFO file of the resource section. This entry contains all of the links and privilege classes required for the 5654A22B user ID. Use the directory entry found in PLANINFO as model input to your system user directory.

2 Add the MDISK statements to the directory entry for 5654A22B. Use Figure 8 on page 9 to obtain the minidisk requirements.

3 Add the 5654A22B directory entry to the system user directory. Change the password for 5654A22B from xxxxx to a valid password, in accordance with your security guidelines.

4 Place the new directory on-line using the DIRECTXA command or an equivalent directory maintenance product, such as DIRMAINT.

Note

All minidisks for the 5654A22B user ID must be formatted before installing XL C/C++ for z/VM Compiler.

5 Make sure minidisks or SFS directories containing previous releases of the compiler are not accessed by the 5654A22B user ID.

6 Continue with section 6.4, “Install the XL C/C++ for z/VM Compiler” on page 19.

6.3.2 Preparing to install XL C/C++ for z/VM Compiler in SFS Directories

1 Obtain the user directory information from the 5654A22B PLANINFO file.

Note: The user directory entry is located at the bottom of the PLANINFO file of the resource section. This entry contains all of the links and privilege classes required for the 5654A22B user ID. Use the directory entry found in PLANINFO as model input to your system user directory.

2 Add the 5654A22B directory entry to the system user directory. Change the password for 5654A22B from xxxxx to a valid password, in accordance with your security guidelines.

3 If you intend to use an SFS directory as the work space for the 5654A22B user ID, include the following IPL control statement in the 5654A22B directory entry:

```
IPL CMS PARM FILEPOOL VMSYS
```

This will cause CMS to automatically access the 5654A22B top directory (VMSYS:5654A22B) as file mode A.

4 Place the new directory online using the DIRECTXA command or an equivalent directory maintenance product, such as DIRMAINT.

5 An SFS install also requires the following steps:

a Determine the number of 4k blocks that are required for SFS directories by adding up the 4K blocks required for each SFS directory you plan to use.

If you intend to use all of the default XL C/C++ for z/VM Compiler SFS directories, the 4K block requirements for each directory are summarized in Figure 8 on page 9.

This information is used when enrolling the 5654A22B to the VMSYS filepool.

b Enroll user 5654A22B in the VMSYS filepool using the ENROLL USER command:

ENROLL USER 5654A22B VMSYS (BLOCKS *blocks*

where *blocks* is the number of 4k blocks that you calculated in the previous step.

This creates the user's top directory, VMSYS:5654A22B.

Note: This must be done from a user ID that is an administrator for VMSYS: filepool.

c Determine if there are enough blocks available in the filepool to install XL C/C++ for z/VM Compiler. This information can be obtained from the QUERY FILEPOOL STORGRP command. The output from this command is a list of storage groups in the filepool and the number of 4K blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install XL C/C++ for z/VM Compiler you need to add space to the filepool. See *z/VM: CMS File Pool Planning, Administration, and Operation* manual for information on adding space to a filepool.

d Create the necessary subdirectories listed in the 5654A22B PLANINFO file using the CREATE DIRECTORY command.

A complete list of default CCXX SFS directory names is provided in Figure 8 on page 9.

set filepool vmsys:
create directory *dirid*

dirid is the name of the SFS directory you're creating, such as:

```
create directory vmsys:5654A22B.CCXX
create directory vmsys:5654A22B.CCXX.object
   :
```

If necessary, see *z/VM: CMS Commands and Utilities Reference* for more information about the CREATE DIRECTORY command.

e Authorize the MAINT user ID (or equivalent user ID that will own the production code disk for XL C/C++ for z/VM Compiler) to READ the test code directory, using the GRANT AUTHORITY command.

grant auth vmsys:5654A22B.CCXX.TBUILD to MAINT (read newread

If necessary, see the *z/VM: CMS Commands and Utilities Reference* manual for more information about the GRANT AUTHORITY command.

f Make sure minidisks or SFS directories containing previous releases of the compiler are not accessed by the 5654A22B user ID.

6.4 Install the XL C/C++ for z/VM Compiler

The *ppfname* used throughout these installation instructions is **5654A22B**, which assumes you are using the PPF supplied by IBM for XL C/C++ for z/VM Compiler. If you have your own PPF override file for XL C/C++ for z/VM Compiler, you should use your file's *ppfname* instead of **5654A22B**. The *ppfname* you use should be used **throughout** the rest of this procedure.

The *compname* used throughout these installation instructions is ONE of the following.

- **CCXX** if installing on minidisks.
- **CCXXSFS** if installing in SFS directories.
- **CCXXK** if installing Kanji on minidisks.
- **CCXXKSFS** if installing Kanji in SFS directories.

You should use the *compname* that you installed with **throughout** this procedure.

1 Logon to the installation user ID **5654A22B** which should have the minimum virtual storage of 256M.

If you have not formatted the minidisks for the 5654A22B user ID they you need to do that before continuing.

2 Create a PROFILE EXEC that will contain the ACCESS commands for VMSES/E and Software Inventory disks, MAINT 5E5 and 51D minidisks.

```
xedit profile exec a
===> input /**/
===> input 'access 5e5 b'
===> input 'access 51d d'
===> input 'set ldrtbls 30'
===> file
```

3 Execute the profile to access MAINT's minidisks.

```
profile
```

4 Establish write access to the Software Inventory disk, if it is not already linked R/W.

Note: If the MAINT 51D minidisk was accessed R/O, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

link MAINT 51d 51d mr
access 51d d

5 Have the XL C/C++ for z/VM Compiler installation tape mounted and attached to 5654A22B at virtual address 181. The VMFINS EXEC requires the tape drive to be attached at virtual address 181. If you have a product envelope SERVLINK file make sure it is available on the A-disk or any work disk that is accessed as file mode C.

6 Install the XL C/C++ for z/VM Compiler.

Note:

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the 'Installing Products with VMFINS' and "Install Scenarios' chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

a If installing from **tape**

vmfins install ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(nomemo nolink**

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

b If installing from a product **envelope** file

vmfins install ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(nomemo nolink env** *envfilename*

envfilename is the file name of the product envelope file. The file type must be SERVLINK.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5654A22B CCXX :PRODID
            5654A22B%CCXX?
            Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX
VMFREQ2805I Product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX has passed
            requisite checking
VMFINT2603I Installing product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX
VMFSET2760I VMFSETUP processing started for 5654A22B CCXX
VMFUTL2205I Minidisk|Directory Assignments:
            String    Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E      R/W  2C2   CXX2C2
VMFUTL2205I APPLY     F      R/W  2A6   CXX2A6
VMFUTL2205I           G      R/W  2A2   CXX2A2
VMFUTL2205I DELTA     H      R/W  2D2   CXX2D2
VMFUTL2205I BUILD0    I      R/W  29E   CXX29E
VMFUTL2205I BASE1     J      R/W  2B2   CXX2B2
VMFUTL2205I --------  A      R/W  191   CXX191
VMFUTL2205I --------  B      R/O  5E5   MNT5E5
VMFUTL2205I --------  D      R/W  51D   MNT51D
VMFUTL2205I --------  S      R/O  190   CMS20
VMFUTL2205I --------  Y/S    R/O  19E   YDISK
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS ENVELOPE 0300
VMFREC1851I (1 of 8) VMFRCAXL processing AXLIST
VMFRCX2159I Loading 0 part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 8) VMFRCPTF processing PARTLST
VMFRCP2159I Loading 0 part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 8) VMFRCCOM processing DELTA
VMFRCC2159I Loading 0 part(s) to DELTA 2D2 (H)
VMFREC1851I (4 of 8) VMFRCALL processing APPLY
VMFRCA2159I Loading part(s) to APPLY 2A6 (F)
VMFRCA2159I Loaded 1 part(s) to APPLY 2A6 (F)
VMFREC1851I (5 of 8) VMFRCALL processing BASE
VMFRCA2159I Loading part(s) to BASE1 2B2 (J)
VMFRCA2159I Loaded 65 part(s) to BASE1 2B2 (J)
VMFREC1851I (6 of 8) VMFRCALL processing SAMPLE
VMFRCA2159I Loading part(s) to LOCALSAM 2C2 (E)
VMFRCA2159I Loaded 178 part(s) to LOCALSAM 2C2 (E)
VMFREC1851I (7 of 8) VMFRCALL processing BUILD
VMFRCA2159I Loading part(s) to BUILD0 29E (I)
VMFRCA2159I Loaded 25 part(s) to BUILD0 29E (I)
VMFREC1851I (8 of 8) VMFRCALL processing BUILDENG
VMFRCA2159I Loading part(s) to BUILD0 29E (I)
VMFRCA2159I Loaded 6 part(s) to BUILD0 29E (I)
VMFREC2760I VMFREC processing completed successfully
VMFINT2603I Product installed
VMFINS2760I VMFINS processing completed successfully
Ready;
```

Figure 9. Sample console output from vmfins install

7 Review the install message log ($VMFINS $MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see the appropriate *z/VM: Messages and Codes*, or use on-line HELP.

vmfview install

Note: You may detach the tape drive now if you wish.

6.4.1 Customization of Compile-time Options (Optional)

Execute the following command to customize the defaults for C/C++ compiler options.

Note: You must have access to the High Level Assembler.

C5654A22 *ppfname compname*

The C5654A22 EXEC uses a default *ppfname* of 5654A22B and a default *compname* of CCXX. If you are using a different *ppfname* or *compname*, then those names must be entered.

This exec will XEDIT an assembler routine CCNCOPT, allow you to enter and save your modifications, and use VMSES/E to call the High Level Assembler to assemble your changes. If you don't make any modifications in XEDIT, then no further processing takes place. You can change the IBM supplied default *FLAG(I)* or add any valid XL C/C++ for z/VM Compiler options, as described in the *XL C/C++ for z/VM User's Guide*. For example, to modify the defaults to be equivalent to entering *cc HELLO (source optimize flag(E)* you could make the following assembler code changes:

```
CCNO00C  CSECT
CCNO00C  RMODE ANY
CCNO00C  AMODE ANY
* You can either code the length in the fullword below, or
* null terminate the string with an X'00' as shown
         DC    F'0'
         DC    CL7'SOURCE '      << notice blank delimiter
         DC    CL9'OPTIMIZE '    << notice blank delimiter
         DC    CL7'FLAG(E)'
         DC    X'00'             (NULL-terminate the string)
         END
```

Figure 10. CCNCOPT assembler routine

```
VMFREP2760I VMFREPL processing started
VMFREP2509I The version vector table 5654A22B VVTLCL E will be updated for the
            part CCNCOPT ASM
VMFREP2760I VMFREPL processing completed successfully
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 2) VMFBDCOM processing CCNBLSAM EXC11801 H, target is LOCALSAM
            2C2 (E)
VMFBDC2219I Processing object =.ASSEMBLE
VMFBLD1851I (1 of 2) VMFBDCOM completed with return code 0
VMFBLD1851I (2 of 2) VMFBDCOM processing CCNBLCPY EXEC J, target is BUILD0 29E
            (I)
VMFBDC2219I Processing object CCNCOPT.ASSEMBLE
VMFBLD1851I (2 of 2) VMFBDCOM completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
VMFREP2760I VMFREPL processing started
VMFREP2509I The version vector table 5654A22B VVTLCL E will be updated for the
            part CCNCOPT TXT
VMFREP2760I VMFREPL processing completed successfully
VMFASM2760I VMFHLASM processing started
DMSUPD181E No update files were found
VMFASM1907I Assembling CCNCOPT
 Assembler Done No Statements Flagged
VMFASM2507I CCNCOPT TXTL1010 created on your E-disk for use in a VMSES/E
            environment
VMFASM2760I VMFHLASM processing completed successfully
PRT FILE 1028 SENT FROM SESELPS3 PRT WAS 1028 RECS 0137 CPY  001 A NOHOLD NOKEEP
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 2) VMFBDCOM processing CCNBLSAM EXC11801 H, target is LOCALSAM
            2C2 (E)
VMFBDC2219I Processing object =.ASSEMBLE
VMFBLD1851I (1 of 2) VMFBDCOM completed with return code 0
VMFBLD1851I (2 of 2) VMFBDCOM processing CCNBLCPY EXEC J, target is BUILD0 29E
            (I)
VMFBDC2219I Processing object CCNCOPT.ASSEMBLE
VMFBDC2219I Processing object CCNCOPT.TEXT
VMFBLD1851I (2 of 2) VMFBDCOM completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
Ready;
```

Figure 11. Sample console output from the C5654A22 customization exec

6.4.2 Update Build Status Table for the XL C/C++ for z/VM Compiler

1 Update the VM SYSBLDS software inventory file for XL C/C++ for z/VM Compiler.

vmfins build ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(serviced nolink**

```
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2603I Processing product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX
VMFREQ2805I Product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX has passed
            requisite checking
VMFINB2603I Building product :PPF 5654A22B CCXX :PRODID 5654A22B%CCXX
VMFSET2760I VMFSETUP processing started for 5654A22B CCXX
VMFUTL2205I Minidisk|Directory Assignments:
            String    Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E      R/W  2C2   CXX2C2
VMFUTL2205I APPLY     F      R/W  2A6   CXX2A6
VMFUTL2205I           G      R/W  2A2   CXX2A2
VMFUTL2205I DELTA     H      R/W  2D2   CXX2D2
VMFUTL2205I BUILD0    I      R/W  29E   CXX29E
VMFUTL2205I BASE1     J      R/W  2B2   CXX2B2
VMFUTL2205I --------  A      R/W  191   CXX191
VMFUTL2205I --------  B      R/O  5E5   MNT5E5
VMFUTL2205I --------  C      R/O  11A5  MJD1A5
VMFUTL2205I --------  D      R/W  51D   MNT51D
VMFUTL2205I --------  S      R/O  190   CMS20
VMFUTL2205I --------  Y/S    R/O  19E   YDISK
VMFSET2760I VMFSETUP processing completed successfully
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I No new build requirements identified
VMFBLD2179I There are no build requirements matching your request at this time.
            No objects will be built
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
VMFINB2603I Product built
VMFINB2173I Executing verification exec V5654A22
*** V5654A22: Installation Verification Beginning...

XL C/C++ for z/VM Installation Verification Test, for OPT(0) RENT
Product Name:  5694A01
Version   1    Release   9    Modification    00
Text Creation Date: 08:190
VALIDATION SUCCESSFUL

XL C/C++ for z/VM Installation Verification Test, for OPT(1)
Product Name:  5694A01
Version   1    Release   9    Modification    00
Text Creation Date: 08:190
VALIDATION SUCCESSFUL
VMFINS2760I VMFINS processing completed successfully
Ready;
```

Figure 12. Sample console output from vmfins build

6.5 Place the XL C/C++ for z/VM Compiler Into Production

6.5.1 Copy the XL C/C++ for z/VM Compiler Into Production

1 Logon to MAINT if you plan to put the XL C/C++ for z/VM Compiler on the 'Y' disk (MAINT's 19E disk). Or logon to the user ID that owns the disk that will contain the 'production' level of the XL C/C++ for z/VM Compiler.

Note: Ensure that the 19E minidisk, or the 'production' disk for the compiler has sufficient space free to hold the compiler files.

a If using minidisks

```
link 5654A22B 29e 29e rr
access 29e e
access 19e f
vmfcopy * * e = = f2 (prodid 5654A22B%CCXX olddate replace
```

The VMFCOPY command will update the VMSES PARTCAT file on the 19E disk.

b If using Shared File System

```
access 5654A22B.CCXX.TBUILD e
access 19e f
vmfcopy * * e = = f2 (prodid 5654A22B%CCXX olddate replace
```

The VMFCOPY command will update the VMSES PARTCAT file on the 19E disk.

2 If you copied XL C/C++ for z/VM Compiler to the MAINT 19E disk, re-save the CMS saved system, to return the 19E minidisk (Y-disk) to 'shared' status. See the 'Placing (Serviced) Components into Production' section of the *z/VM: Service Guide* for detailed information about how to save the CMS saved system.

The XL C/C++ for z/VM Compiler is now installed and built on your system.

6.6 Post-Installation Considerations

Upon successful installation, the compiler can be installed into a segment. If you choose to create the XL C/C++ for z/VM Compiler segment then it will get used instead of the modules on the test or production build disks.

See Chapter 8.0, “Define and Build The XL C/C++ for z/VM Compiler Saved Segment” on page 37 for a full description on how to customize and install in saved segments.

7.0 Service Instructions

This section of the Program Directory contains the procedure to install CORrective service to XL C/C++ for z/VM Compiler. VMSES/E is used to install service for XL C/C++ for z/VM Compiler.

To become more familiar with service using VMSES/E, you should read the introductory chapters in:

- *VMSES/E Introduction and Reference*

This manual also contains the command syntax for the VMSES/E commands listed in the procedure.

Note: Each step of the servicing instructions must be followed. Do not skip any step unless directed to. All instructions showing accessing of disks assume the use of default minidisk addresses. If different minidisk addresses are used, or if using a shared file system, change the instructions appropriately.

7.1 VMSES/E Service Process Overview

The following is a brief description of the main steps in servicing XL C/C++ for z/VM Compiler using VMSES/E.

- Merge Service

  Use the VMFMRDSK command to clear the alternate apply disk before receiving new service. This allows you to easily remove the new service if a serious problem is found.

- Receive Service

  The VMFREC command receives service from the delivery media and places it on the Delta disk.

- Apply Service

  The VMFAPPLY command updates the version vector table (VVT), which identifies the service level of all the serviced parts. In addition, AUX files are generated from the VVT for parts that require them.

- Reapply Local Service (if applicable)

  All local service must be entered into the software inventory to allow VMSES/E to track the changes and build them into the system. Refer to Chapter 7 in the *z/VM: Service Guide* for this procedure.

- Build New Levels

  The build task generates the serviced level of an object and places the new object on a test BUILD disk.

- Place the New Service into Production

  Once the service is satisfactorily tested it should be put into production by copying the new service to the production disk, re-saving the Saved Segments etc.

7.2 Servicing XL C/C++ for z/VM Compiler

Note: A minimum of 256M of virtual storage and 30 LDRTBLS is recommended.

7.2.1 Prepare to Receive Service

Electronic Service (envelope file)

If you have received the service electronically or on CD-ROM, follow the appropriate instructions to retrieve and decompress the envelope file to your A-disk. The decompression is currently done by using the DETERSE MODULE (shipped with VMSES/E).

The documentation envelope and the service envelope files must have a file type of SERVLINK. Make note of the file names that you are using as you will need to enter them in place of the variable *envfilename* in the VMFREC commands that follow.

The *ppfname* used throughout these servicing instructions is **5654A22B**, which assumes you are using the PPF supplied by IBM for XL C/C++ for z/VM Compiler. If you have your own PPF override file for XL C/C++ for z/VM Compiler, you should use your file's *ppfname* instead of **5654A22B**. The *ppfname* you use should be used **throughout** the rest of this procedure, unless otherwise stated.

The *compname* used throughout these servicing instructions is ONE of the following:

- **CCXX** if English product installed on minidisks.
- **CCXXSFS** if English product installed in SFS directories.
- **CCXXK** if Kanji product installed on minidisks.
- **CCXXKSFS** if Kanji product installed in SFS directories.

which assumes you are using the component name within the 5654A22B PPF file. If you specify your own *ppfname*, you should use the *compname* from that file. The same *compname* should be used **throughout** the rest of this procedure.

1 Log onto XL C/C++ for z/VM Compiler service user ID **5654A22B**

2 Establish access to the software inventory disk.

Note: If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

link maint 51D 51D mr **access 51D D**	The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

3 Have the XL C/C++ for z/VM Compiler Corrective service tape mounted and attached to ***5654A22B*** as address 181. If you have a service envelope SERVLINK file make sure it is available on the A-disk or any work disk that is accessed as file mode C.

4 Establish the correct minidisk access order.

vmfsetup 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}*

5654A22B is the name of the PPF file that was shipped with the product. If you have your own PPF override you should substitute your PPF name for 5654A22B.

```
VMFSET2760I VMFSETUP processing started for 5654A22B CCXX
VMFUTL2205I Minidisk|Directory Assignments:
            String    Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E      R/W   2C2  CXX2C2
VMFUTL2205I APPLY     F      R/W   2A6  CXX2A6
VMFUTL2205I           G      R/W   2A2  CXX2A2
VMFUTL2205I DELTA     H      R/W   2D2  CXX2D2
VMFUTL2205I BUILD0    I      R/W   29E  CXX29E
VMFUTL2205I BASE1     J      R/W   2B2  CXX2B2
VMFUTL2205I --------  A      R/W   191  CXX191
VMFUTL2205I --------  B      R/O   5E5  MNT5E5
VMFUTL2205I --------  C      R/O  11A5  MJD1A5
VMFUTL2205I --------  D      R/W   51D  MNT51D
VMFUTL2205I --------  S      R/O   190  CMS20
VMFUTL2205I --------  Y/S    R/O   19E  YDISK
VMFSET2760I VMFSETUP processing completed successfully
Ready;
```

Figure 13. Sample console output from vmfsetup

5 Receive the documentation. VMFREC, with the INFO option, loads the documentation and displays a list of all the products on the tape.

a If receiving the service from tape

vmfrec info

This command will load the service memo to the A-disk (191).

b If receiving the service from an envelope file

vmfrec info (env *envfilename*

envfilename is the file name of the documentation envelope (SERVLINK) file.

This command will load the service memo to the minidisk where the service envelope is stored.

```
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of COR ENVELOPE created on 31 July 08
VMFREC2159I Loading COR 0001 to 191 (A)
VMFREC2159I Loading COR DOCUMENT to 191 (A)
VMFREC2159I Loading 5654A22B MEMO to 191 (A)
VMFREC2160I There are 3 tape file(s) for 5654A22B CCXX on volume 1
VMFREC2760I VMFREC processing completed successfully
```

Figure 14. Sample console output from vmrec info

6 Check the receive message log ($VMFREC $MSGLOG) for warning and error messages.

vmfview receive

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

7 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

vmfmrdsk 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **apply**

This command clears the alternate APPLY disk.

```
VMFMRD2760I VMFMRDSK processing started
VMFMRD1937I Merge of APPLY started
VMFMRD1938I Merging APPLY 2A6 to 2A2
VMFMRD2065I APPLY 2A2 is now 1 percent full
VMFMRD1939I Merge of APPLY completed
VMFMRD2760I VMFMRDSK processing completed successfully
```

Figure 15. Sample console output from vmfmrdsk

8 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see the appropriate *z/VM: Messages and Codes*, or use online HELP.

vmfview mrd

7.2.2 Receive the Service

Note: If you are installing multiple service tapes, you can receive all of the service for this prodid before applying and building it.

For **each** service tape or electronic envelope you want to receive, repeat the following two steps (before you do the Apply steps):

1 Receive the service.

a If receiving the service from tape

vmfrec ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}*

This command receives service from your service media. All new service is loaded to the DELTA disk.

b If receiving the service from the PTF envelope file

vmfrec ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(env** *envfilename*

envfilename is the file name of the service (PTF) envelope (SERVLINK) file.

This command receives service from your service media. All new service is loaded to the DELTA disk.

```
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of COR ENVELOPE created on 31 July 08
VMFREC1851I (1 of 3) VMFRCAXL processing AXLIST
VMFRCX2159I Loading 4 part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 3) VMFRCPTF processing PARTLST
VMFRCP2159I Loading 1 part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 3) VMFRCCOM processing DELTA
VMFRCC2159I Loading 2 part(s) to DELTA 2D2 (H)
VMFREC2189I Updating Requisite table 5654A22B SRVREQT, Description table
            5654A22B SRVDESCT and Receive Status table 5654A22B SRVRECS
            with 1 new PTFs from COR 0001
VMFREC2760I VMFREC processing completed successfully
```

Figure 16. Sample console output from vmfrec

2 Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see the appropriate *z/VM: Messages and Codes*, or use online HELP.

vmfview receive

7.2.3 Apply the Service

1 Apply the new service.

vmfapply ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}*

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

You must review the VMFAPPLY message log if you receive a return code (RC) 4, as this may indicate that you have local modifications that need to be reworked.

```
VMFAPP2760I VMFAPPLY processing started
VMFAPP2106I Apply list 5654A22B contains 1 PTFs that need to be applied
            and 0 PTFs that are already applied
VMFAPP2102I 1 of 1 PTFs processed
VMFAPP2105I VMFAPPLY processing completed successfully.
            The Apply list 5654A22B contains 1 PTFs.
            0 PTFs were already applied.
            1 PTFs applied successfully.
            0 PTFs were included.
            0 PTFs were excluded or require excluded PTFs.
            0 PTFs failed
VMFAPP2103I The Software Inventory has been updated on the 2A6 (F) disk
```

Figure 17. Sample console output from vmfapply

2 Review the apply message log ($VMFAPP $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see the appropriate *z/VM: Messages and Codes*, or use online HELP.

vmfview apply

Note

If you get the message VMFAPP2120W then re-apply any local modifications before building the new XL C/C++ for z/VM Compiler. Refer to chapter 7 in *z/VM: Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions need to be made:

- **outmode localmod** should be **outmode localsam**
- **zvm** should be **5654A22B**
- *compname* should be one of **CCXX | CCXXSFS | CCXXK | CCXXKSFS**
- *appid* should be **5654A22B**
- *fm-local* should be the fm of 2C2
- *fm-applyalt* should be the fm of 2A6

Keep in mind that, when you get to the "Rebuilding Objects" step in the *z/VM: Service Guide*, you should return to this program directory at 7.2.4, "Update the Build Status Table" on page 32.

7.2.4 Update the Build Status Table

1 Update the Build Status Table with serviced parts.

vmfbld ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(status**

This command updates the Build Status Table.

```
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD2180I There are 2 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

Figure 18. Sample console output from vmfbld status

Note

If the $PPF files have been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
            serviced:
VMFBLD2185R 5654A22B $PPF
VMFBLD2185R When source product parameter files are serviced, all
            product parameter files built from them must be recompiled
            using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
            parameter files built to you A-disk and exit VMFBLD so
            you can recompile your product parameter files with VMFPPF
VMFBLD2185R Enter one (1) to continue only if you have already
            recompiled your product parameter files with VMFPPF
```

0

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5654A22B $PPF
on 191 (A) from level $PFnnnnn
```

vmfppf 5654A22B *

Note: If you've created your own PPF override, use your PPF name instead of 5654A22B.

copyfile 5654A22B $PPF a = = d (olddate replace
erase 5654A22B $PPF a

Note: **Do not** use your own PPF name in place of 5654A22B for the COPYFILE and ERASE commands.

vmfbld ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(status 1**

Re-issue VMFBLD to complete updating the build status table. If you have your own PPF name then you should use it on the VMFBLD command.

When you receive the prompt that was previously displayed, enter a 1 to continue.

2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

vmfview build

7.2.5 Build Serviced Objects

1 Rebuild XL C/C++ for z/VM Compiler serviced parts.

vmfbld ppf 5654A22B *{CCXX* **|** *CCXXSFS* **|** *CCXXK* **|** *CCXXKSFS}* **(serviced**

```
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I No new build requirements identified
VMFBLD1851I (1 of 2) VMFBDMOD processing CCNBLPMB EXEC J, target is BUILD0 29E
            (I)
VMFBDM2219I Processing object CCNDRVR.MODULE
VMFBLD1851I (1 of 2) VMFBDMOD completed with return code 0
VMFBLD1851I (2 of 2) VMFBDSBR processing CCNBLSGL EXEC J, target is BUILD0 29E
            (I)
VMFSBR2000I Objects in segment build list CCNBLSGL EXEC have been built or
            deleted. Any segments using this build list will have to be rebuilt.
VMFBLD1851I (2 of 2) VMFBDSBR completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

Figure 19. Sample console output from vmfbld serviced

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific build messages, see the appropriate *z/VM: Messages and Codes*, or use online HELP.

vmfview build

7.3 Place the Serviced XL C/C++ for z/VM Compiler Into Production

7.3.1 Rebuild the Saved Segment (if a segment is used).

1 Re-save the XL C/C++ for z/VM Compiler segments.

ipl cms
vmfbld ppf segbld esasegs segblist CCNSEG (serviced

```
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 1) VMFBDSEG processing SEGBLIST EXC00000 D, target is BUILD
            51D (D)
VMFBDS2115I Validating segment CCNSEG
VMFBDS2002I A DEFSEG command will be issued for 1 segment(s).
VMFBDS2219I Processing object CCNSEG.SEGMENT
DMSDCS358E Skeleton segment CCNSEG has already been reserved
HCPNSS440I Saved segment CCNSEG was successfully saved in fileid 0051.
VMFBLD1851I (1 of 1) VMFBDSEG completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

Figure 20. Sample console output from vmfbld segbld

2 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see the appropriate *z/VM: Messages and Codes*, or use on-line HELP.

vmfview build

7.3.2 Copy the New XL C/C++ for z/VM Compiler Serviced Files Into Production

1 Logon to MAINT if you plan to put the XL C/C++ for z/VM Compiler on the 'Y' disk (MAINT's 19E disk). Or logon to the user ID that owns the disk that will contain the 'production' level of the XL C/C++ for z/VM Compiler.

a If using minidisks

```
link 5654A22B 29e 29e rr
access 29e e
access 19e f
vmfcopy * * e = = f2 (prodid 5654A22B%CCXX olddate replace
```

The VMFCOPY command will update the VMSES PARTCAT file on the 19E disk.

b If using Shared File System

```
access 5654A22B.CCXX.TBUILD e
access 19e f
vmfcopy * * e = = f2 (prodid 5654A22B%CCXX olddate replace
```

The VMFCOPY command will update the VMSES PARTCAT file on the 19E disk.

2 If you copied XL C/C++ for z/VM Compiler to the MAINT 19E disk, re-save the CMS saved system, to return the 19E minidisk (Y-disk) to 'shared' status. See the 'Placing (Serviced) Components into Production' section of the *z/VM: Service Guide* for detailed information about how to save the CMS saved system.

You have now finished servicing XL C/C++ for z/VM Compiler.

8.0 Define and Build The XL C/C++ for z/VM Compiler Saved Segment

Segments are defined to the system using the segment mapping tool VMFSGMAP. Once the segments are defined VMFBLD is used to build them.

For more information on using VMSES/E for saved segments, review the chapter, 'Using VMSES/E to Define, Build, and Manage Saved Segments in the *z/VM: Saved Segments Planning and Administration* manual.

Note: The defining and building of the XL C/C++ for z/VM Compiler saved segments should be performed from the installation userid. If you move any segments that are currently defined on your system you must ensure that they are rebuilt from the userid that maintains them.

8.1 Define and Build Saved Segments Using VMSES/E

Note: You must

- Have Class E privileges to install a saved segment.
- Have a virtual storage size at least 0.5 MB greater than the address of the end of the segment. A minimum virtual storage of 256M is recommended if using default address range.
- Ensure that the shared segment does not overlap any other shared segment or saved system. For details, see the *z/VM: Saved Segments Planning and Administration* manual.

1 Logon to the installation userid **5654A22B**.

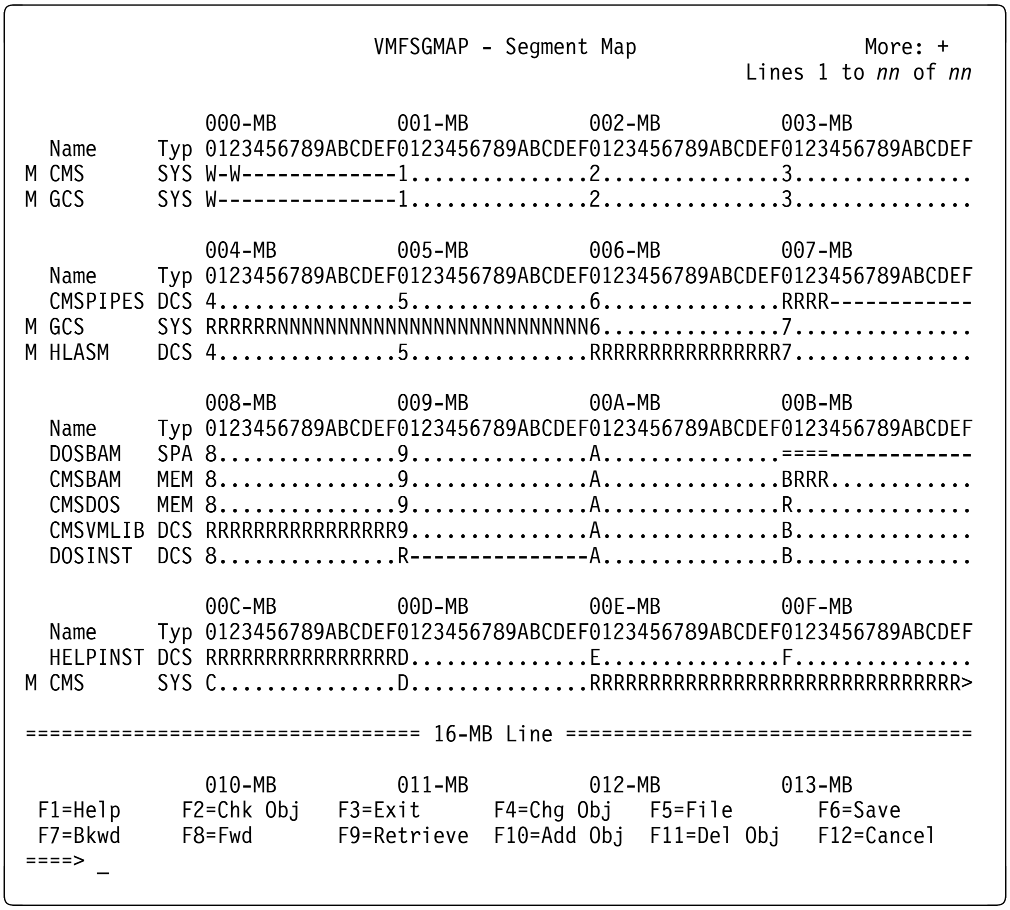
2 Establish write access to the VMSES/E and software inventory disks.

```
link maint 51D 51D mr
access 51d D
```

3 Add XL C/C++ for z/VM Compiler segment object definitions to the SEGBLIST EXC00000 build list.

```
vmfsgmap segbld esasegs segblist
```

This command displays a panel for making segment updates. See Figure 21 on page 38 for an example of the Segment Map panel that will be displayed.



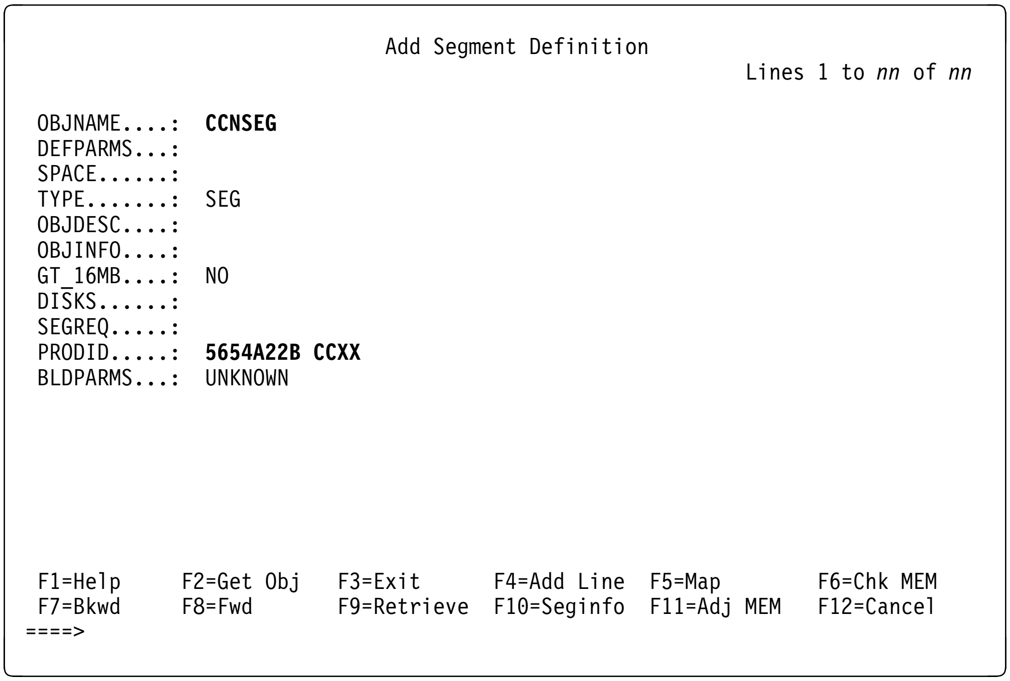
```
                              VMFSGMAP - Segment Map                  More: +
                                                             Lines 1 to nn of nn

               000-MB          001-MB          002-MB          003-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
M CMS      SYS W-W-------------1...............2...............3...............
M GCS      SYS W---------------1...............2...............3...............

               004-MB          005-MB          006-MB          007-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  CMSPIPES DCS 4...............5...............6...............RRRR------------
M GCS      SYS RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNN6...............7...............
M HLASM    DCS 4...............5...............RRRRRRRRRRRRRRRR7...............

               008-MB          009-MB          00A-MB          00B-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  DOSBAM   SPA 8...............9...............A...............====------------
  CMSBAM   MEM 8...............9...............A...............BRRR............
  CMSDOS   MEM 8...............9...............A...............R...............
  CMSVMLIB DCS RRRRRRRRRRRRRRRR9...............A...............B...............
  DOSINST  DCS 8...............R---------------A...............B...............

               00C-MB          00D-MB          00E-MB          00F-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  HELPINST DCS RRRRRRRRRRRRRRRRD...............E...............F...............
M CMS      SYS C...............D...............RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

================================= 16-MB Line ==================================

               010-MB          011-MB          012-MB          013-MB
 F1=Help     F2=Chk Obj   F3=Exit      F4=Chg Obj   F5=File       F6=Save
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Add Obj  F11=Del Obj   F12=Cancel
====> _
```

Figure 21. Segment Map panel example.

4 Go to Add Segment Definition panel by pressing PF10.

F10

F10 will take you from the Segment Map panel to the Add Segment Definition panel. See Figure 22 on page 39 to see the Add Segment Definition panel that will be displayed. You will type in the highlighted information in the the next step.



```
                          Add Segment Definition
                                                        Lines 1 to nn of nn

 OBJNAME....:  CCNSEG
 DEFPARMS...:
 SPACE......:
 TYPE.......:  SEG
 OBJDESC....:
 OBJINFO....:
 GT_16MB....:  NO
 DISKS......:
 SEGREQ.....:
 PRODID.....:  5654A22B CCXX
 BLDPARMS...:  UNKNOWN







 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map        F6=Chk MEM
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Seginfo  F11=Adj MEM   F12=Cancel
====>
```

Figure 22. Initial "Add Segment Definition" panel example.

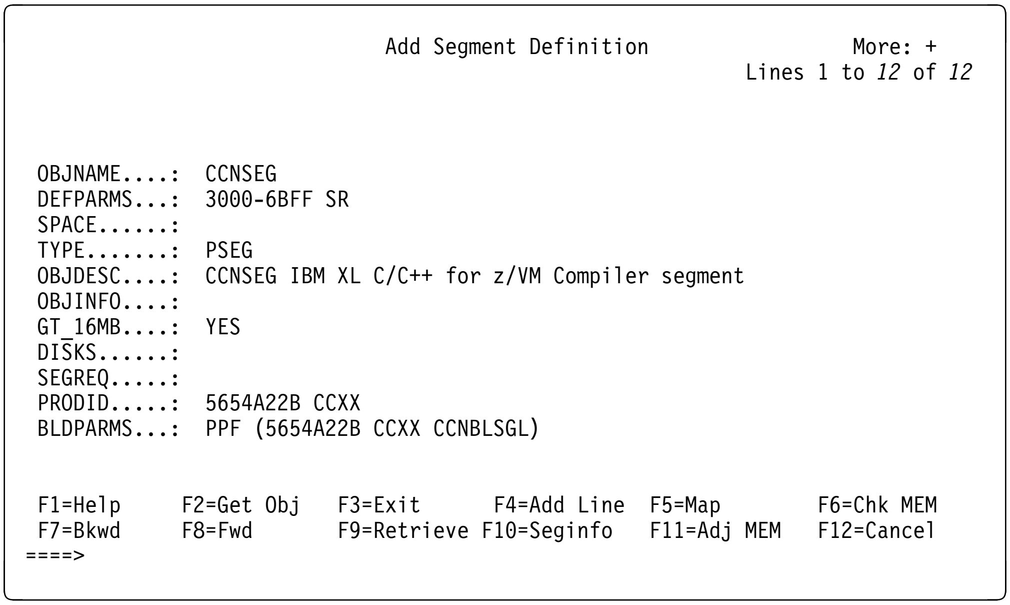
5 Obtain the CCXX segment definitions from the PRODPART file. Fill in the fields on the Add Segment Definition panel with the following information (displayed after the colon) and then use function key F10.

Note: In the panel examples the CCXX compname was used. You need to use the compname that you used during installation.

OBJNAME....: **CCNSEG**
PRODID.....: **5654A22B** *{CCXX* | *CCXXSFS* | *CCXXK* | *CCXXKSFS}*

F10

F10 will obtain the XL C/C++ for z/VM Compiler segment information from the 5654A22B PRODPART file. See Figure 23 on page 40 for the refreshed Add Segment definition panel that will be displayed.



```
                              Add Segment Definition              More: +
                                                            Lines 1 to 12 of 12



 OBJNAME....:  CCNSEG
 DEFPARMS...:  3000-6BFF SR
 SPACE......:
 TYPE.......:  PSEG
 OBJDESC....:  CCNSEG IBM XL C/C++ for z/VM Compiler segment
 OBJINFO....:
 GT_16MB....:  YES
 DISKS......:
 SEGREQ.....:
 PRODID.....:  5654A22B CCXX
 BLDPARMS...:  PPF (5654A22B CCXX CCNBLSGL)


 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map        F6=Chk MEM
 F7=Bkwd     F8=Fwd       F9=Retrieve F10=Seginfo   F11=Adj MEM   F12=Cancel
====>
```

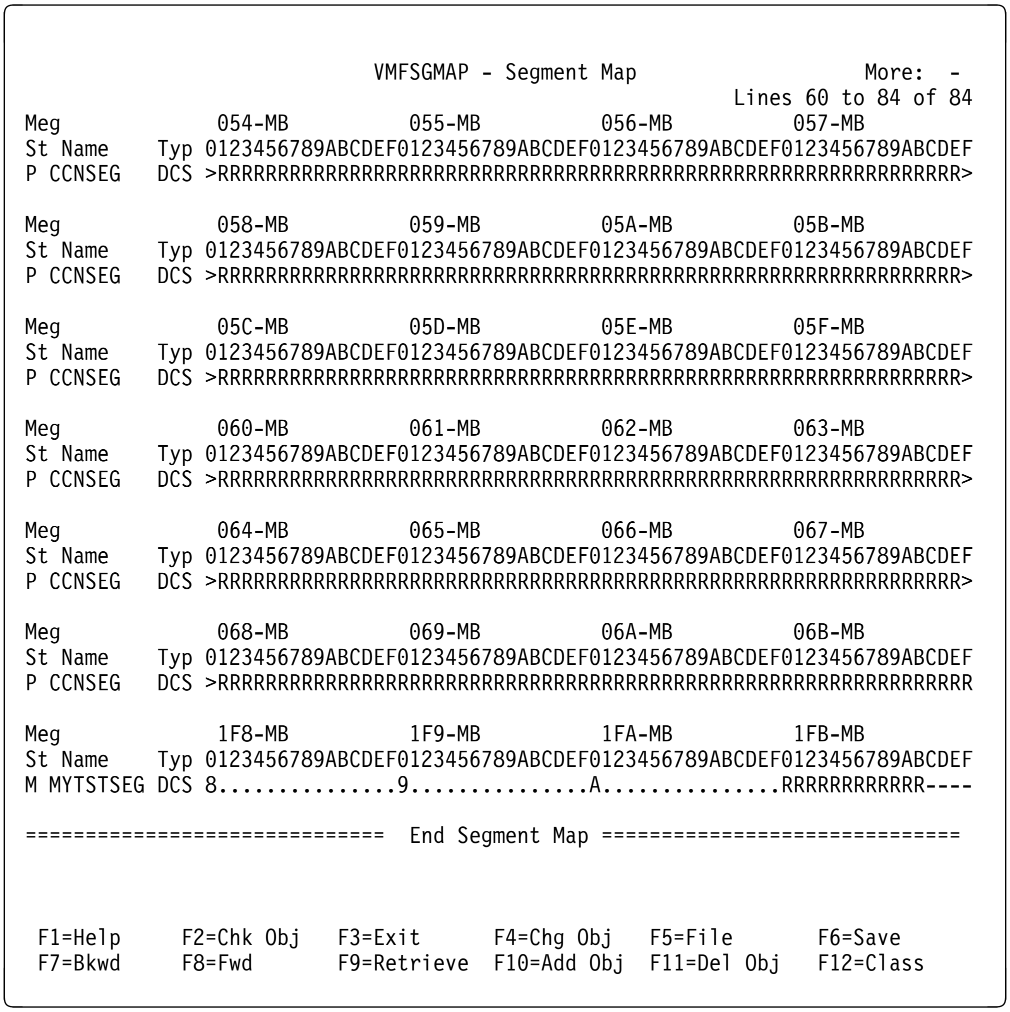
Figure 23. Segment Definition panel showing CCNSEG Segment information

6 Go back to the Segment Map panel.

F5

F5 will return you to the Segment Map panel. See Figure 24 on page 41 for the refreshed Segment Map panel that will be displayed.

If you have your own PPF override then you need to change the BLDPARMS field to reflect this.



```
                             VMFSGMAP - Segment Map                   More:  -
                                                           Lines 60 to 84 of 84
Meg             054-MB          055-MB          056-MB          057-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

Meg             058-MB          059-MB          05A-MB          05B-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

Meg             05C-MB          05D-MB          05E-MB          05F-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

Meg             060-MB          061-MB          062-MB          063-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

Meg             064-MB          065-MB          066-MB          067-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

Meg             068-MB          069-MB          06A-MB          06B-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
P CCNSEG   DCS >RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

Meg             1F8-MB          1F9-MB          1FA-MB          1FB-MB
St Name    Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
M MYTSTSEG DCS 8...............9...............A...............RRRRRRRRRRRR----

==============================  End Segment Map ==============================



 F1=Help     F2=Chk Obj   F3=Exit      F4=Chg Obj   F5=File       F6=Save
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Add Obj  F11=Del Obj   F12=Class
```

Figure 24. Segment Map panel with CCNSEG Segments

7 Save new information and exit from the Segment Map panel.

F5

```
Ready; T=nn.nn/nn.nn hh:mm:ss
```

F5 saves all changed information and exits the map panel.

8 Prepare to build the segments.

a Redefine and clear the virtual storage

```
define storage 256M
ipl 190 clear parm nosprof instseg no
```

** DO NOT press ENTER at the VMREAD!**

IPL 190 to clear your virtual machine. This command bypasses the execution of the system profile (SYSPROF EXEC) without loading the installation saved segment (CMSINST).

```
access (noprof
```

Bypass the execution of the PROFILE EXEC.

b Access the VMSES/E code

```
access 5E5 B
```

c Establish write access to the Software Inventory Disk

```
link maint 51D 51D mr
access 51d D
```

d Reset Loader Tables.

```
set ldrtbls 30
```

9 Issue VMFBLD command to build the XL C/C++ for z/VM Compiler segments.

```
vmfbld ppf segbld esasegs segblist CCNSEG (all
VMFBLD PPF SEGBLD ESASEGS SEGBLIST CCNSEG (ALL
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 1) VMFBDSEG processing SEGBLIST EXC00000 D, target is BUILD
            51D (D)
VMFBDS2115I Validating segment CCNSEG
VMFBDS2002I A DEFSEG command will be issued for 1 segment(s).
VMFBDS2219I Processing object CCNSEG.SEGMENT
HCPNSS440I Saved segment CCNSEG was successfully saved in fileid 1577.
VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and must be moved to
            the S disk.
VMFBLD1851I (1 of 1) VMFBDSEG completed with return code 4
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed with warnings
Ready; T=0.27/0.29 11:44:40
```

Figure 25. Sample Console ouput for CCNSEG Segment Load

10 Use VMFVIEW to review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before going on.

vmfview build

11 If you received the message:

VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and must be moved to the S disk

then the SYSTEM SEGID must be copied over to the S-disk in order to stay in sync with the system's SEGID. Remember to re-SAVE CMS to avoid the `Shared S-STAT not available` message. You can perform both of these actions from the MAINT user ID.

Appendix A. Overriding the Product Parameter File (PPF)

This section provides information to help you create a product parameter file (PPF) override. The example used in this section shows how to change the shared file system (SFS) file pool where the XL C/C++ for z/VM Compiler files reside, when using SFS. You can also refer to Chapter 21, 'Product Parameter File Syntax, in the *z/VM: VMSES/E Introduction and Reference*.

Note: Do **not** modify the product supplied 5654A22B $PPF or 5654A22B PPF files to change the VMSYS file pool name or any other installation parameters. If the 5654A22B $PPF file is serviced, the existing $PPF file will be replaced, and any changes to that file will be lost. By creating your own $PPF override, your updates will be preserved.

The following process describes changing the default file pool name, VMSYS to MYPOOL1:

1 Create a new $PPF override file, or edit the override file created via the 'Make Override Panel' function.

xedit *overname* **$PPF** *fm***2**

overname is the PPF override file name (such as "myCCXX") that you want to use.

fm is an appropriate file mode. If you create this file yourself, specify a file mode of A.

If you modify an existing override file, specify a file mode of A or D, based on where the file currently resides (A being the file mode of a R/W 191 minidisk, or equivalent; D, that of the MAINT 51D minidisk).

2 Modify the Variable Declarations (**:DCL.**) section for the CCXXSFS override area so that it resembles the **:DCL.** section as shown below. This override will be used for the installation of XL C/C++ for z/VM Compiler.

```
:CCXXSFS. CCXX
*==========================================================*
*  Variable definitions                                    *
*==========================================================*
:DCL. UPDATE
 &191    DIR MYPOOL1:5654A22B.                 * A DISK
 &SAMPZ  DIR MYPOOL1:5654A22B.CCXX.SAMPLE      * SAMPLE/LOCAL FILES
 &DELTZ  DIR MYPOOL1:5654A22B.CCXX.DELTA       * PRODUCT SERVICE
 &APPLX  DIR MYPOOL1:5654A22B.CCXX.APPLYALT    * AUX AND INVENTORY FILES
 &APPLZ  DIR MYPOOL1:5654A22B.CCXX.APPLYPROD   * PRODUCTION APPLY DISK
 &BAS1Z  DIR MYPOOL1:5654A22B.CCXX.OBJECT      * BASE DISK
 &BLD0Z  DIR MYPOOL1:5654A22B.CCXX.TBUILD      * BUILD DISK
:EDCL.
:END.
```

This override will update the ***:DCL.*** section of the CCXXSFS override area of the 5654A22B $PPF file.

3 If your $PPF override file was created at file mode A, copy it to file mode D—the Software Inventory minidisk (MAINT 51D).

file
copyfile *overname* **$PPF** *fm* **= = d (olddate**

4 Compile your changes to create the usable *overname* PPF file.

vmfppf *overname* **CCXXSFS**

where *overname* is the file name of your $PPF override file.

```
VMFPPF2760I VMFPPF processing started for OVERNAME CCXXSFS
VMFOVE2760I VMFOVER processing started
VMFOVE1954I Locating CCXXSFS tag in file OVERNAME $PPF on disk D1
VMFOVE1954I Locating CCXX tag in file OVERNAME $PPF on disk D1
VMFOVE1954I Applying override CCXXSFS from file OVERNAME $PPF
VMFOVE2760I VMFOVER processing completed successfully
VMFPPF2760I VMFPPF processing completed successfully for OVERNAME CCXXSFS
VMFPPF2760I VMFPPF processing completed successfully
```

Figure 26. Sample console output from vmfppf

Now that the *overname* PPF file has been created, you should specify *overname* instead of 5654A22B as the PPF name to be used for those VMSES/E commands that require a PPF name.

Appendix B. Segment Build Lists (CCNBLSGL)

```
********************************************************************
* IBM XL C/C++ for z/VM Compiler                                   *
* LICENSED MATERIALS - PROPERTY OF IBM                             *
* 5654-A22 (C) COPYRIGHT IBM CORP. - 2003, 2008                    *
* ALL RIGHTS RESERVED                                              *
* US GOVERNMENT USERS RESTRICTED RIGHTS - USE,                     *
* DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP                  *
* SCHEDULE CONTRACT WITH IBM CORP                                  *
********************************************************************
*  Buildlist for C/C++ for z/VM compiler in NSS                    *
********************************************************************
:FORMAT. 2
*
:OBJNAME. CCNSEG.SEGMENT
:BLDREQ.  CCNBLPMB.CBXFINIT.MODULE
          CCNBLPMB.CCNDRVR.MODULE
          CCNBLPMB.CCNEDFLT.MODULE
          CCNBLCPY.CCNEFILT.MODULE
          CCNBLCPY.CCNEOPTP.MODULE
          CCNBLPMB.CCNEP.MODULE
          CCNBLCPY.CCNEPP.MODULE
          CCNBLCPY.CCNER.MODULE
          CCNBLPMB.CCNETBY.MODULE
          CCNBLCPY.CCNMSGE.MODULE
:GLOBAL. TXTLIB SCEELKED
:OPTIONS. LOADFUNC ( LSEG CBXFINIT )
          LOADFUNC ( LSEG CCNDRVR  )
          LOADFUNC ( LSEG CCNEDFLT )
          LOADFUNC ( LSEG CCNEFILT )
          LOADFUNC ( LSEG CCNEOPTP )
          LOADFUNC ( LSEG CCNEP    )
          LOADFUNC ( LSEG CCNEPP   )
          LOADFUNC ( LSEG CCNER    )
          LOADFUNC ( LSEG CCNETBY  )
          LOADFUNC ( LSEG CCNMSGE  )
:EOBJNAME.
```

Figure 27. Contents of CCNBLSGL Build List

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services or features discussed in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes to the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Canada Ltd. Laboratory
Lab Director
B3/KB7/8200/MKM
8200 Warden Avenue
Markham, Ontario L6G 1C7
Canada
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities on non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operationg platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks and Service Marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at

www.ibm.com/legal/copytrade.shtml

Adobe, the Adobe logo, PostScript and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Reader's Comments

IBM XL C/C++ for z/VM version 1 release 2.0

You may use this form to comment about this document, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

For each of the topics below please indicate your satisfaction level by circling your choice from the rating scale. If a statement does not apply, please circle N.

RATING SCALE

very satisfied	**←--------------------------------→**			**very dissatisfied**	**not applicable**
1	**2**	**3**	**4**	**5**	**N**

	Satisfaction					
Ease of product installation	1	2	3	4	5	N
Time required to install the product	1	2	3	4	5	N
Contents of program directory	1	2	3	4	5	N
Readability and organization of program directory tasks	1	2	3	4	5	N
Necessity of all installation tasks	1	2	3	4	5	N
Accuracy of the definition of the installation tasks	1	2	3	4	5	N
Technical level of the installation tasks	1	2	3	4	5	N
Installation verification procedure	1	2	3	4	5	N
Ease of putting the system into production after installation	1	2	3	4	5	N

- Is this the first time your organization has installed this product?

  □ Yes
  □ No

- Were the people who did the installation experienced with the installation of VM products using VMSES/E?

  □ Yes

  How many years of experience do they have? __________

  □ No

- How long did it take to install this product? ____________________
- If you have any comments to make about your ratings above, or any other aspect of the product installation, please list them below:

  ____

Please provide the following contact information:

Name and Job Title

Organization

Address

Telephone

Thank you for your participation.

Please send the completed form to the following address, or give to your IBM representative who will forward it to the IBM XL C/C++ for z/VM Development group:

IBM XL C/C++ Development
C2/YF3/8200/MKM
8200 Warden Ave
Toronto Lab
Markham, Ontario
Canada - L6G 167

Printed in USA

GI11-2899-00