CA VM:Schedule™

Installation Guide

Version 2.0, First Edition

9/25/2013

CA Technologies Product References

This document references the following CA Technologies products:

- CA Mainframe VM Product Manager
- CA VM:Director™ for z/VM (CA VM:Director)
- CA VM:Secure™ for z/VM (CA VM:Secure)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

CA VM:Schedule 2.0 First Edition 9/25/2013

The following documentation updates have been made since the last release of this documentation:

Global change—All references to the user ID for system administrator and maintenance functions changed from VMRMAINT to VMANAGER.

Global change—Names and branding of related products were updated.

Global change—All references to installing the product from tape were removed.

Contents

Chapter 1: Overview

This document describes how to install, deploy, and service the CA VM:Schedule product.

This section contains the following topics:

Audience

Readers of this book must have knowledge in the following areas:

- CP, CMS components
- VMSES/E procedures
- VM environment and installing software in this environment
- Your organization's IT environment, processes, and procedures

You may need to work with the following personnel:

- Systems programmer, for CP and CMS definitions
- Storage administrator, for disk space allocation
- Security administrator, for establishing resource authorizations

Installation Process

The installation process is a series of steps which installs product materials and deploys a product instance from a tape image envelope file. You obtain the envelope file by downloading it or transferring it from a DVD. Following initial installation preparations, the product code is loaded to a staging area, where it can be serviced. Next, the product code is deployed either into a server virtual machine, or to a user accessible location for use. This allows a single copy of the code to be maintained in the staging area over time, and deployed to the various machines and runtime locations where the product code is accessed for execution.

The following steps describe the installation process:

1. Prepare for installation by ensuring that hardware, software, and security prerequisites are satisfied.
2. Install the product code for a specific product and version level.
    a. Retrieve informational files to determine the contents of the envelope file, and obtain the latest information about the product or products in the file.
    b. Retrieve the resource requirements for a particular product. These requirements include user ID definitions and disk space allocations.
    c. Allocate disk space to hold the product materials, either on minidisks or Shared File System directories.
    d. Load the product code from the envelope file. The code is loaded to a staging area where it can be updated with any needed fixes, without impacting running product servers.
    e. Update the build status table to complete the product code installation.
3. Deploy the product. This activity may occur many times for a single set of product code to create multiple executable instances of the product.
    a. Define the attributes for a product server virtual machine or a runtime code location. A default set of server attributes is provided in the product code, and is tailored, if needed, for a specific server instantiation.
    b. Allocate any needed server virtual machines and the associated disk space for the server.
    c. Copy the needed code and other files from the product staging area to where it will be used by a server or by an end user. The staging area may then be maintained and updated without impacting running server or user processing.
    d. Test the deployed product.

Service Process

The service process is a series of steps which installs updates to product materials and deploys a product instance from the updated materials. Following initial installation preparations, the product code is updated on the staging area created during initial installation. Next, the updated product code is deployed either into a server virtual machine or to a user accessible location for use.

Format Conventions for Code Syntax

This section describes the format and conventions used to document commands, utilities, and user exits. Each convention provides examples, describing how to use commands, how to use options, or how the system responds to user entries.

Note: The examples and instructions throughout this document use VMANAGER as the user ID for system administrator and maintenance functions. Also, this document uses [assign the value for svmuserid in your book] as the user ID for the service virtual machine (svm). These user IDs are the default values. If you use non-default user IDs for the system administrator or the svm, replace the default values throughout this document with the values that you use.

Command Abbreviations

When a command contains uppercase and lowercase letters, then the uppercase letters denote the shortest acceptable abbreviation that you can use to type the command. However, when a code item appears entirely in uppercase letters, you cannot abbreviate the item.

You can type the code item in uppercase letters, lowercase letters, or any combination.

Example:

```
CMDNAme
```

In this example, you can enter CMDNA, CMDNAM, or CMDNAME in any combination of uppercase and lowercase letters.

Continuation

The code syntax or code fragment definitions can continue from one line to the next line. The following examples describe code continuation:

Example 1:

```
A | B C | D
```

This code is equivalent to the following code:

```
A
| B C
| D
```

Example 2:

```
{choice1 | choice2 | choice99}
```

This code is equivalent to the following code:

```
 { choice1
 | choice2
 | choice99 }
```

Default Values

An underlined code item denotes the default value. The system uses the default value unless you override it. You can override the default value by coding an option from the available list.

Example:

[parm1 | *parm2* | *parm3*]

In this example, the code item *parm1* is the default value, and this is used by the system when you do not specify any of the options. However, you can code *parm1*, *parm2*, or *parm3.*

Keywords and Constants

A keyword name or constant always appears in uppercase letters. Code the keyword name or constant exactly as shown in the following example:

Example:

STOP {*tracenumber* | * | [USER] *userid*}

This example displays the USER keyword.

Optional Choices

Defines optional code items—denoted by square brackets around a code item.

Example:

```
CMDName [parm1]
```

In this example, you can choose *parm1* or no parameter at all. However, when two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

Multiple Optional Choices

When two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

Example:

In this example, you can choose *parm1, parm2, parm3,* or nothing at all.

```
[parm1 | parm2 | parm3]
```

Positional Parameters

Commands with positional parameters are identified by nested square brackets. Each positional parameter requires the specification of all previous positional parameters. The following example describes the positional parameter:

Example:

```
CMDName [posparm1 [posparm2 [posparm3]]]
```

In this example, *posparm3* can be specified only when *posparm1* and *posparm2* are also specified.

Repeatable Choices

A list of code items enclosed in square brackets and followed by an ellipsis means that you can select more than one item or, in some cases, repeat a single item.

Example:

```
[value1 | value2 | value3] ...
```

In this example, you can choose a single value, more than one value, or none of the values.

Repetition

An ellipsis following a code item means that the code item can be repeated.

Example:

```
Repeat…
```

Required Choices

You must select one item from a list of items when they are enclosed in curly braces. The items are separated by a vertical bar character.

Example:

```
CMDNAme {A | B | C}
```

In this example, your choice results in CMDNAME A, CMDNAME B, or CMDNAME C.

Special Symbols

The following list describes the meaning of the special symbols used in codes:

- {} (encloses a list of operands, one of which is required).
- [] (enclose an optional operand or operands).
- " " (enclose the name of a syntax fragment)
- underline (identifies a default value)
- | (separates alternative operands)
- . . . (Indicates that the preceding item or group can be repeated).

Symbols

The following list displays symbols. These symbols should be coded exactly as they appear in the code syntax.

- * (Asterisk)
- : (Colon)
- , (Comma)
- = (Equal Sign)
- — (Hyphen)
- ( )(Parentheses)
- . (Period)

Syntax Fragments

Some codes use fragments, when the code syntax is too lengthy. The fragment name appears between double quotes in the code syntax.

The expanded fragment appears in the syntax after all other parameters or at the bottom of the code syntax. A heading with the fragment name identifies the expanded fragment.

Example:

```
CMDName "Parms"
Parms:

[A | B | C]
```

In this example, the fragment is named "Parms", and the expanded fragment appears at the bottom of the code syntax.

System Response

Uppercase characters represent system responses or prompts.

Example:

```
ENTER YOUR LOGON PASSWORD:
```

This example displays a system response.

User–Entered Commands or Records

User-entered commands are shown in lowercase letters even though you can enter commands in either upper or lower case.

Example:

```
vmsecure addentry writers tcom (noformat nowait
```

This example shows what a user-entered command looks like.

In this example "writers tcom" is the file name and file type of the directory entry you are adding.

However, if the entry is a record, it appears in uppercase letters.

Example:

```
ACCESS DRCT 1B0 U
```

This example shows a configuration record that is entered by a user.

Variables

Lowercase items in italics denote variables.

Example:

CMDNAme *varname*

In this example, *varname* represents a variable that you must specify when you code the command CMDNAME.

Chapter 2: Preparing for Installation

This section describes what you need to know and do before you install the product.

This section contains the following topics:

Hardware Requirements

This product requires a level of the z/VM licensed program which is currently supported by IBM. IBM specifies the hardware level of each computer system supported by each level of VM, and only those hardware levels are supported for running this product.

Software Requirements

The following software is required for installing this product:

- An installed VM system running a release of VM which is supported by IBM.
- An installed deployment of the CA Mainframe VM Product Manager system. See the *CA Mainframe VM Product Manager Installation Guide* for details on installing and configuring this facility.

Resource Requirements

All user ID characteristics, directory entry definitions, and disk space resources are defined in a PLANINFO report created by VMSES/E during the installation process, as described below. You must be prepared to allocate these resources during the installation procedure.

Concurrent Releases

You can install this release of CA VM:Schedule and continue to use a deployed instance of an older release for your production environment. If you plan to continue to run a previous release, consider the following points:

Certain public files for this product, such as the product command MODULE, may only operate with a specific version of the product server. Take care to prevent product public files from one level from overlaying the same files from an earlier level of the product. You should keep such public files on different public disks or directories.

Chapter 3: Installing Product Materials

This section contains the following topics:

Installation Process Overview

The following is a brief description of the main steps in installing this product using VMSES/E.

1. Plan your Installation

   Use the VMFINS command to load several VMSES/E files from the product envelope and to obtain product resource requirements.

2. Allocate Product Resources

   The information obtained from the previous step is used to allocate the appropriate minidisks (or SFS directories) and a user ID to own them. This set of disks contains the loaded product materials and is where the code is serviced. The allocation of resources used to deploy the product is described separately in the chapter Deploying the Product (see page 27).

3. Install the Product Materials

   Use the VMFINS command to load the product files from the envelope 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.

4. Update the Software Inventory

   Use the VMFINS command to update the software inventory to prepare the product materials for later service application by updating the Software Inventory status tables.

Step 1. Plan Your Installation

Follow these steps to obtain planning information for this product.

1. Establish the CA Mainframe VM Product Manager Environment (see page 20).
2. Load Product Control Files and VMFINS PRODLIST (see page 20).
3. Load and Generate the Planning Resource File (see page 21).

Step 1.1 Establish the CA Mainframe VM Product Manager Environment

Follow these steps to establish the CA Mainframe VM Product Manager environment:

1. Log on to the VMANAGER user ID to access the CA Mainframe VM Product Manager environment.
2. Make the SERVLINK file available to the VMANAGER user ID. The SERVLINK file must be on a minidisk or Shared File System directory accessed as file mode A or C.

Note: The PROFILE EXEC on the VMANAGER user ID will access the disks or SFS directories needed for the CA Mainframe VM Product Manager environment as follows:

- The work disk is accessed as file mode A.
- The CA Mainframe VM Product Manager code is accessed as file mode B.
- The VMSES CA Software Inventory disk is accessed as file mode D.
- The CA Mainframe VM Product Manager database is accessed as file mode V.
- The CA Mainframe VM Product Manager Common disk is accessed as file mode W.
- The IBM VMSES system disk is accessed as file mode X.

Step 1.2 Load Product Control Files and VMFINS PRODLIST

The product control files include the following:

- Memo-to-Users file (MEMO)
- Product Parameter File (PPF)
- PRODPART file

The VMFINS PRODLIST contains a list of products on the installation envelope file.

The VMFINS command loads the control files to the Software Inventory Disk and creates the VMFINS PRODLIST file on the A-Disk. VMFINS loads from an envelope file. The envelope is either downloaded or retrieved from a DVD media.

Issue the following command to install the product control files and create the VMFINS PRODLIST file:

```
VMFINS INSTALL INFO ( NOMEMO ENV ZVMD020A
```

NOMEMO

Specifies that the Memo-to-Users file is to be loaded, but VMSES/E will not issue a prompt to send it to the system printer. The CMS file type of this file is MEMO.

ENV

Indicates the value following this option keyword, which is the name of the envelope file for a specific product and release. For this release the CMS file name is ZVMD020A. The CMS file type of this file is SERVLINK.

Step 1.3 Load and Generate the Planning Resource File

The planning resource file (PLANINFO) contains the planning information required to install the product. The VMFINS command with the PLAN option loads the necessary components from the envelope file and generates the PLANINFO file.

Files may be loaded either to minidisks or to Shared File System (SFS) directories using the following command:

```
VMFINS INSTALL PPF ppfname component ( PLAN NOMEMO ENV ZVMD020A
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on the Shared File System directories.

PLAN

Creates a PLANINFO file. This file contains product requisites and resources required for the product (user IDs, minidisks, and SFS directories). PLAN does not generate, allocate, or commit any system resources.

NOMEMO

Specifies that the Memo-to-Users file is to be loaded, but VMSES/E will not issue a prompt to send it to the system printer. The CMS file type of this file is MEMO.

ENV

Indicates the value following this option keyword, which is the name of the envelope file for a specific product and release. For this release the CMS file name is ZVMD020A. The CMS file type of this file is SERVLINK.

During the execution of the VMFINS command, you are prompted to create an override for the *ppfname* and *component* you specified:

```
VMFINS2601R Do you want to create an override for :PPF ZVMD020A
VMSCHEDULE :PRODID ZVMD020A%VMSCHEDULE?
Enter 0 (No), 1 (Yes) or 2 (Exit)
```

Do one of the following:

- If you are accepting the default resource names and values, specify NO to the override prompt and skip to the next section.
- If you specify YES, you are presented with a panel for changing resource values such as disk owners, disk addresses, or SFS directory names. See the appendix Create Product Parameter File (PPF) Override (see page 49) for a sample of creating a PPF override.

Note: If you are not familiar with creating PPF overrides using VMFINS, you should review the *Using the Make Override Panel* section of the IBM *VMSES/E Introduction and Reference* before you continue.

More information:

Create Product Parameter File (PPF) Override (see page 49)

Step 2. Allocate Product Resources

The VMFINS command in the previous section produces a report file containing a list of the resources needed to install and service the product. The file has the same CMS file name as the PPF file, and the file type is PLANINFO. It is created on the VMANAGER A-disk. Several alternative methods are available to allocate the needed user ID entry and disk space specified in the PLANINFO file.

In this step, disk space is allocated and owned by a VM user ID created specifically for one release of this product. The default user ID for this user is the same as the PPF CMS file name.

Disk space may be allocated either on minidisks or on Shared File System (SFS) directories. Minidisk space is often easier to setup, but is more complex to manage in that they need to be enlarged when more files are added during service. SFS directories provide more automatic space management, but need more time to setup initially.

Note: For more information about how the VMALLOC command is configured and used, see the appendix Allocating User ID Entries and Disk Space (see page 55), and then issue the command as described.

To create the product staging area user ID and allocate minidisk space, enter the following command:

```
VMALLOC PRODUCT ppfname component
```

ppfname

Specifies the CMS file name for the Product Parameter File for this specific version of the product. If you created a PPF override file, use its file name here. If you use the supplied PPF, its name is ZVMD020A.

component

Specifies the name of the component. Use the name VMSCHEDULE for installing on minidisks. Use the name VMSCHEDULESFS for installing on SFS.

Depending on the configuration described in the appendix Allocating User ID Entries and Disk Space (see page 55), you may have to perform additional steps to finish the allocation task using DIRECTXA or a Directory Manager product.

Allocating SFS Directories

An SFS Installation requires the following additional steps:

Enroll the user ZVMD020A in a SFS filepool using the CMS ENROLL command. This command may only be issued from a user authorized as an SFS Administrator for the filepool.

Create the necessary directories for user ZVMD020A with the CMS CREATE DIRECTORY command. The suggested fully qualified directory names are defined in the *:DCL* section of the VMSCHEDULESFS component override section of the ZVMD020A PPF file.

Give the VMANAGER user ID WRITE and NEWWRITE access to those directories with the CMS GRANT AUTHORITY command.

Step 3. Install the Product Materials

Use the VMFINS command to load the product files from the installation media to the BUILD and BASE minidisks or SFS directories. Enter the following command:

```
VMFINS INSTALL PPF ppfname component ( ADD NOMEMO ENV ZVMD020A
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

ADD

Loads the product materials onto disk areas. These materials will later be deployed to a product server for production use.

NOMEMO

Specifies that the Memo-to-Users file is to be loaded, but VMSES/E will not issue a prompt to send it to the system printer. The CMS file type of this file is MEMO.

ENV

Indicates the value following this option keyword, which is the name of the envelope file for a specific product and release. For this release the CMS file name is ZVMD020A. The CMS file type of this file is SERVLINK.

Step 4. Update the Software Inventory

Update the software inventory to prepare the product materials for later service application by updating the Software Inventory status tables.

Enter the following command to update the software inventory:

```
VMFINS BUILD PPF ppfname component ( SERVICED
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

SERVICED

Specifies that the Software Inventory Status tables be built.

Chapter 4: Deploying the Product

If you are migrating from an existing version of this product, which was installed using AIM and VMRMAINT, you must use the procedure in Migrating CA VM:Schedule from AIM to CA Mainframe VM Product Manager (see page 59) to deploy the product. This is expected to be a one-time action.

If you are upgrading an existing version of this product, which was installed using CA Mainframe VM Product Manager, you must use the procedure in the section Deploying the Product for an Upgrade (see page 34) to deploy the product.

If none of those situations apply, only then proceed with the procedure in the section Deploying the Product for the First Time (see page 27).

This section contains the following topics:

Deploying the Product for the First Time

The following is a brief description of the main steps in deploying an instance of this product by issuing CA Mainframe VM Product Manager commands. One installation of the product materials for a specific product version can be used to deploy multiple instances of the product. These instances might run on the same VM system, or on different VM systems at your installation.

1. Define the Product Deployment

   Use the VMDEFINE command to establish a set of parameters for the deployment. These parameters include the server machine user ID and the target disks or directories to contain the product files used by the server or end user.

2. Tailor the Product Deployment

   Follow the steps in this section to modify the default values for the allocation and deployment parameters established by the VMDEFINE command. Tailor the values to meet your requirements.

3. Allocate Resources for Production Use

   Use the VMALLOC command to create any user IDs and allocate disk space needed for the product deployment. Server products need a server virtual machine and its own copy of the product code. A running server is protected from code changes made by service updates, because it has its own copy.

   If you are installing the product into a Single System Image (SSI) cluster, define your product server as a USER machine in the z/VM directory. You will only need one CA VM:Schedule service virtual machine in an SSI cluster to service all the member's scheduled requests. Make sure all the resources assigned to the entry are available to all systems the service virtual machine could run on.

4. Deploy the Product Materials

   Use the VMDEPLOY command to populate the product disk space from the installed product materials. Initial configuration files are created and the deployment is readied for its first use.

5. Start the Product

   Follow the steps in this section to bring up the deployed product server machine for the first time.

6. Test the Product

   Follow the steps in this section to perform initial testing of the deployed server virtual machine.

7. Make the Product Available

   This step describes how to place the deployed product into production use.

Step 1. Define the Product Deployment

The VMDEFINE command creates an entry in the VMSERVER NAMES file. Attributes for the deployment are defined as tags in this standard NAMES file. A set of initial attributes for a deployment is presented and may be updated during the definition phase.

To bring up the initial NAMES file modification screen, enter the VMDEFINE command. The format is:

`VMDEFINE` *`name ppfname component`*

name

Specifies the user ID of the product server machine to be created.

ppfname

Specifies the CMS file name for the Product Parameter File for this specific version of the product. If you created a PPF override file, use its file name here. If you use the supplied PPF, its name is ZVMD020A.

component

Specifies the name of the component. Use the name VMSCHEDULE for installing on minidisks. Use the name VMSCHEDULESFS for installing on SFS.

More information:

Product Tags in VMSERVER NAMES File (see page 53)

Step 2. Tailor the Product Deployment

The VMDEFINE command creates the entry and then enters a NAMES panel for you to view or modify the definitions. You may tailor the characteristics of the CA VM:Schedule product server machine on the VMSERVER NAMES panel. Name panel usage is described in the IBM NAMES command documentation and HELP file. To change entries just overtype the values and press PF6 to change the entry. Press PF3 to exit the panel.

More information:

Product Tags in VMSERVER NAMES File (see page 53)

Required Tags

The tag names in this section are required. Although changing the device numbers is permitted, CA strongly suggests that you use the standard device numbers. Do not remove any of the following tags:

- RUNTIME
- PUBLIC
- LOCAL
- DBASE

Removable Tags

The tag names in this section define values for minidisks that are allocated by default to hold an alternate and a previous version of the product materials. Having three sets of product code makes it easy to switch between product releases or between fix levels. If you do not want to define disk space, insert a leading asterisk in the values for the following tags:

- ALTRUNTIME
- ALTPUBLIC
- PREVRUNTIME
- PREVPUBLIC

Commenting the values will request that the minidisks not be defined during deployment.

Step 3. Allocate Resources for Production Use

The *ppfname* PLANINFO file produced during product materials installation contains the directory entries for user IDs created for each product deployment as well as a list of the disk space needed.

You defined the name of the product server virtual machine when you issued the VMDEFINE command in Step 1 above. You may have altered the definition to add optional product server disks. In this step, you combine the definition in the VMSERVER NAMES file with the product user ID configuration and disk size information from the PLANINFO file to create the defined server machine and allocate disk space for the product deployment.

For more information about how the VMALLOC command is configured and used, see the appendix Allocating User ID Entries and Disk Space (see page 55) and then issue the command as described.

Enter the following command to create the user ID and allocate disk space:

```
VMALLOC SERVER name
```

name

Specifies the user ID of the product server machine to be deployed.

Depending on the configuration described in the appendix Allocating User ID Entries and Disk Space (see page 55), you may have to perform additional steps to finish the allocation task using DIRECTXA or a Directory Manager product.

Step 4. Deploy the Product Materials

The steps in this section create a usable product instance by copying the installed product build materials to the disk areas you allocated in the previous step.

Step 4.1. Copy Product Materials to Runtime Areas

Enter the VMDEPLOY command to verify the product server disks, create default configuration files, and copy product materials to the runtime disk space.

`VMDEPLOY name INIT`

name

Specifies the user ID of the product server machine being deployed.

INIT

Specifies that a new server is being deployed. A set of default configuration files and a startup PROFILE EXEC file are placed on the disk defined by the LOCAL tag in the VMSERVER NAMES file entry. The disks defined on the DEPLOY tag are populated with product materials.

Note: For more information about deploying optional Product Materials, such as HELP files, see the *CA Mainframe VM Product Manager Reference Guide.*

Step 4.2. Enter LMP Statements

Follow the procedure in the *CA Mainframe VM Product Manager Reference Guide* to define the LMP (License Management Program) key for this product.

Step 4.3. Establish Required Authorizations

If an External Security Manager product is used at your installation, establish the required authorizations for the deployed user IDs. The following table shows the authorizations needed by the default user IDs for this product.

Deployed User ID	Required Authorizations
VMSCHED	Read access to VMANAGER 0195 minidisk. Read access to VMANAGER 01FF minidisk.

Step 5. Start CA VM:Schedule

1. Log on to VMSCHED and at the prompt to start the product enter YES.
2. CA VM:Schedule initializes, then displays this message:

   ```
   VM:Schedule initialization complete on mm/dd/yy
   ```

3. Disconnect the CA VM:Schedule service virtual machine.

Step 6. Test CA VM:Schedule

Test Scheduling Requests

1. Log on to VMANAGER.
2. Schedule an EXEC named DAYCHECK EXEC to execute on your user ID every day at noon by entering this command:

   ```
   vmsched schedule req1 daycheck (at 12:00 again daily
   ```

   A message displays and indicates when your request will run for the first time.
3. Schedule a request named REQ2 to initiate every hour of every weekday to run a procedure named RPT by entering this command:

   ```
   vmsched schedule req2 rpt (at :00:01 every 1 on mon-fri again weekly
   ```

   A message displays and indicates when your request will run for the first time.
4. Use the EXEC command and the CA VM:Schedule VMDMSG EXEC to schedule a message to execute at 9 a.m. Friday, initiating a total of two times, at five-minute intervals.

   ```
   vmsched exec req3 vmdmsg maint upgrade (at 9 on fri every :5 for
   2
   ```

   A message displays and indicates when your request will run for the first time.

Query Schedule Requests

1. Enter this command to display basic information about each scheduled request:

   ```
   vmsched query *
   ```

2. To display information about REQ1, including details on each specific option used to schedule the request, enter a QUERY command with the LONG option:

   ```
   vmsched query req1 (long
   ```

Change Request Status

1. To skip the first two initiations of REQ2, enter the following:

   ```
   vmsched skip req2 2
   ```

   A message displays and states that REQ2 will be skipped two times.

2. To verify that REQ2 has been rescheduled appropriately, enter the following:

   ```
   vmsched query req2
   ```

3. To postpone only the next initiation of REQ1 for 15 minutes, enter the following:

   ```
   vmsched delay req1 for :15
   ```

   A message displays and states that the next initiation of REQ1 will be delayed for 15 minutes.

4. To query the request, enter the following:

   ```
   vmsched q req1
   ```

   Querying the request tells you that the next initiation of REQ1 is scheduled for 12:15 instead of 12:00.

5. To cancel all the test requests, enter the following:

   ```
   vmsched cancel req*
   ```

   A series of messages display and state that REQ1, REQ2, and REQ3 have been cancelled.

6. To verify that all the scheduled requests have been cancelled, enter the following:

   ```
   vmsched query *
   ```

   These cancelled requests remain in the CA VM:Schedule request database, VMSCHED IRBDB, for three days, that is, if the PURGE record in the VMSCHED CONFIG file has not been changed.

Step 7. Make the Product Available

After you finish installing and testing the product, you must make it available to your users before they can use it. For more information about making the product available to your users, see the chapter "Releasing Products to Users" in the *CA Mainframe VM Product Manager Reference Guide.*

Deploying the Product for an Upgrade

The following is a brief description of the main steps in deploying an instance of this product by issuing CA Mainframe VM Product Manager commands. Use this process to deploy to a target user ID that is running a previous release of the same product. The procedure allows an easy transition to a new product release while preserving the ability to go back to the older release, if necessary.

Important! We recommend that you back up the product configuration files, program materials, and data files before upgrading the product.

1. Redefine the Product Deployment

   Use the VMSERVER command to update a set of parameters for the deployment. These parameters include the Product Parameter File (PPF) name, and possibly the target disks to contain the product files used by the server or end user.

2. Allocate Resources for Production Use

   Use the VMALLOC command to update any user IDs and possibly modify the disk space needed for the product deployment. Some product minidisks may need to be enlarged for a new release.

3. Deploy the Product Materials

   Use the VMDEPLOY command to populate the product disk space from the installed product materials.

4. Back Out the Upgrade

   Follow the steps in this section to restore the previous release of the product, if necessary.

5. Discard the Product Materials from the Older Release

   Follow the steps in this section to discard the previous release of the product and release the disk space it occupied.

Step 1. Redefine the Product Deployment

The VMSERVER command displays an entry in the VMSERVER NAMES file. Attributes for the deployment are defined as tags in this standard NAMES file. A set of previously defined attributes for a deployment is presented and may be updated during the redefinition phase. In this step you will update the entry for a deployment to reflect attributes of the new product release.

To display the NAMES file modification screen, enter the VMSERVER command. The format is:

VMSERVER *name*

name

Specifies the user ID of the product server machine to be updated. You assigned this name when the product was originally installed, as described in Define the Product Deployment.

A new product release is installed with a new Product Parameter File. You will update the PPF Name field on the screen by overtyping it.

For the first token, specify the CMS file name for the Product Parameter File for this specific version of the product. If you created a PPF override file, use its file name here. If you use the supplied PPF, its name is ZVMD020A.

For the second token, specify the component name VMSCHEDULE for a minidisk installation or VMSCHEDULESFS for a Shared File System installation.

After updates are made, press PF6 to change the entry. Press PF3 to exit the panel.

More information:

Product Tags in VMSERVER NAMES File (see page 53)

Step 2. Allocate Resources for Production Use

Retrieve the needed directory statements and minidisk definitions for the new release of CA VM:Schedule from the product materials you previously loaded from the distribution media. The VMALLOC command must be set up first as described in the Allocating User ID Entries and Disk Space (see page 55) section. In the simplest form, create or update a VMALLOC CONFIG file to add a device or volume record. To produce a directory entry, enter the following command:

```
VMALLOC SERVER name ( DIRECT
```

name

Specifies the user ID of the product server machine you are migrating.

Match the retrieved directory entry with your existing directory entry for the CA VM:Schedule server. Upgrade your current definition to match the requirements for the new release of CA VM:Schedule.

Step 3. Deploy the Product Materials

The steps in this section create a usable product instance. Copy the installed product build materials to the disk areas you allocated in the previous step.

Step 3.1. Copy Product Materials to Runtime Areas

Enter the VMDEPLOY command to verify the product server disks, and copy product materials to the runtime disk space.

```
VMDEPLOY name [PRIMARY|ALTERNATE]
```

name

Specifies the user ID of the product server machine being deployed. For more information about this ID, see Define the Product Deployment.

PRIMARY

Copy to the product primary disk space. This value results in overwriting the deployed Product Materials from the older product release or version with the new release materials. We suggest that you backup all deployed minidisks before using this operand. Restoring the backup could be used to revert to the older release, if needed.

ALTERNATE

Copy to the alternate disk space location. This value retains your older product material deployment. Setting the Runtime Environment tag value in the VMSERVER NAMES file to PRIMARY or ALTERNATE causes that set of deployed disks to be used when the product server is next started. Testing the new release, or backing out to the old release just requires setting a new value for Runtime Environment and restarting the server.

Note: For more information about the individual tag definitions for NAMES file entry, see the appendix Product Tags in VMSERVER NAMES File (see page 53).

More information:

Product Tags in VMSERVER NAMES File (see page 53)

Step 3.2. Enter LMP Statements

Follow the procedure in the *CA Mainframe VM Product Manager Reference Guide* to define the LMP (License Management Program) key for this product, if necessary.

Step 4. Back Out the Upgrade

The steps in this section restore the previous release of CA VM:Schedule if that step becomes necessary.

To display the NAMES file modification screen, enter the VMSERVER command. The format is:

VMSERVER *name*

name

Specifies the user ID of the CA VM:Schedule server. For more information about this ID, see Define the Product Deployment.

The first step in Deploying the Product for an Upgrade documented how to update the entry in the VMSERVER NAMES file to define a new Product Parameter File. You updated the PPF Name field on the screen by overtyping it. Reverse that step to restore the field to its original contents.

If you deployed this new product release to the ALTERNATE disk space, you can direct the product to revert to using the PRIMARY disk space the next time it is started. Deploying to ALTERNATE space leaves the older release intact on the PRIMARY disks. To bring back the old release of the product, set the Runtime Environment tag value to PRIMARY and restart the server.

If you deployed the new product release to the PRIMARY disk space, you can repopulate that space with the older release product materials. To copy product materials to the runtime disk space, enter the VMDEPLOY command using the following format.

```
VMDEPLOY name PRIMARY
```

name

Specifies the user ID of the product server machine being deployed.

PRIMARY

Copies product files to the product PRIMARY disk space. This value results in overwriting the upgraded Product Materials from the newer product release or version with the older release materials.

After you resolve the situation that required you to back out the upgrade, repeat the upgrade procedure from the beginning to re-install the upgrade.

Step 5. Discard the Product Materials from the Older Release

When you have no further use for the the previous release of CA VM:Schedule, follow the steps in this section to discard the previous release and free the disk space that it occupied.

Each release of a product is installed and maintained on minidisks or SFS directories owned by a unique user ID. For this release, that ID is ZVMD020A. Using your Directory Manager product or procedures, simply remove the product specific user ID which was used by the older release. Use the VMSERVER command to ensure that no entry in the VMSERVER NAMES file still references the old product materials ID.

Step 6. Make the Product Available

After you finish upgrading and testing the product, you must make it available to your users before they can use it. For more information about making the product available to your users, see the chapter "Releasing Products to Users" in the *CA Mainframe VM Product Manager Reference Guide*.

Chapter 5: Servicing Your Product

This section provides information on how to perform service updates using VMSES/E.

This section contains the following topics:

VMSES/E Service Overview

Service is a process that includes the following:

- Correcting a software problem
- Circumventing a software problem
- Adding function to a previously installed product

There are two basic types of service.

- Corrective service is intended to resolve a specific software defect. A corrective service envelope contains a specific Program Temporary Fix (PTF) for one specific problem, plus any requisite fixes.
- Preventive service is a collection of fixes as of a certain calendar date. It might contain all published PTFs, or some subset of them (for example, high impact fixes). The procedure for processing either type of service is identical, so the procedures below work for either type. In a collection of fixes, VMSES will ignore any fixes in the collection which have already been applied to your product materials.

Note: For more information about VMSES Service Process, see the chapter "Using VMSES/E for Service" in the IBM *VMSES/E Introduction and Reference* manual.

Servicing VMSES/E components consists of the following steps; each step is described in detail in later sections.

1. Prepare to Receive Service

   Load informational files from the PTF envelope file. Establish the search order for working with the product code. Optionally, you may merge the alternate APPLY disk to the production APPLY disk to isolate the new service you are about to receive.

2. Receive the Service

   Load the product service from the PTF envelope file, using the VMSES VMFREC command.

3. Apply the Service

   Define a new service level by applying the PTFs. The VMSES VMFAPPLY command validates the loaded PTFs before adding them to the maintenance level.

4. Update the Build Status Table

   Use the VMSES VMFBLD command with the STATUS option to update the Build Status Table.

5. Build Serviced Objects

   Use the VMSES VMFBLD command with the SERVICED option to build the serviced objects.

6. Deploy Service to Production

   Use the VMDEPLOY command to copy the serviced code to the production disks.

Step 1. Prepare to Receive Service

Follow these steps to prepare to receive service.

1. Prepare the VMANAGER Administration User ID (see page 41).
2. Receive the Documentation (see page 41).
3. [Optional] Merge the Alternate APPLY disk (see page 42).

Note: The *ppfname* used throughout these servicing instructions is ZVMD020A, which assumes that you are using the PPF supplied for CA VM:Schedule. If you have created your own PPF override file for the product, you must use your override file name instead of ZVMD020A. The *ppfname* must be used throughout the rest of this procedure, unless otherwise stated.

Step 1.1. Prepare the VMANAGER Administration User ID

Log on to VMANAGER.

The PROFILE EXEC establishes the correct search order for applying service.

It is a good practice to create a backup copy of the current CA VM:Schedule disks or SFS directories, which contain product materials. Save this backup until you have completed installing the service and you are confident that the service runs correctly.

Make the service envelope (SERVLINK) file available on the A-disk or any minidisk or SFS directory accessed as file mode C.

Setup the installed product materials access search order by issuing the VMFSETUP command as follows:

```
VMFSETUP ppfname component ( LINK
```

ppfname

Specifies the value of the PPF filename. In most cases, use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, use the file name of your PPF override file.

component

Specifies the name of the component. Use the name VMSCHEDULE for installing on minidisks. Use the name VMSCHEDULESFS for installing on SFS.

Important! The steps in the service application procedure assume that the product search order has been established before you begin the procedure. The search order is lost if disks are released manually, if CMS is restarted by the IPL command, or if you logoff and logon again. If the search order is lost at any time, re-establish it by issuing the VMFSETUP command before proceeding to the next VMFREC, VMFMRDSK, VMFAPPLY, or VMFBLD command.

Step 1.2. Receive the Documentation

Load from an envelope file by issuing the following command:

```
VMFREC INFO ( ENV envfilename
```

INFO

Specifies that the documentation, including the product service memo, is loaded to the A-disk.

ENV

Indicates the value following this option keyword, which is the name of the envelope file for the service being applied. The CMS file type of this file is SERVLINK.

Check the receive message log ($VMFREC $MSGLOG) for warning and error messages. Use the PF5 key to show all status messages which identify the products with service.

```
VMFVIEW RECEIVE
```

Read the product memo file ZVMD020A MEMO for the latest information about this service.

Step 1.3. [Optional] Merge the Alternate APPLY disk

This step is optional. The APPLY string contains the files that define service levels. Before receiving a new service, you may choose to consolidate service levels by merging previously processed service from the alternate APPLY disk to the production APPLY, where it is permanently stored. The merge operation provides a clean alternate APPLY disk for new service. Issue the VMFMRDSK command as follows:

```
VMFMRDSK ppfname component APPLY
```

ppfname

Specifies the value of the PPF filename. In most cases, use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, use the file name of your PPF override file.

component

Specifies the name of the component. Use the name VMSCHEDULE for installing on minidisks. Use the name VMSCHEDULESFS for installing on SFS.

If you merged your APPLY disks, review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before proceeding.

```
VMFVIEW MRD
```

Note: For more information about handling specific error messages, see the appropriate VM Messages and Codes publication, or use online HELP.

Step 2. Receive the Service

Receive the service for each service electronic envelope you want to process. If you have multiple service envelopes to process, you may receive all of the service before applying and building it. Enter the following command:

```
VMFREC PPF ppfname component ( ENV envfilename
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

ENV

Indicates the value following this option keyword, which is the name of the envelope file for the service being applied. The CMS file type of this file is SERVLINK.

Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before proceeding.

```
VMFVIEW RECEIVE
```

Note: For more information about handling specific error messages, see the appropriate VM Messages and Codes publication, or use online HELP.

Step 3. Apply the Service

Apply all received service. 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) of a 4, as this may indicate that you have local modifications that need to be reworked.

To apply the service, enter the following command:

```
VMFAPPLY PPF ppfname component
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before proceeding.

```
VMFVIEW APPLY
```

Note: For information about handling specific error messages, see the appropriate VM Messages and Codes publication, or use online HELP.

Step 4. Update the Build Status Table

To update the build status table, enter the following command:

```
VMFBLD PPF ppfname component ( STATUS
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

Use VMFVIEW to review the build status messages.

```
VMFVIEW BUILD
```

Step 5. Build Serviced Objects

To build the serviced objects, enter the following command:

```
VMFBLD PPF ppfname component ( SERVICED
```

PPF

Indicates the values following this keyword, and the values are the Product Parameter File (PPF) file name, followed by the name of a component defined in the PPF.

In most cases, to specify the *ppfname* use the name of the supplied PPF file for this release, which is ZVMD020A. If you have created a PPF Override file, as described below and in the appendix Create Product Parameter File (PPF) Override (see page 49), you specify the file name of your PPF override file.

For the *component* value, use VMSCHEDULE for installing on minidisks. Specify VMSCHEDULESFS for installing on Shared File System directories.

Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before proceeding.

```
VMFVIEW BUILD
```

Note: For information about handling specific error messages, see the appropriate VM Messages and Codes publication, or use online HELP.

Step 6. Deploy Service to Production

To deploy serviced product materials to production, enter the following VMDEPLOY command to copy serviced product materials to the runtime disk space.

```
VMDEPLOY name [PRIMARY|ALTERNATE]
```

name

Specifies the user ID of the product server machine being deployed.

PRIMARY

Copy to the product primary disk space. This value results in overwriting the deployed Product Materials from the product release or version with the new serviced materials. We suggest that you backup all deployed minidisks before using this operand. Restoring the backup could be used to remove a level of service, if needed.

ALTERNATE

Copy to the alternate disk space location. This value retains your older product material deployment. Setting the Runtime Environment tag value in the VMSERVER NAMES file to PRIMARY or ALTERNATE causes that set of deployed disks to be used when the product server is next started. Testing the new service, or backing out to the old level just requires setting a new value for Runtime Environment and restarting the server.

Appendix A: References

This section contains the following topics:

VMSES/E Facilities

This product is installed and serviced using the VMSES/E component of VM. VMSES/E is described in the IBM *VMSES/E Introduction and Reference* manual for your release of VM. VMSES/E provides the following:

- Commands to install Product Materials from an envelope file
- Commands to receive service, apply service, and build serviced components
- A software inventory database that stores product status, Program Temporary Fixes (PTFs) which have been applied, and requisite relationships between products and fixes.

VMSES/E Commands

This section provides a brief description of VMSES commands and their purpose:

- VMFINS EXEC

  Retrieve product resource planning documents.

  Install the product materials from a distribution media.

  Delete previously installed products that are no longer used.

- VMFVIEW EXEC

  Review stored messages from previously issued commands.

- VMFREC EXEC

  Receive Program Temporary Fixes from a distribution media.

- VMFAPPLY EXEC

  Apply PTFs by resolving requisite relationships between fixes.

- VMFBLD EXEC

  Constructs usable forms from serviced parts.

- VMFSETUP EXEC

  Establishes a minidisk or Shared File System search order for one product version.

- VMFSIM EXEC

  Obtain the name(s) and/or status of local modifications you have installed.

- VMFREM EXEC

  Remove an installed local modification or PTF.

Appendix B: Create Product Parameter File (PPF) Override

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 product installation files reside.

You must never modify the product supplied ZVMD020A $PPF or ZVMD020A PPF files to change the file pool name or any other installation parameters. If the ZVMD020A $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 NEWPOOL.

This section contains the following topics:

Step 1. Create a new $PPF override file

Create an override file either manually, using an editor, or use the *Make Override Panel* function when prompted by the VMFINS INSTALL command during product installation.

Create or modify the variable declarations area (:DCL) so it looks like the following sample.

```
************************************
* VM:Schedule - SFS install       *
************************************
:OVERLST. VMSCHEDULESFS
:VMSCHEDULESFS. VMSCHEDULESFS ZVMD020A
:DCL. UPDATE
&LMODZ DIR NEWPOOL:ZVMD020A.LOCALMOD
&APPLX DIR NEWPOOL:ZVMD020A.APPLYALT
&APPLZ DIR NEWPOOL:ZVMD020A.APPLYPROD
&DELTZ DIR NEWPOOL:ZVMD020A.DELTA
&BLD0Z DIR NEWPOOL:ZVMD020A.TSYSTEM
&BLD4Z DIR NEWPOOL:ZVMD020A.TPUBLIC
&SAMPZ DIR NEWPOOL:ZVMD020A.SAMPLE
&BLD6Z DIR NEWPOOL:ZVMD020A.HELP
&BAS1Z DIR NEWPOOL:ZVMD020A.BASE
:EDCL.
:END.
```

If your $PPF override file was not created on the Software Inventory Minidisk at file mode letter D, then move the file to that disk. Compile your changes to create the usable PPF file from the $PPF file.

VMFPPF *overname* VMSCHEDULESFS

overname

Defines the name of your override file.

Step 2. Allocate the SFS Directories

Use the ENROLL command to allocate an SFS file space for the owner of the directories and then issue a CREATE DIRECTORY command to create the directories named in your PPF Override file.

Allow the VMANAGER user ID to write into the directories by issuing the GRANT AUTHORITY command for each directory, with the attributes WRITE and NEWWRITE.

Note: For more information about the use and format of the commands ENROLL, CREATE DIRECTORY, and GRANT AUTHORITY, see IBM SFS documentation.

Step 3. Continue with the Product Installation

Restart the product installation in the chapter Installing Product Materials (see page 19) at step 1.3 by issuing the VMFINS command with the PLAN option. In that step, and in subsequent steps, use the name of your PPF Override file wherever the *ppfname* is present in the install commands.

Appendix C: Product Tags in VMSERVER NAMES File

This section lists the NAMES file tags that CA VM:Schedule uses.

During deployment, you create a NAMES entry with default file tag values by using the CA Mainframe VM Product Manager VMDEFINE command. You can alter these defaults during the deployment, when the screen created by VMDEFINE appears. After the deployment, you can modify the entries in the NAMEs file using the VMSERVER command from the VMANAGER user ID.

Note: For more information about the VMSERVER command, see the *CA Mainframe VM Product Manager Reference Guide.* For more information about processing NAMES, see the Usage Notes in the help panel brought up by the HELP NAMES command.

TAG Name	Definition of the Tag value
Nick	Name of a CA VM:Schedule product Server Virtual Machine (SVM)
PPFname	The Product Parameter File (PPF) and base component name associated with this product server. This tag is modified when upgrading to a newer product release.
Environment	'primary' or 'alternate' or 'previous' This tag is modified to select between two sets of deployed product disks. The selected set becomes active the next time the product is started. The deploy tag contains the tag names of disks defined as primary disks. The same tag names are prefixed by the string "alt" to form the names of the tags defining the alternate set of product disks. The same tag names are prefixed by the string "prev" to form the names of the tags defining the previous set of product disks.
Startcmd	Command to start the product server MODULE
Prodname	Product identifier is a single token unique to each product. It must not be modified.
Deploy	List of tag names. Each tag defines a product minidisk, which will be populated by the VMDEPLOY command. You must not modify this tag value.
Runtime	Primary minidisk containing the code executed in the product server virtual machine.

TAG Name	Definition of the Tag value
Public	Primary minidisk containing the code executed by the general user from the user's own user ID.
Help	Minidisk or SFS directory containing the product HELP files
Altruntime	Alternate version of the runtime disk. It usually contains files from a new release of the product or files with fixes applied.
Altpublic	Alternate version of the public disk. It usually contains files from a new release of the product or files with fixes applied.
Prevruntime	Previous version of the runtime disk. It usually contains files from a previous release of the product or versions of files before fixes were applied.
Prevpublic	Previous version of the public disk. It usually contains files from a previous release of the product or versions of files before fixes were applied.
Local	A work disk accessed as file mode A
Dbase	A work disk that contains the request database

Appendix D: Allocating User ID Entries and Disk Space

Installing product materials and deploying product servers requires that you create or update CP User Directory Entries, and that you allocate minidisk space. Performing these tasks depends on policies and procedures that are specific to your installation. The CA Mainframe VM Product Manager provides the VMALLOC command to assist you in performing these tasks.

This section contains the following topics:

Configuring the VMALLOC command processor

You issue the VMALLOC command to either generate the data you need to allocate these resources, or to actually perform the resource allocation task automatically.

The actions performed by the VMALLOC command depend on parameters you specify in an optional configuration file. To specify these parameters, create a VMALLOC CONFIG file on the CA Mainframe VM Product Manager configuration file disk, which is accessed as file mode V.

The following configuration file statements are supported:

`DEVTYPE [3390 | 3380 | 9336 | FB-512]`

This statement specifies MDISK Directory statements will be produced that define space in units of cylinders for Count Key Data (CKD) devices or 512-byte blocks for Fixed Block Architecture (FBA) devices.

`VOLUME` *`volser`*

This statement specifies MDISK Directory statements will be produced that define space to be allocated on a specific disk volume.

`USE [VMSECURE | VMDIRECT]`

This statement specifies that an already installed and operational CA VM:Secure or CA VM:Director product is available, and that it will be used to automatically allocate user ID entries and minidisk space. When this statement is specified, make sure that the following conditions are met:

1. The CA Mainframe VM Product Manager must have a copy of the appropriate VMSECURE or VMDIRECT MODULE file on an accessed minidisk or SFS directory. If your CA VM:Secure or CA VM:Director has another name, specify that name on the USE statement.
2. The CA VM:Secure or CA VM:Director server PRODUCT CONFIG file must contain a “PRODUCT VMSISERV VMANAGER” record to authorize the VMANAGER ID for automatic allocation.
3. An entry for VMANAGER must exist in the VMSECURE MANAGERS or VMDIRECT MANAGERS file

`POOL` *`poolname`*

This statement specifies that automatic allocation will use a specific disk sub-pool defined in the DASD CONFIG file of your CA VM:Secure or CA VM:Director server. The statement is ignored if the USE statement is not present.

Using the VMALLOC command processor

The VMALLOC command operates in one of two modes, depending on whether the “USE” configuration file statement is specified. Without it, VMALLOC generates an input file suitable for processing by the DIRECTXA command, or by a VM Directory Manager product. When “USE” is defined, VMALLOC calls CA VM:Secure or CA VM:Director to allocate the specified user ID and disk space.

The syntax of the command is given earlier, in the chapters where you use it.

Using VMALLOC with the DIRECTXA Command

You may add user ID entries and minidisk definitions to your USER DIRECT file and bring the changes online with the DIRECTXA command. See the IBM *CP Planning and Administration* guide for your release of VM for details on creating user directory entries. VMALLOC generates a directory entry file on the A-disk for use as input to DIRECTXA.

MDISK statements are generated as comments which specify the required disk sizes in units of blocks of a specified block size. If you configure the DEVTYPE statement, sizes are given in units of cylinders or FBA 512-byte blocks. If you configure the VOLUME statement, the specified volume will appear in the generated MDISK statements.

Using VMALLOC with a Directory Manager Product

VMALLOC generates a directory entry file on the A-disk for use as input to a directory manager product, such as DIRMAINT from IBM, CA VM:Secure, or CA VM:Director. See the relevant product manuals for details on adding user IDs and minidisk space with these products.

MDISK statements are generated as comments which specify the required disk sizes in units of blocks of a specified block size. If you configure the DEVICE statement, sizes are given in units of cylinders or FBA 512-byte blocks. If you configure the VOLUME statement, the specified volume will appear in the generated MDISK statements.

Using VMALLOC Automatic Resource Allocation with CA VM:Secure or CA VM:Director

If either of these CA directory manager products is installed and the "USE" parameter is configured, then VMALLOC calls the CA Directory Manager product directly to allocate the needed user ID entries and disk space.

Appendix E: Migrating CA VM:Schedule from AIM to CA Mainframe VM Product Manager

The previous versions of CA VM:Schedule reside in an AIM-maintained environment using VMRMAINT. The current versions reside in a VMSES-maintained environment using VMANAGER and the CA Mainframe VM Product Manager. This section describes the recommended procedure for the one-time initial migration of the product from the AIM environment to the VMSES environment.

This section contains the following topics:

Prepare for Product Migration

To prepare your environment for this migration, perform the product installation steps that the Installing Product Materials (see page 19) section describes.

Important! We recommend that you back up your existing environment before proceeding.

Before you perform this procedure, ensure that you understand the concepts and the procedures that the *CA Mainframe VM Product Manager Reference Guide documents*. This procedure converts the AIM installed CA VM:Schedule release to use the CA Mainframe VM Product Manager for server startup processing and server administration tasks. This setup is similar to the installation that the CA Mainframe VM Product Manager performs.

Update the Product Virtual Machine Definition

The CA VM:Schedule Server Virtual Machine (SVM) requires minor changes in the virtual machine configuration to operate with the CA Mainframe VM Product Manager.

Note: The changes in this section may be performed without an outage of your VMSCHED server.

The server is first defined in the VMSERVER NAMES file, which serves as the main software manager database. Issue the following command to define the server:

`VMDEFINE name ppfname component`

name

Specifies the user ID of the product server machine you are upgrading. In most cases, the name is VMSCHED.

ppfname

Specifies the CMS file name for the Product Parameter File for this specific version of the product. If you created a PPF override file, use its file name here. If you use the supplied PPF, its name is ZVMD020A.

component

Specifies the name of the component. Use the name VMSCHEDULE for installing on minidisks. Use the name VMSCHEDULESFS for installing on SFS.

The VMDEFINE command creates the entry and then enters a NAMES panel for you to view or modify the definitions. You can tailor the characteristics of the CA VM:Schedule product server machine on the VMSERVER NAMES panel.

For definitions of all available tag values for the CA VM:Schedule product, see the appendix Product Tags in VMSERVER NAMES File (see page 53). Make any necessary modifications to the NAMES file entry to match your current CA VM:Schedule configuration.

Note: For more information about NAMES panel usage, see the IBM NAMES command documentation and HELP file.

To change the definitions, overtype the values and press PF6. Press PF3 to exit the panel.

Retrieve the needed directory statements and minidisk definitions for the new release of CA VM:Schedule from the product materials you previously loaded from the distribution media. The VMALLOC command must be setup first as described in the appendix Allocating User ID Entries and Disk Space (see page 55). In the simplest form, create or update a VMALLOC CONFIG file to add a device or volume record. Issue the following command to produce a directory entry:

```
VMALLOC SERVER name ( DIRECT
```

name

Specifies the user ID of the product server machine you are migrating.

Update the Product Initialization Files

The final set of changes requires an outage of your CA VM:Schedule server. The server startup files are modified to interface with the CA Mainframe VM Product Manager startup interface. Shutdown the CA VM:Schedule server machine before proceeding.

Compare the CP Directory entry file created by VMALLOC with your existing directory entry for the CA VM:Schedule server, and update your current definition to match the requirements for the new release of CA VM:Schedule. The following additional changes may be required, depending on what you have defined already:

- Add an alternate runtime and an alternate public disk at addresses 0292 and 0293 respectively. These disks are populated when you deploy your new release.
- Increase the size of existing disks following the size requirements from the DIRECT file created by VMALLOC.
- Remove the LINK to the VMRMAINT 192 as 1FF.
- Add a LINK to the VMANAGER 1FF disk as 1FF.
- Add a LINK to the VMANAGER 195 disk as 1FE.

From the VMANAGER user ID, issue the VMDEPLOY command to upgrade the server startup procedure and deploy the new release product code. Assuming the server user ID is VMSCHED, execute the command as follows:

```
VMDEPLOY name INIT ALTERNATE
```

name

Specifies the user ID of the product server machine you are migrating.

VMDEPLOY checks all minidisks before any changes are made. Each disk is checked for correct size and block size. Minidisks that have errors are flagged with a message. Fix any problems reported with the minidisks and then rerun the preceding VMDEPLOY command. If all minidisks are correct, then VMDEPLOY proceeds to load code.

Because the new CA VM:Schedule code is deployed to the ALTERNATE series of minidisks, your current release remains intact. VMDEPLOY will also create a backup copy of the original PROFILE EXEC on the VMSCHED 191 as PROFILE OLDEXEC, then it will create a new PROFILE EXEC that contains the following command:

```
EXEC PROFRUN
```

PROFRUN integrates the server initialization process with the CA Mainframe VM Product Manager procedures.

Use XEDIT to add authorizations for your CA Mainframe VM Product Manager user ID, VMANAGER, to the CA VM:Schedule configuration file.

Note: For specific information about updating the configuration file, see the *CA VM:Schedule System Administrator's Guide*.

- Add authorizations for the VMANAGER user ID, so that it has the same authorizations as the old VMRMAINT user ID.
- Add this record to the VMSCHED CONFIG file:

  ```
  PRODUCT VMSISERV VMANAGER
  ```

Logon to the CA VM:Schedule server user ID and perform the following steps:

1. In response to the initial prompt asking whether you want to start the server, enter NO.
2. Using XEDIT, add any desired PROFILE statements from your PROFILE OLDEXEC file to the PROFILE EXEC, but leave the PROFRUN call as the last line.

3. Check for any product materials from your old release of CA VM:Schedule on the A-Disk of the server. Issue the command:

   ```
   VMNDFCAD
   ```

4. Any duplicate file on the A-Disk should replace the corresponding file on the D-Disk. These duplicate files were probably created when fixes were applied to the release of CA VM:Schedule installed by AIM. Copy these duplicate files to the D-DISK using the COPYFILE command with the OLDDATE and REPLACE options, then erase the copy of the duplicated file from the A-Disk.

Your CA VM:Schedule server is now setup to run under control of the CA Mainframe VM Product Manager. If the ENVIRONMENT tag definition in the NAME file entry is set to PRIMARY, then the next time you start it, the server will execute the older release of CA VM:Schedule that you originally installed using AIM.

When you are ready to test the new CA VM:Schedule release, set the ENVIRONMENT tag definition to ALTERNATE and restart the server virtual machine. When you decide to complete the upgrade to the new CA VM:Schedule release, move the new release materials to the PRIMARY disk space, and overwrite the older release product materials. Use the following format to enter the VMDEPLOY command to copy product materials to the primary runtime disk space.

```
VMDEPLOY name PRIMARY
```

name

Specifies the user ID of the product server machine being deployed.

PRIMARY

Copies product files to the product PRIMARY disk space. This value results in overwriting the older release Product Materials with the newer release materials.