

Rocket Model 204 Installation Guide for IBM z/VM

Version 7 Release 4.0

May 2012
204-74-zVM-IG-01



Notices

Edition

Publication date: May 2012

Book number: 204-74-zVM-IG-01

Product version: Rocket Model 204 Installation Guide for IBM z/VM Version 7 Release 4.0

Copyright

© Computer Corporation of America 1989-2012. All Rights Reserved.

Computer Corporation of America is a wholly-owned subsidiary of Rocket Software, Inc.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

License agreement

This software and the associated documentation are proprietary and confidential to Rocket Software, Inc., are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulation should be followed when exporting this product.

Contact information

Web Site: www.rocketsoftware.com

Rocket Software, Inc. Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA
Tel: +1.617.614.4321
Fax: +1.617.630.7100

Contacting Technical Support

If you have current support and maintenance agreements with Rocket Software and CCA, contact Rocket Software Technical support by email or by telephone:

Email: m204support@rocketsoftware.com

Telephone :

North America +1.800.755.4222

United Kingdom/Europe +44 (0) 20 8867 6153

Alternatively, you can access the Rocket Customer Portal and report a problem, download an update, or read answers to FAQs. You will be prompted to log in with the credentials supplied as part of your product maintenance agreement.

To log in to the Rocket Customer Portal, go to:

www.rocketsoftware.com/support

and click **Rocket M204**.

Contents

About this Guide

1 Introduction

In this chapter	1
Model 204.....	1
Documentation on Model 204 V7R1.0 features	2
Supported releases	2
Preinstallation environment.....	2
Operating system requirements.....	2
Hardware requirements.....	2
Model 204 installation	2
Software delivery.....	2
Model 204 software.....	3
Archive identifier	3
Autofix files	3
Additional Rocket Model 204 features	3
Using Freeway/204	3
Installation considerations for Freeway/204	4
CPUIDZAP file.....	4

2 Preparing to Install Model 204

In this chapter	7
Overview	7
Using mathematical functions	7
Gathering reference materials	8
Model 204 and related documentation	8
Decryption keys	8
Additional software package documentation.....	8
Defining the MAINT204 directory entry	9
MAINT204 machine	9
MAINT204 files.....	9
Sample MAINT204 directory definition.....	9
Defining the MODEL204 directory entry.....	11
MODEL204 machine.....	11
System file definitions	12
Minidisk addresses.....	12
Sample MODEL204 directory definition	12
Creating Model 204 system files	14
Defining the GCS directory entry.....	16
Sample GCS service machine directory definition	16
Defining the user directories	17
Sample user directory definition	18
Setting up the Cross-Reference facility service machine	18
XREF service machine: directory definition	18
Sample PROFILE EXEC for the service machine.....	19
Modifying CCAIN	20
Initializing disks	20
Changing access from R/R to R/W.....	20

CMS-format disks.....	20
Variable-format disks	21
3 Installing the Model 204 Nucleus	
In this chapter	23
Installation checklist.....	23
Contents of the Model 204 VMARC archive.....	24
Downloading the VMARC file archiving module	25
Downloading the installation archive.....	25
Unpack installation program	26
Unpack installation files.....	26
Unpack individual archives (optional)	27
Modifying the CPUIDZAP ZAP file	27
Decrypting optional features	27
Customizing MAINT204 and MODEL204 profiles	28
MAINT204 profile.....	28
Customizing the MAINT204 profile	28
MODEL204 profile	28
Customizing the MODEL204 profile.....	29
Specifying TXTLIBs	29
SORT utility	30
TXTLIBs and sort exits OI15CMS and OI35CMS	30
Preparing user exits	30
Assembling ACCTEXIT, SCTYEXIT, and ALLOEXIT user exits	30
AMODE settings.....	31
Required file names and file types.....	31
If you do not use these exits.....	31
ACCTEXIT: accounting exit.....	31
SCTYEXIT: security exit.....	32
ALLOEXIT: allocation exit.....	34
Guidelines for allocation exits.....	34
Preparing site-specific CUST, CDTB, FUNU, and MSGU modules	35
Modifying \$functions.....	35
Implementing the date/time stamp feature	36
Installing the date/time stamp feature.....	36
Customizing the date/time stamp field name in CUST source code.....	37
Assembling CUST, CDTB, FUNU, and MSGU	38
Preparing FLOD exits	38
Modifying the ONLINE loadlist	38
Setting FLOD exits OI15CMS and OI35CMS.....	39
Customizing parameter values (M204USER PARMS).....	39
M204CCA PARMS file	40
M204USER PARMS file	41
Examples of modifying M204USER PARMS	44
Generating modules and saved segments.....	45
Using the M204GEN command defaults.....	45
Saving segments	46
Copying modules to the 195 disk.....	47
Preparing the GCS server component	47
VTAM Interface.....	47
Horizon Interface.....	48
Parallel Query Option/204.....	48
4 Completing the Installation	
In this chapter	49

Overview	49
If you are upgrading from an earlier release of Model 204	50
Model 204 operating modes	50
Defaults assumed	50
ONLINE command	50
Backup facilities	51
Allocating Model 204 files	51
M204UTIL ALLOCATE and the ZBLDTAB EXEC	51
Access minidisks with appropriate file modes	51
Allocating operational work files	52
Allocating the demonstration database	52
Checking allocations	53
Modifying EXECs and CCAIN streams	54
Modifying sample EXECs	54
M204MOUN EXEC	55
M204SPL EXEC	55
M204UNLD EXEC	55
Creating password and group data sets	55
Support for expiring passwords	56
z/OS-format disk	56
Creating the password data set	56
Creating the CCAGRP file	56
Applying early warnings	57
Applying early warnings with Autofix	57
Bringing up the Model 204 Online	58
Running recovery in subsequent runs	59
Bringing down the Model 204 Online	59
Accessing the Model 204 Online	60
Multiuser Online	60
Interactive single-user mode	62
BATCH2 mode	64
IFAM1 and IFAM2 modes	65
Single-user IFDIAL mode	66
Using standalone Model 204 utilities	66
Types of standalone utilities	67
Error tracking	67
Using the CCAPRINT file	67
Using console messages	68
Using audit trail information	68
Using automatic dumps	68
Using the VMDUMP command	68

5 Preparing to Install Dictionary/204

In this chapter	71
Overview	71
Dictionary/204 facilities	72
Overview of EXEC and CCAIN installation files	72
Using EXECs	72
D204 EXEC	73
Printing a copy of a Model 204 EXEC or CCAIN	73
Required storage	73
Customizing EXECs	73
FILEDEF modifications	73
Additional modifications to EXECs and CCAIN files	74

CMS-format FILEDEF statement syntax	74
z/OS-format FILEDEF statement syntax	75
Required privileges	75
Modifying LOGIN CCAIN and LOGINSM CCAIN	76
Linking 193 disk as C	76
Modifying LOGIN CCAIN	77
Modifying LOGINSM CCAIN	77
Files allocated and used by EXECs.....	77
File usage.....	78
Customizing common FILEDEF statements.....	79

6 Installing or Upgrading Dictionary/204

In this chapter	81
Overview	82
Task list	82
Common procedures	83
Allocating, creating, and restoring M204INST	83
Modifying D204FDEF EXEC to allocate space for M204INST	83
Running D204 DDPROCI EXEC.....	84
Preparing M204PROC.....	84
Procedure file.....	84
Calculating M204PROC size	84
Allocating and creating M204PROC	84
Increasing M204PROC size for additional products	85
Back up M204PROC	85
Restoring distribution procedure files	86
Using the RESPROC EXEC	86
Transferring procedures from the distribution files using TRANSFR.....	87
Overview for new Dictionary/204 installations	88
Calculating file sizes.....	89
CCASYS file.....	89
METADATA file	90
DATALINK file.....	91
D204RPT file	91
M204TEMP file	92
M204DCTL file	92
D204SYS file.....	93
Installing the standard Dictionary/204 facilities.....	94
D204 INSTALL command	94
Backing up files.....	96
Bringing up the Model 204 Online	96
ONLINE EXEC	96
Modifying the user-created EXEC.....	96
Setting up the Online.....	96
Recovery	97
Bringing up the system	98
Authorizing Dictionary/204 access for all users	98
Applying early warnings.....	98
Upgrading Dictionary/204	98
Backing up Dictionary/204 files.....	98
Using D204 BACKUP	98
Using D204 RESTORE	100
Upgrading the standard Dictionary/204 facilities	100
Backing Up Dictionary/204 files	101

Preparing existing files with DDGEN	101
Bringing up the Online	101
Preparing existing files for file management	101
7 Dictionary/204 Operation and File Maintenance	
In this chapter	103
Overview	103
When to modify EXECs	104
Backing up Dictionary/204 files	104
Files to be backed up	104
Using D204 BACKUP	104
Modifying D204FDEF EXEC and CCAIN files	105
Backing up M204PROC	105
Restoring Dictionary/204 files	106
Using D204 RESTORE	106
Modifying D204FDEF EXEC and CCAIN files (if necessary)	106
Restoring M204PROC	107
Reorganizing Dictionary/204 files	107
Files to be reorganized	107
Using D204 REORGANIZE	108
Reorganizing D204RP	108
Including site-defined Dictionary/204 extensions	108
EXECs and CCAIN for files that might require reorganization	109
Modifying the D204FDEF EXEC and CCAIN file	109
Populating Dictionary/204 from a Model 204 file	110
Using D204 DDGEN	110
Modifying D204FDEF EXEC	110
Modifying DDGEN EXEC	111
Modifying the DDGEN CCAIN file	111
Running DDGEN	111
Setting up D204 DDGEN based on Dictionary/204 files	111
Using D204 DDGENSET	111
Modifying the D204FDEF EXEC	112
Modifying the DDGENSET EXEC	112
Modifying the DDGENSET CCAIN	112
Running DDGENSET	113
Editing the OUTDDG1 file	113
Updating files using delayed processing	113
Listing files that have pending commands	113
Modifying D204FDEF EXEC and DDFIMCMD CCAIN files	113
LOGADD parameter	114
Reinitializing M204TEMP	114
When to use D204 DDTINIT	114
Modifying DDTINIT CCAIN	115
8 Installing Connect ★ Support	
In this chapter	117
Overview	117
Before you begin	118
Installing SQL	118
Product and feature dependencies	118
SQL installation steps	118
Installing SQL catalog facilities and optional subsystems	119
If you have already installed the SQL catalog or subsystems	119
Using CCACAT	119

TSFPROC: File organization changed	119
SQL catalog and subsystem installation steps.....	119
Installing SQL catalog subsystems	121
Installing CCATSF.....	121
Installing CCACATREPT.....	121
Enabling Connect★ access	121
Reinstalling SQL catalog files and optional subsystems	122
Rebuilding the SQL catalog	122
Updating your catalog without rebuilding.....	123
Setting SQL runtime parameters	124
Loading the demonstration database DDL.....	124
Renaming your schema	124
Verifying SQL installation.....	124
A Object Modules	
In this appendix	127
Object modules.....	127
B Utilities and EXECs	
In this appendix	135
Model 204 Utility commands	135
AUDIT204 utility program	136
MERGEJ utility program	136
ZBLDTAB utility program.....	137
ZCTLTAB utility program	137
Usage Notes	138
Examples.....	139
z/VM utility commands.....	139
M204APND utility program	139
M204CMS command.....	140
M204CRYP command	140
M204FDEF command	143
M204LDEF command.....	144
M204XFER command.....	144
M204UTIL functions.....	145
M204UTIL utility program	145
Shared DASD	146
ALLOCATE function	146
ERASE function	147
INITIAL function	148
LABEL function	149
LIST function	149
RENAME function	150
Internal EXECs	150
M204MOUN EXEC.....	150
M204SPL EXEC.....	150
M204UNLD EXEC	150
M204GEN command	151
Input to the M204GEN command	151
M204GEN syntax.....	151
M204GEN configurations	152
Return codes	153
M204SAVE command.....	154
Storage required for using M204SAVE for saved segments	154
Using M204SAVE for MSS generation	155

Return codes	156
Maintaining the MAINT204 machine	156
MAINTACC command	156
MAINT204 links to database minidisks	156
C Decision Making	
In this appendix	159
Overview	159
Choosing CMS-format or variable-format files	159
CMS-format files	160
Variable-format files	161
Determining whether to generate modules or segments	162
Summary of configurations	162
Saved segments	163
When to use saved segments	163
TPROCESS	163
Saved segments	164
When to use SSPs	164
Saved segments: architectural differences	164
Determining whether to use default segment definitions	165
Saved segment definitions	166
Default saved segment definitions	166
D Changing Saved Segment Definitions	
In this appendix	169
Overview	169
Guidelines for changing saved segment definitions	170
If you are upgrading from an earlier Model 204 release	170
Guidelines for changing the M204USER PARMs file	170
Where to load saved segments	170
TEST and PROD saved segments	171
Reentrant and non-reentrant code	171
MSSs and SSPs	171
Loadlists	172
Location	172
Saved segment loadlist names	172
Loadlist format	173
%SLC statement syntax	173
VMFLOAD command not supported for loadlists	174
FLOD exits	174
z/VM examples	174
Changing the origin of M204USR	174
Changing the origin of M204CMS/M204IFM1 and M204CMS/M204ONLN	174
Making M204USR an MSS and packing it with other segments	175

Index

About this Guide

This guide contains the information needed to install the Model 204 Database Management System (DBMS), including Dictionary/204, in a z/VM environment.

The installation and maintenance processes described in this guide provide a summary of the Model 204 system characteristics, operations, and installation procedures that are unique to the z/VM version of Model 204. This guide is a supplement to the documents that comprise the Model 204 documentation set.

This document is not customized to your installation's requirements and may supply instructions for features that you do not use.

Audience

This guide is directed to the person or people responsible for installing and maintaining Model 204 at your site.

Model 204 documentation set

The complete commercially released documentation for the latest version of Model 204 is available for download from the Rocket M204 customer portal.

To access the Rocket Model 204 documentation:

1. Navigate to:
<http://www.rocketsoftware.com/m204>
2. From the drop-down menu, select **Products > Model 204 > Documentation**.
3. Click the link to the current release and select the document you want from the list.
4. Click the .zip file containing the document.
5. Choose whether to open or save the document:
 - Select **Open** and double-click the pdf file to open the document.
 - Select **Save as** and select a location to save the zip file to.

Documentation conventions

This guide uses the following standard notation conventions in statement syntax and examples:

Convention	Description
TABLE	Uppercase represents a keyword that you must enter exactly as shown.
TABLE <i>tablename</i>	In text, italics are used for variables and for emphasis. In examples, italics denote a variable value that you must supply. In this example, you must supply a value for <i>tablename</i> .
READ [SCREEN]	Square brackets ([]) enclose an optional argument or portion of an argument. In this case, specify READ or READ SCREEN.
UNIQUE PRIMARY KEY	A vertical bar () separates alternative options. In this example, specify either UNIQUE or PRIMARY KEY.
TRUST <u>NOTRUST</u>	Underlining indicates the default. In this example, NOTRUST is the default.
IS {NOT LIKE}	Braces ({ }) indicate that one of the enclosed alternatives is required. In this example, you must specify either IS NOT or IS LIKE.
item ...	An ellipsis (. . .) indicates that you can repeat the preceding item.
item ,...	An ellipsis preceded by a comma indicates that a comma is required to separate repeated items.
All other symbols	In syntax, all other symbols (such as parentheses) are literal syntactic elements and must appear as shown.
<i>nested-key</i> ::= <i>column_name</i>	A double colon followed by an equal sign indicates an equivalence. In this case, <i>nested-key</i> is equivalent to <i>column_name</i> .
Enter your account: sales11	In examples that include both system-supplied and user-entered text, or system prompts and user commands, boldface indicates what you enter. In this example, the system prompts for an account and the user enters sales11 .
File > Save As	A right angle bracket (>) identifies the sequence of actions that you perform to select a command from a pull-down menu. In this example, select the Save As command from the File menu.
EDIT	Partial bolding indicates a usable abbreviation, such as E for EDIT in this example.

1

Introduction

In this chapter

- Model 204
- Supported releases
- Model 204 installation
- Additional Rocket Model 204 features

Model 204

The Model 204 database management system (DBMS) provides facilities for the creation, control, query, and maintenance of database files.

Data intensive batch and Online application systems can be developed with Model 204's self-contained User Language and embedded TP monitor. Application languages, such as Assembler, COBOL, PL/I, and FORTRAN can communicate with Model 204 functions through the Model 204 Host Language Interface. Model 204 supports SQL queries from PCs.

Model 204 operates in a service virtual machine through a separate program, the CMS Interface to Model 204. The Model 204 CMS Interface provides services and mechanisms required by Model 204 that are not normally available in CMS.

If you are installing Model 204 in an environment with multiple versions of CMS, you need to generate a separate copy of the CMS Interface for each CMS version.

Documentation on Model 204 V7R1.0 features

If you are migrating to V7R4.0 from a Model 204 release prior to V7R1.0, you need to understand the functionality and features that were introduced in Model 204 V7R1.0.

Refer to the *Model 204 Version 7, Release 1.0 Release Notes* and the Model 204 V7R1.0 guides, available for direct download from the Rocket Software M204 web site. (See “Model 204 documentation set” on page iii.)

Supported releases

To find the operating system environments that have been certified with Model 204 V7R4.0:

1. Go to the Rocket M204 web site:
<http://www.rocketsoftware.com/m204>
2. From the **Products** drop-down menu, select
Model 204 > Operating System Support

This list is updated when Rocket certifies different environments with Model 204 releases. If you have questions about an environment that is not listed, contact Technical Support.

You perform most installation steps the same way for all supported z/VM environments. This guide explicitly notes, where appropriate, any z/VM release-specific variations in the installation process.

Preinstallation environment

High Level assembler (HLASM) is required to correctly assemble source.

Operating system requirements

Model 204 V7R4.0 requires the following operating system support: z/VM Version 5 Release 4.0 or later.

Hardware requirements

Model 204 V7R4.0 requires IBM zSeries processors.

Model 204 installation

Software delivery

Installation from tape is no longer supported.

Rocket provides you with site access logon and password to download the M204 installation components as well as product installation and authorization keys.

Model 204 software

Model 204 is available in a VMARC archive on the FTP server.

See “Downloading the installation archive” on page 25 for details.

The VMARC archive file includes:

- Model 204 nucleus files
- Dictionary/204 files
- Installation files
- Demonstration databases

Archive identifier

The archive identifier is CM $rrrn$, where rrr is a three-digit release number and n is a digit from 0 to 9: for example, CM7400.

Autofix files

If early warnings are available for this release, also download Autofix files from the Rocket Software Technical Support web site. The Autofix files include maintenance early warnings for Model 204 and User Language products. For more information, see “Applying early warnings” on page 57 and refer to the *Rocket Model 204 Autofix Installation and Operations Guide*.

Additional Rocket Model 204 features

Besides the Model 204 core product, a number of additional features are available. Some of these, such as Dictionary/204, are released with Model 204 at no extra cost to your site. Others are available as a separately purchased feature.

Using Freeway/204

Model 204 provides customers with a free version of some Model 204 software features. This software is fully functional, although the number of simultaneous users is limited. When you purchase a feature, a zap is provided which opens the door to expanded usage.

Furthermore, SNA Communications Server (formerly VTAM) is no longer an encrypted feature offering, but is automatically included as a fully functional component of Model 204.

These changes are designed to offer you the opportunity to try PQO/204, Horizon, and particularly, the SQL and Remote Command Line facilities of Connect ★. Connect ★, using Horizon communication capabilities, is the cornerstone for another Rocket Software product, Analytics/204.

Installation considerations for Freeway/204

If you are concerned about the increased size of the Model 204 nucleus caused by this packaging decision or wish to defer this new benefit, you need to override some of the new default settings found in M204GEN. See “M204GEN command” on page 151 for more information.

Customers who are already enjoying the advantages of these products will see no growth in code size, but should review the installation process for changes that might affect their final configuration.

Note: If these features are included for the first time in your installation, Early Warning maintenance for these features should be monitored. Pay particular attention to the maintenance directed at the RSQL module.

If your site has purchased any of these features, use the decryption keys provided by Rocket so that you can decrypt a part of the feature before using it. The decryption process is discussed in Chapter 3.

For each feature that you are installing, review the appropriate section of this manual before continuing. The Model 204 add-on features described in this manual are:

- Connect ★
- Horizon
- Parallel Query Option (PQO)

CPUIDZAP file

After installing the Model 204 software, you must modify the CPUIDZAP file with values provided by Rocket. This process is discussed in Chapter 3.

When a CPUID zap is needed

Your site needs a zap when:

- you install a new CPU
- you conduct disaster recovery on a different CPU
- an additional virtual machine has a virtual serial number that differs from the real machine serial number.

How to obtain a new or additional CPUID zap

You are provided with a CPUID zap when you request a new release of Model 204. If you need a zap at any other time, contact Technical Support.

2

Preparing to Install Model 204

In this chapter

- Overview
- Gathering reference materials
- Defining the MAINT204 directory entry
- Defining the MODEL204 directory entry
- Creating Model 204 system files
- Defining the GCS directory entry
- Defining the user directories
- Setting up the Cross-Reference facility service machine
- Initializing disks

Overview

You must set up the z/VM environment into which you plan to install Model 204. This chapter explains how to prepare the z/VM environment.

Using mathematical functions

The Model 204 mathematical functions can access certain IBM mathematical routines. Model 204 supports the FORTRAN run-time library (VS FORTRAN

Version 2 Release 6) or the newer LE mathematical functions that are included in the z/VM operating system, if selected.

For mathematical functions...	Apply for z/VM...
VS FORTRAN Version 2.6	UN99430
LE	Not required

Check with IBM to see if any other PTFs are required.

The FORTRAN library is required to use the User Language Math Pack \$functions.

Gathering reference materials

As you begin Model 204 installation, consult the following documentation as needed. For information on accessing the documentation, see “Model 204 documentation set” on page iii.

Model 204 and related documentation

- *Rocket Model 204 Autofix Installation and Operations Guide*
- *Connect ★ Suite Installation and Programming*
- *Rocket Model 204 Dictionary/204 and Data Administration Guide*
- *Rocket Model 204 File Manager’s Guide*
- *Rocket Model 204 Late Breaking News*
- *Rocket Model 204 Parameter and Command Reference*
- *Rocket Model 204 Release Notes*
- *Rocket Model 204 System Manager’s Guide*

Decryption keys

Use the decryption keys specific to your site provided by Rocket.

Additional software package documentation

If you are installing or completing installations of other software packages, collect these documents as well; for example: *Rocket M204 Horizon: Intersystem Processing Guide*.

Defining the MAINT204 directory entry

MAINT204 machine

This section discusses assigning the minidisks to the Model 204 maintenance virtual machine, MAINT204. The MAINT204 machine is a separate virtual machine used to perform maintenance tasks related to one or more Model 204 service virtual machines.

MAINT204 files

You must load all files necessary to install and run Model 204 on the MAINT204 machine. Although you can run Model 204 from the MAINT204 machine, you cannot edit EXECs while Model 204 is running. By using the Model 204 service machine, with links to the MAINT204 disks, to run Model 204, you allow MAINT204 to be the location where EXECs can be modified. All the Model 204 files loaded onto the MAINT204 machine are in CMS format.

Sample MAINT204 directory definition

A sample entry in the User Directory for a Model 204 maintenance virtual machine with a user identifier of MAINT204 is shown below. Use this example to create a MAINT204 directory entry for your site. Make sure that items that are in italics (for example, *password*) conform to your site's configuration parameters and standards. Enter items not in italics exactly as they are shown.

```

USER      MAINT204      password  11M  16M  EG
ACCOUNT   account        distcode
OPTION    MIH
MACHINE   node
IPL       CMS PARM AUTOOCR
CONSOLE   009 3210 T userid
SPOOL     00C 2450 READER A
SPOOL     00D 2450 PUNCH A
SPOOL     00E 1403      A
LINK      CMSowner 190 190 RR
LINK      CMSowner 19E 19E RR
MDISK     191 type start size volser MR
MDISK     193 type start size volser MR
MDISK     194 type start size volser MR
MDISK     195 type start size volser MR
MDISK     200 type start size volser MR

```

USER statement

The virtual machine storage size varies according to the locations of z/VM saved segments.

The maximum virtual machine size must be large enough to contain the Model 204 saved segment storage areas. The user class E is required if saved segments are to be saved from this machine.

LINK statements

CMSowner is the name of the virtual machine where system-wide files are kept (usually MAINT). Be sure to link the system disks that contain the system sortlib (for example, SYNCSORT) and the FORTRAN TXTLIB to the maintenance machine.

MDISK 191 statement

The 191 minidisk must contain the PROFILE EXEC, MAINTACC EXEC, and work areas for Model 204 maintenance.

Recommended value: 750 4K blocks (minimum size)

MDISK 193 statement

The 193 minidisk holds the various files used to install and operate Model 204 and associated end-user products. The 193 disk contains the input files for the M204GEN command. Model 204 modules are generated with the M204GEN command and installed here. The MODEL204 service virtual machine requires access to this minidisk in read-only mode.

Recommended value: 3750 4K blocks (minimum size)

MDISK 194 statement

The 194 minidisk must hold the Model 204 TEXT files, encrypted code, loadlists, macro libraries, source programs, and EXEC procedures that generate and maintain Model 204 modules.

Recommended value: 4,500 4K blocks (minimum size)

MDISK 195 statement

The 195 disk is a suggested minidisk address for the files that access Model 204 from CMS. This disk contains files that a CMS user needs to connect to a Model 204 service machine.

As an alternative to the 195 disk, you can use a system-owned disk or any other disk that a service machine can access to load this set of files. For example, these files can reside on the 193 disk or on a common system tools disk.

MAINT204 might require access to the MODEL204 service machine's database minidisks to perform data set maintenance functions. For convenience, you can define the address used by MAINT204 to access the linked service machine's database minidisks to be the same as the address defined in the service virtual machine's directory. For example:

LINK MAINT204 . . . 195 195.

If multiple Model 204 service machines are established, also define their 195 (or equivalent address) minidisks to the maintenance machine. You can assign sequential virtual addresses (196, 197, . . .) to these disks in the maintenance machine's configuration.

Recommended value: 2300 4K blocks (minimum size)

MDISK 200 statement

You can load database files distributed in the Model 204 DUMP format to the 200 disk or any other minidisk of sufficient size. Calculations should include only the sizes of the files for the database applications to be installed at your site. Table 2-1 lists the names and approximate amounts of space required to load the various Model 204 application database files.

Table 2-1. Application names and space allocations

Application files	Size (4K blocks)
Dictionary/204	1250
Model 204 demo files	1300
M204INST	700
SQL procedure files	300

If you plan to install procedure files for *other* products on this disk, refer to the appropriate documentation and adjust the disk size accordingly.

Recommended value: 4000 4K blocks (minimum size)

TEMP disk

In addition to the MDISK requirements listed above, a TMP disk is needed to hold the software archive downloaded from the Rocket Model 204 web site. Once the installation has completed, the TMP disk may be discarded.

Recommended value: 9000 4K blocks (minimum size)

Defining the MODEL204 directory entry

MODEL204 machine

The MODEL204 service machine is the machine used to run Model 204 and store Model 204 database files. The MODEL204 machine is accessed by Model 204 users when running Model 204. You can have more than one service machine; for example, you may want to establish a second service machine like MODEL204 for a test environment.

System file definitions

If your site has system files defined for an earlier release of Model 204, you do not need to redefine them.

Minidisk addresses

You can assign any addresses to these minidisks. However, because Model 204 performs I/O scheduling based on the virtual configuration presented to it, follow the usual channel and control-unit separation procedures. The overriding consideration for address assignment is the real configuration of these devices. Space management considerations may dictate using separate minidisks for the operational files. The MODEL204 minidisk requires allocating not more than 50 1K blocks, excluding operational files.

Sample MODEL204 directory definition

The following example is a sample entry in the User Directory for a Model 204 service virtual machine with a user identifier of MODEL204. Make sure that items in italics, such as *password*, conform to your site parameters and standards. Enter items not in italics exactly as shown.

```

USER      MODEL204  password minstorage maxstorage  G
ACCOUNT   account  distcode
OPTION    MIH MAXCONN num
MACHINE   mode
IUCV      ALLOW
IPL       CMStype PARM AUTOCR
CONSOLE   009    3210 T userid
SPOOL     00C    2540 READER A
SPOOL     00D    2540 PUNCH A
SPOOL     00E    1403 A
LINK      CMSowner 190 190 RR
LINK      CMSowner 19E 19E RR
LINK      M204maint 193 193 RR
LINK      M204maint 195 195 RR
MDISK     191 type start size volser MR
MDISK     vcu1 type start size volser MWV
MDISK     vcu2 type start size volser MWV
    
```

USER statement

The required virtual machine storage size varies according to the parameters that are specified to Model 204.

The following example is a formula for calculating a rough estimate of the region Model 204 requires in a CMS environment. Increase the estimated size to allow storage of the CMS nucleus and file directories. Refer to the *Model 204*

Command Reference Manual for more information about the NUMBUF, NUSERS, NSERVS, and SERVSZ parameters.

```
Region size = load module size + (NUMBUF * 6184)
+ (NUSERS * 10240) + (NSERVS * SERVSZ)
```

Under z/VM, the load module size is zero if a configuration of Model 204 executing in a discontinuous saved segment (DCSS) is being used. Otherwise, you must include the size of the load module and the size of the CMS Interface load module to determine the required virtual machine size. You can define storage beyond that which is used by Model 204 with no perceptible side effects on either Model 204 or overall system performance, provided that the additional storage does not affect the ability to employ z/VM shared segments within the service virtual machine.

OPTION statement

You can specify the MAXCONN and CPUID operands in the OPTION statement. (The OPTION statement in the sample Model 204 User Directory does not include the CPUID operand in its definition.)

- **MAXCONN operand**—If you expect more than four users to connect to the service machine, reset MAXCONN *num* to a number greater than the number of users.
- **CPUID operand**—The identification of a z/VM service virtual machine that runs Model 204 employs the virtual machine user identifier, the processor serial number stored by the STIDP instruction, and a system management facility (SMF) system ID of *CMSn* for the file enqueueing. From a file enqueueing point of view, each service machine is treated as a separate CPU.

If you are using the CPUID operand of the OPTION statement in a Model 204 service machine's User Directory entry, or the Control Program SET CPUID command, one Model 204 service machine can operate on a number of separate real processors with the same file enqueueing identification.

This facility is useful for emergency backup or similar situations. Under normal circumstances, only one service machine with a particular combination of user identifier and processor serial number should be active on the processors in a multiple-CPU complex.

The format for the CPUID operand is:

```
CPUID processor-serial-number
```

where *processor-serial-number* is a 6-digit number.

MACHINE mode statement

Specify the mode of the virtual machine.

IUCV statement

You must include the IUCV ALLOW statement, because the service machine needs to support IUCV connections for other machines.

IPL statement

Specify the type of CMS machine being IPLed.

LINK statements

CMSowner is the name of the virtual machine where system-wide files are kept, usually MAINT.

M204maint is the name of the virtual machine where Model 204 system-wide files are kept, usually MAINT204.

In the example, the files used to access Model 204 from z/VM are made available by linking to the MAINT204 193 and 195 disk. You can tailor the actual owner and address of the disk that contains these files, as well as the link address, to meet your site requirements.

MDISK statements

The Model 204 service machine uses CMS minidisks to hold its EXEC procedures and other files related to the operation of Model 204 under z/VM.

Use the *mmm* and *nnn* minidisks for storing the Model 204 database application and Model 204 system files. These minidisks can be either variable-format or CMS-format disks. However, you must allocate Model 204 files on variable-format disks.

Creating Model 204 system files

You need to create a number of files to serve as Model 204 system files. Model 204 uses these files as it is running, performing recovery, or loading files.

Table 2-2 provides summary information to help you determine the size and format to use for these Model 204 system files.

Table 2-2. Model 204 system files

DDname	Function	Usage	Shared?	Disk format
CCAAUDIT	Audit trail of system activity	High. Recommend use of AUDIT204 to obtain copy from CCAJRNL.	No	Variable or CMS
CCAGRP	Group file definitions; no DCB required	Low; number of group file definitions, plus three pages for FCT; 61 definitions/page for average of nine files/group	Yes	Variable

Table 2-2. Model 204 system files (Continued)

DDname	Function	Usage	Shared?	Disk format
CCAIN	User 0 input	Low to medium (read-only)	Yes	CMS
CCAJLOG	Audit trail information (unformatted)	Can be high; is dependent on system activity	No	Variable
CCAJRNL (CCARF)	Journal file for recovery and auditing purposes. DCB: <ul style="list-style-type: none"> • LRECL: 6749 • RECFM: U • DSORG: PS 	Can be high; is dependent on system activity	No	Variable
CCAPRINT	User 0 output	Low; recommend a virtual printer	No	Variable or CMS
CCASERVER	Server swap area; largest server data set; no DCB required	High	No	Variable
CCASNAP	Snap dumps	Low; dependent on number of snaps taken; virtual printer recommended	No	Variable
CCASTAT	Access control or security file	Low	Yes	Variable
CCASYS	System file	High	Yes	Variable
CCATEMP	Working storage; disk buffering sort space; precompiled code; no DCB required	High; average of 25 pages/user plus 200 pages per SUBSYSTEM; page=PAGESZ of 6184	No	Variable
CHKPOINT (RESTART)	Update checkpoint file; ROLLBACK preupdate; page images stored, plus checkpoint records; DCB: <ul style="list-style-type: none"> • LRECL: 6184 • RECF: F (B) • DSORG: PS 	High	No	Variable or CMS
SORT5	Deferred updates for FLOD	High	No	Variable or CMS
TAPE1	Input data for FILELOAD	High	No	Variable or CMS
TAPE2	Deferred updates for FLOD	High	No	Variable or CMS
TAPE3	Deferred variable length updates for FLOD-7	High	No	Variable or CMS

Table 2-2. Model 204 system files (Continued)

DDname	Function	Usage	Shared?	Disk format
TAPE5	FRV deferred updates for FLOD	High	No	Variable or CMS

File loading procedures

Files may be loaded using a one-step procedure, FILELOAD, or a multistep procedure, such as a five- or seven-step FLOD.

The SORT5, TAPE1, TAPE2, TAPE3, and TAPE5 files are used in file load invocation of Model 204. Refer to the *Model 204 File Manager's Guide* for more information about FILELOAD.

Defining the GCS directory entry

This section describes how to define the Group Control System (GCS) service machine directory entry. You need to do this only if you will be using the VTAM Interface or Horizon Interface. If this section does not apply to you, go to "Defining the user directories" on page 17, which describes how to define the user directories.

Sample GCS service machine directory definition

The following sample shows a z/VM directory entry for the CMS/VTAM Interface service machine. Items in italics, for example, *password*, should conform to your installation configuration parameters and standards. Enter items not in italics exactly as they are shown.

```
USER M204VMVT password 5M 6M G
ACCOUNT account distcode
OPTION MIH ECMODE MIH MAXCONN nnn
MACHINE mode
```

* Allow any virtual machine to use server.

```
IUCV ALLOW MSGLIMIT 255
IPL GCS PARM AUTOLOG
CONSOLE 009 3270
```

* For z/OS add:

```
NAMESAVE GCS
```

```
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
```

* Provide links so that IPL CMS is possible.

```
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
```

* LINK to disk with PROFILE and M204VMVT LOADLIB.
LINK MAINT204 193 191 RR

OPTION statement

The OPTION MAXCONN *nnn* statement specifies how many concurrent IUCV paths this virtual machine can open. The CMS and VTAM Interfaces use one IUCV path per opened VTAM access control block (ACB).

Set the MAXCONN number high enough to accommodate all ACBs that are opened at once. Usually this is at least two per ONLINE virtual machine: one for terminal handling (IODEV=7), and one for each link the system manager defines for the Online. Use a number somewhat higher than you currently need to avoid changing this directory entry each time a new ACB is added.

LINK statements

The 191 disk of the GCS service machine is read-only. Link to the MAINT204 193 disk where the M204VMVT LOADLIB and sample PROFILE GCS are installed.

IUCV ALLOW service machine

The sample directory entry authorizes any virtual machine in the system to connect to an IUCV ALLOW service machine. You can restrict connections to specifically authorized virtual machines by omitting IUCV ALLOW and using IUCV user ID statements instead. Be sure to add the MSGLIMIT 255 operand.

For more information about the IUCV statement and other options, refer to the appropriate IBM documentation.

VTAM GCS group

The z/VM system programmer must allocate a slot in the VTAM GCS group for this service machine. This machine should not be in the authorization list; it is intended to run entirely in problem state.

Defining the user directories

You need to define a directory for each CMS user of Model 204.

The user machine includes links to the maintenance machine (MAINT204) or to a system disk that provides access to the M204 EXEC.

Sample user directory definition

The sample below shows a user machine directory entry. Items in italics should conform to your installation configuration parameters and standards. Enter items not in italics exactly as they are shown.

```
USER      user01 password 2M 2M G
ACCOUNT   account distcode
IPL       CMS PARM AUTOCR
CONSOLE   009 3210 T userid
SPOOL     00C 2450 READER A
SPOOL     00D 2450 PUNCH A
LINK      M204maint 193 193 rr
LINK      M204maint 195 195 rr
SPOOL     00E 1403 A
LINK      CMSowner 190 190 rr
LINK      CMSowner 19E 19E rr
MDISK     191 type start size volser MR
```

LINK statements

193 is the virtual address of the disk that contains the files that general users use to access Model 204 in z/VM, usually MAINT204.

CMSowner is the name of the virtual machine where system-wide files are kept, usually MAINT.

M204maint is the name of the Model 204 server machine where Model 204 system-wide files are kept, usually MAINT204. The 193 and 195 disks are the virtual addresses of the disks containing the files that general users use to access Model 204 in z/VM.

Setting up the Cross-Reference facility service machine

The Dictionary/204 Cross-Reference facility (XREF) produces reports in batch mode. In z/VM, this requires a service machine, which is similar in function to the service machine required by a Model 204 Online.

This section explains how to set up the XREF service machine and create an appropriate PROFILE EXEC.

If you are not installing Dictionary/204, or if you are reinstalling Dictionary/204, these procedures are not necessary. You can skip to “Initializing disks” on page 20.

XREF service machine: directory definition

A z/VM system programmer at your installation must set up the service machine according to these guidelines:

- Two megabytes is the recommended size for the service machine's 191 disk.
- Set up the z/VM user ID for the XREF service machine as an autologin ID.
- Use any name for the XREF user ID. Be sure to insert the XREF user ID in your CCAIN stream. See "Modifying CCAIN" on page 20.
- The only permanent disks that need to be linked are the MAINT204 disk and the disk where the standard sort utility is stored at your z/VM installation. Link the sort disk in the service machine directory entry.
- If MAINT204 is also linked in the directory, there is no need to include Step 2 of the following PROFILE EXEC.

Sample PROFILE EXEC for the service machine

The PROFILE EXEC performs the preliminary steps necessary to run cross reference jobs:

1. Allocates work space by linking to a temporary disk.
2. Links to the MAINT204 disk, where the ALERT EXEC and the M204XREF and M204LIST modules reside.
3. Runs the ALERT EXEC.

Comments (marked by /*....*/) explain optional lines and site-dependent parameters.

```
/* Sample Profile Exec */
/* */
/* 1. Get work space by linking to a temporary */
/* work disk. Doing this (as opposed to using */
/* the XREF 191 disk as a work area) makes it */
/* easier to experiment with number of */
/* cylinders required for Cross Reference */
/* jobs at your site. */
/* Change "T3390" to the correct volume at your */
/* site, "200" to an available address for */
/* a work disk, "CYL 20" to estimated */
/* work space needed. */
/* Do not change file mode "D" unless you make a */
/* corresponding change in the XREF SYSTEM EXEC. */
```

```
`CP DEFINE T3390 200 CYL 20`
```

```
if rc =92 & rc = 0 then exit rc
push 'WRKDSK'
push 'YES'
'FORMAT 200 D'
if rc = 0 then exit rc

/* 2. Link to the MAINT204 disk. Unnecessary if */
/* this disk is linked in the XREF userid's */
/* directory entry */

'CP LINK MAINT204 193 193 RR READ'
if rc = 0 then exit rc

'ACCESS 193 C'
if rc = 0 then exit rc

/* 3. Invoke the ALERT EXEC (a loop executed */
/* while waiting for reader input). */

push 'EXEC ALERT'
exit
```

Modifying CCAIN

When a z/VM user ID has been established for XREF, insert the following Model 204 command in your CCAIN job stream:

```
DEFINE PUNCH XREFSERV WITH SCOPE=SYSTEM -
ROUTER=VM ID=xrefid NAME=M204XREFEXEC
```

where *xrefid* identifies the user ID established for XREF.

Initializing disks

After defining the Model 204 maintenance machine, MAINT204, and service machine, MODEL204, you need to initialize the CMS-format minidisks.

Changing access from R/R to R/W

In order to format the disks, the disk access must be read/write. Use the MAINTACC EXEC to change any R/R disks to R/W. See Appendix B for more information.

CMS-format disks

Use the CMS FORMAT command to initialize the CMS-format disks.

Use the FORMAT command for all the MAINT204 minidisks, because all distributed files are loaded on CMS-format disks. The Model 204 191 disk should be a CMS-format disk. However, all, or almost all, of the remaining Model 204 files are variable-format.

The use of CMS-format disks for the MODEL204 service machine, other than the 191 disk, is not recommended due to the possibility of reduced performance. See Appendix C for more information.

Variable-format disks

Link to the Model 204-owned minidisks that are to contain any variable-format files and initialize them by using the M204UTIL INITIAL function. It is assumed that space allocation on variable-format minidisks is done within the CMS environment.

The initialization process *destroys all data* on the volumes. The format of the M204UTIL command for minidisk initialization is:

```
M204UTIL INITIAL cuu label (options...)
```

For more information about M204UTIL syntax and options, see Appendix B.

3

Installing the Model 204 Nucleus

In this chapter

- Installation checklist
- Contents of the Model 204 VMARC archive
- Downloading the installation archive
- Modifying the CPUIDZAP ZAP file
- Decrypting optional features
- Customizing MAINT204 and MODEL204 profiles
- Specifying TXTLIBs
- Preparing user exits
- Preparing site-specific CUST, CDTB, FUNU, and MSGU modules
- Preparing FLOD exits
- Customizing parameter values (M204USER PARMS)
- Generating modules and saved segments

Installation checklist

This chapter describes how to install the Model 204 nucleus. Before completing the tasks presented in this chapter, you must prepare the z/VM environment, as described in Chapter 2.

This installation checklist summarizes the steps you perform to install the Model 204 nucleus.

Step	Task	Page
1.	Download Model 204 installation archive file and run the installation program. Modify the CPUIDZAP file.	page 25 page 27
2.	Use the M204CRYP command to decrypt optional features such as DBCS.	page 27
3.	Customize the MAINT204 maintenance virtual machine profile.	page 28
4.	Customize the MODEL204 service virtual machine profile.	page 29
5.	Link the MAINT204 maintenance virtual machine to the system disks that contain the SORT, FORTRAN, COBOL, and LE TXTLIBs.	page 29
6.	Prepare user-written accounting, security, and allocation exits.	page 30
7.	Prepare site-specific CDTB, FUNU, and MSGU modules for translation tables, functions, and error messages.	page 35
8.	Prepare FLOD exits—modify the Online loadlist, adjust the %SLC statement, and set FILETYPE to TEXT for the OI15CMS and OI35CMS exits.	page 38
9.	Modify the M204USER PARMS file to customize the configuration defaults.	page 39
10.	Generate modules and saved segments from text decks with the M204GEN command.	page 45

Contents of the Model 204 VMARC archive

Table 3-1 lists the contents of the Model 204 VMARC.

Table 3-1. Contents of the Model 204 archive files

Archive File	Unpacks files to this minidisk	Contents
DISK191	191	<ul style="list-style-type: none"> DISKnnn EXECs MAINTACC EXEC MAINPROF EXEC
DISK193	193	<ul style="list-style-type: none"> EXEC and CCAIN files to install and operate Model 204 and add-on products Sample ONLINE EXEC for Model 204 execution HELP files for Model 204 modules and EXECs SERVPROF EXEC M204TRCE TEXT

Table 3-1. Contents of the Model 204 archive files

Archive File	Unpacks files to this minidisk	Contents
DISK194	194	<ul style="list-style-type: none"> • Model 204 object (TEXT) modules • Encrypted object modules for separately purchased features • M204CRYP EXEC and related routines; use M204CRYP to decrypt the encrypted object modules • M204GEN and M204SAVE EXEC files, which create the Model 204 module and saved segments (DCSS or SSP/MSS) and CNTRL files used to assemble the source files • Materials used by M204GEN to generate Model 204 modules • Model 204 maclibs • Sample source files for user-written exits • CPUIDZAP ZAP skeleton file
DISK195	195	<ul style="list-style-type: none"> • TXTLIBs used in generating Host Language Interface programs and the CMS Interface • M204 EXEC and the files used by M204 EXEC
DISK200	200	<ul style="list-style-type: none"> • Dictionary • Installation files • Demonstration database files, distributed in Model 204 DUMP format (identified by the file type M204DUMP)
DISK205	205	<ul style="list-style-type: none"> • SAS C runtime library

Downloading the VMARC file archiving module

Model 204 is provided in a VMARC archive format, which requires the VMARC module to unpack the files.

If you do not already have the VMARC module on your system, you can download it from IBM.

1. Navigate to the IBM downloads web page for VMARC:
<http://www.vm.ibm.com/download/#downvmarc>
2. Scroll up the page and read the license agreement.
3. Follow the downloading instructions on the web page.

Downloading the installation archive

Procedure

Model 204 is available in a VMARC archive on the FTP server.

1. Go to the following web site:
<ftp://ftp.cca-int.com/>
2. Enter the username and password supplied to you by Rocket.
3. Open the zVM folder and download the vmarc file to your PC in binary

format.

4. Create a 50 cylinder temporary disk on MAINT204 as disk B.

The following code shows a sample EXEC to create the temporary disk:

```
&TRACE OFF
&ERROR &EXIT &RETCODE
CP DEF T3390 AS 291 CYL 50
&STACK YES
&STACK TINST
FORMAT 291 B
&EXIT
```

5. Transfer the vmarc file on your PC to the B disk and name the file CMrrrn VMARC, where rrr is the release level and *n* is a numeric 0-9. Be sure to transfer the file in binary format.
6. If you already have a version of Model 204 installed, remove previous installation files from disks.
 - If disks 193, 194, 195, 200, and 205 contain files pertaining to the previous release, delete any existing files from these disks.
 - If disks 193, 194, and 195 are read only, change them to read/write by reaccessing them:

```
ACC 193 C
ACC 194 D
ACC 195 E
```

Unpack installation program

The INSTALL EXEC will access the appropriate minidisks and unpack the individual VMARC archives. Unpack the INSTALL EXEC by entering the following:

```
VMARC UNPK CMrrrn VMARC B INSTALL EXEC B
```

for example:

```
VMARC UNPK CM7400 VMARC B INSTALL EXEC B
```

Unpack installation files

Execute the installation program with the following command:

```
EXEC INSTALL
```

You will see the installation files being loaded onto the appropriate minidisks.

Unpack individual archives (optional)

If you need to unpack files to individual minidisks, use the following command:

```
VMARC UNPK CMrrrn VMARC B DISKnnn VMARC B (REPLACE
```

For example:

```
VMARC UNPK CM7400 VMARC B DISK195 VMARC B (REPLACE
```

This creates a file on the B disk called:

```
DISK195 VMARC B
```

To list the files in the DISK195 archive:

```
VMARC LIST DISK195 VMARC B
```

To load the files to the 195 disk (E disk):

```
VMARC UNPK DISK195 VMARC B * * E (REPLACE
```

Modifying the CPUIDZAP ZAP file

Rocket generates and sends you a CPUID zap created for your individual site.

Use the zap you receive from Rocket to modify the skeleton CPUIDZAP ZAP file on the 194 disk.

You do not need to change the first line in the skeleton CPUIDZAP ZAP file:

```
NAME module KOMM$
```

When the module is generated with the M204GEN command, the correct module name is inserted and the zap is automatically included.

Decrypting optional features

The Model 204 installation contains all the object code required to use separately purchased options. The object modules that enable the use of these optional features are distributed in encrypted form. You must decrypt the appropriate options before running the M204GEN command:

Syntax

The format of the M204CRYP command is:

```
M204CRYP DECODE feature-name key ([NOREPLACE | REPLACE])
```

Where

- *feature-name* is the name of the feature. The valid feature names are:

Feature name	Purpose
DBCS	Double-byte character set

- *key* is the 8-byte numeric key that decrypts the object modules for the specified feature. The key is provided by Rocket.
- *NOREPLACE*, the default, checks that files having the same file ID as the output file do not already exist. If a file with the same file ID exists, an error message is displayed and the M204CRYP command terminates.
REPLACE causes the output file to replace an existing file with the same file identifier.

For details on the generated report and error messages, see “M204CRYP command” on page 140.

Customizing MAINT204 and MODEL204 profiles

MAINT204 profile

You need to create a PROFILE EXEC for the MAINT204 maintenance virtual machine that reflects several site-specific variables, such as links to a common tools disk or inclusion of CP and CMS SET commands. You can use the sample profile that is included with the Model 204 installation software as a basis for customization. The name of the sample profile is MAINPROF EXEC.

Customizing the MAINT204 profile

To customize the MAINT204 profile, perform these steps:

1. Rename the MAINPROF EXEC by issuing the command:

```
RENAME MAINPROF EXEC A PROFILE = =
```

2. Modify the PROFILE EXEC, which is on the MAINT204 A disk, to conform to your site requirements. The MAINPROF EXEC executes the MAIN-TACC RR command, which accesses the 193, 194, and 195 minidisks in read-only mode.
3. Run the customized PROFILE EXEC by issuing the command:

```
EXEC PROFILE
```

MODEL204 profile

The MODEL204 service virtual machine is usually logged on by a system or automatic operator using the AUTOLOG command. Thereafter, its operation normally requires no attention. The PROFILE EXEC on this machine performs these functions:

- Initiates console spooling
- Accesses the 193 and 195 minidisks
- Executes the ONLINE command to run Model 204

Customizing the MODEL204 profile

You can use the sample profile that is included with the Model 204 installation software as a basis for customization. The name of the sample profile is SERVPROF EXEC.

Use the same general procedure as you used for customizing the MAINT204 profile (see “Customizing the MAINT204 profile”). However, you need to copy the SERVPROF EXEC from the MAINT204 193 (C) minidisk to the MODEL204 191 (A) minidisk.

Recommendations

Follow these recommendations for customizing the MODEL204 profile.

- Add this statement to your MODEL204 profile to avoid producing unnecessary VMDUMPs:

```
SET AUTODUMP OFF
```

- In a z/OS environment, saved segments can exist in the virtual machine address space. For saved segments *not* loaded immediately after an IPL, reserve space for the saved segment by adding this statement to your MODEL204 profile:

```
SEGMENT RESERVE M204ONLN
```

Otherwise, if the space is already used, the load fails.

If you have a test machine as well, you might want to modify the statement in the test machine’s profile as follows:

```
SEGMENT RESERVE T204ONLN
```

If you issue the SEGMENT RESERVE statement for M204ONLN or T204ONLN, the segment is purged after the service machine terminates (EOJ) for M204ONLN/T204ONLN.

- The service virtual machine can be logged on by a system or automatic operator using the AUTOLOG command. Thereafter, its operation normally requires no attention.

Specifying TXTLIBs

You need to link the MAINT204 maintenance machine to the system disks that contain these TXTLIBs:

Library	Purpose
SORT TXTLIB	Used for operation of Model 204. See “SORT utility” on page 30 for additional details.

Choose **one** of the following. Either FORTRAN TXTLIB or LE TXTLIB is required for load module generation:

Library	Purpose
FORTRAN TXTLIB	Contains the FORTRAN mathematical functions used by the Model 204 mathematics functions and Audit module. The prerequisite is version 2.6.
LE TXTLIB	Contains the LE mathematical functions used by the Model 204 mathematics functions and Audit module.

You must also update the M204USER PARMs file to include the name of these TXTLIBs at your site. The M204USER PARMs file is used as input to the M204GEN command, which needs to access the correct TXTLIB libraries.

SORT utility

It is your responsibility to provide a utility program that sorts index entries associated with file loading. SyncSort CMS software from Syncsort, Inc. functions correctly with Model 204 and meets the requirement for possible active multiple concurrent sorts during a 5-step FASTLOAD.

Other SORT utility programs might function correctly, but they have not been tested. You cannot use CMS SORT to load deferred Ordered Index updates.

To find out which SYNCSORT release you have, issue this command:

```
EXEC SYNCLEVL
```

TXTLIBs and sort exits OI15CMS and OI35CMS

The sort exits OI15CMS and OI35CMS are provided with the installation software.

They reside on the D(194) disk with a file type of TXT nnn , where nnn is the version/release. For example, the file type for Version 7.4.0 is TXT740.

Typically, they are placed in a TXTLIB or reside on an accessed disk with a file type of TEXT.

Preparing user exits

Model 204 enables you to use user-written exits and modules.

To install any combination of the installation-specific facilities, the text files for the associated modules must be available on an accessed disk. These exits are linked into the M204CMS module.

Assembling ACCTEXIT, SCTYEXIT, and ALLOEXIT user exits

To assemble the ACCTEXIT, ALLOEXIT, and SCTYEXIT exits, use the VMFHLASM EXEC supplied by IBM and the NCMS CNTRL file from the Model 204 installation software.

AMODE settings

The AMODE of the ACCTEXIT and ALLOEXIT exit routines is called in the same AMODE as the ONLINE and IFAM1 modules with which they are running. When M204CMS initializes, it determines the AMODE of the ONLINE or IFAM1 application with which it runs. For example, if M204CMS loads and calls an ONLINE generated with the XA parameter of the M204GEN command, M204CMS always calls the ACCTEXIT and ALLOEXIT routines in AMODE 31.

Required file names and file types

You must use the following file names and file types for the user-written exits. These names are automatically generated when you assemble the exits using VMFHLASM.

- ACCTEXIT TXT nnn (for system accounting purposes)
- ALLOEXIT TXT nnn (for data set allocation control)
- SCTYEXIT TXT nnn (for security relating to file opening)

The TXT nnn file type represents the current release; for example, TXT740 is V7R4.0. The exits are described in this section.

If you do not use these exits

If you do not use these exits, issuing the M204GEN command causes you to receive one or more messages indicating that the exit is not found. You can ignore the messages.

ACCTEXIT: accounting exit

The ACCTEXIT exit is included with the Model 204 installation software. Model 204 accounting is described in the *Model 204 System Manager's Guide*. A user-written accounting exit can be incorporated in the interface to Model 204 to process the statistics records that Model 204 generates.

To install an accounting exit, the TEXT file for the ACCTEXIT module must be available on an accessed disk and have the correct file type. This exit is installed when you issue an M204GEN command with the CMS configuration specified.

z/OS basic mapping support macros

ACCTEXIT should not use CMS file system macros to perform I/O to CMS files. Use z/OS basic mapping support (BSAM) macros (OPEN, CLOSE, READ, WRITE, and CHECK) for I/O to z/OS, z/VSE, or CMS disks.

LOGOUT records and since-last records

To activate accounting record production, specify at least one of these parameters on User 0's parameter line:

Parameter	Creates...
SMFLORN	LOGOUT records.
SMFSLRN	Since-last records. SMFSVC is fixed at SVC 241.

Accounting exit calling conventions

Each time Model 204 writes a statistics record, the CMS Interface to Model 204 invokes a routine called ACCTEXIT. Table 3-2 lists the calling conventions.

Table 3-2. Accounting exit calling conventions

Register	Contents
GRP 1	Address of SMF record
GRP 10	CMSWORK address
GRP 13	OS-format save area
GRP 14	Return address
GRP 15	Address of ACCTEXIT

Guidelines for accounting exits

The exit processes the accounting record and should do this as quickly and simply as possible. Take care to preserve the CMS environment in which Model 204 is executing. In particular, the exit must use standard z/OS linkage conventions for General Registers 13 and 14. The contents of any other register can be destroyed.

If the discontinuous saved segment (DCSS) version of the Model 204 CMS Interface is used, the exit must not be self-modifying.

AMODE 31 or 24

The ACCTEXIT exit can execute in either AMODE 31 or 24. It is important to exit ALLOEXIT in the same AMODE in which you entered it.

SCTYEXIT: security exit

The handling of protection indicators located in the data set control block for a z/OS data set conforms to z/OS conventions.

If a write-protected data set is opened for output, or if a read-protected data set is opened in any mode, an abnormal termination occurs. Model 204 intercepts

this abnormal termination condition and restarts the user whose attempted action caused the violation.

The decision to permit data sets to be opened is performed by a security exit routine, SCTYEXIT, and is part of the CMS Interface for Model 204. A sample version of SCTYEXIT is included with the installation software. You can substitute a user-written security exit routine.

Replacing the default security exit

The source for the default version of SCTYEXIT is distributed as SCTYEXIT ASSEMBLE.

To replace the default security exit, the TEXT file for the SCTYEXIT module must:

- Be available to the maintenance virtual machine.
- Reside on a disk that is accessed ahead of the service virtual machine's 194 disk, such as the service virtual machine's 191 disk, and have the correct file type.

Security exit calling conventions

Table 3-3 lists the calling conventions for SCTYEXIT.

Table 3-3. Security exit calling conventions

Register	Contents
GRP 0	Address of 44-character data set name, or zero for a CMS file
GRP 1	Address of data set control block (DSCB) for a z/OS data set, file system control block (FSCB) for a CMS file, or complement of data extent block (DEB) address for a previously opened z/OS data set
GRP 3	Open parameter list entry for DCB
GRP 10	CMSWORK address
GRP 13	Address of standard z/OS-format save area
GRP 14	Return address
GRP 15	Address of SCTYEXIT

Guidelines for security exits

This exit is installed when you issue an M204GEN command with the CMS configuration specified. The security exit sets the return code to indicate whether or not to access the file.

- A return code of zero indicates that access is permitted.

- A return code of nonzero indicates that access is denied.

If the discontinuous saved segment (DCSS) version of the Model 204 CMS Interface is used, the exit must not be self-modifying.

AMODE 24

Because the SCTYEXIT exit always gets control in AMODE 24, no changes are required to run a SCTYEXIT exit with Model 204. However, if you are upgrading from a previous version of Model 204, you must reassemble SCTYEXIT with the new maclibs to pick up CMS Interface and CMS operating system control block changes.

ALLOEXIT: allocation exit

Model 204 supports dynamic allocation via the ALLOCATE and FREE commands, as described in the *Model 204 System Manager's Guide*. These commands cause Model 204 to issue an SVC 99 instruction to perform the allocation. The CMS Interface to Model 204 performs SVC 99 simulation. You can incorporate a user-written allocation exit into the interface to validate allocation requests.

Allocation exit calling conventions

Each time Model 204 executes an ALLOCATE or FREE command, the CMS Interface invokes the ALLOEXIT routine to Model 204. Table 3-4 lists the calling conventions.

Table 3-4. Allocation exit calling conventions

Register	Contents
GRP1	Address of SVC 99 request block
GRP 10	CMSWORK address
GRP 13	Address of standard z/OS-format save area
GRP 14	Return address
GRP 15	Address of ALLOEXIT

Guidelines for allocation exits

If the saved segment version of the Model 204 CMS Interface is used, the exit must not be self-modifying.

To install an allocation exit, the TEXT file for the ALLOEXIT module must be available on an accessed disk and have the correct file type. Issue the M204GEN command and specify CMS as the operand.

For information about the SVC 99 request block format, refer to the appropriate IBM documentation. The allocation exit sets the return code to indicate whether or not to allow the allocation to proceed.

- A return code of zero indicates that allocation is to be performed
- A return code of nonzero indicates that allocation is to be prevented.

AMODE 31 or 24

The ALLOEXIT exit can execute in either AMODE 31 or 24. It is important to exit ALLOEXIT in the same AMODE in which you entered it.

Preparing site-specific CUST, CDTB, FUNU, and MSGU modules

You can also incorporate installation-defined Date Time stamp customization, translation tables, functions, and error messages. The associated modules are CUST, CDTB, FUNU, and MSGU, respectively. Mechanisms for constructing translation tables and functions are described in the *Model 204 System Manager's Guide*.

It is recommended that if you think your site will make use of the TCP/IP Sockets Interface for Model 204, link the FUNU module during the installation process, even if you have not yet developed your own functions. This ensures access to the \$ASCII and \$EBCDIC functions.

Modifying \$functions

The FUNU module contains any user-written \$functions. Once you have installed Model 204, all code receives control in 31-bit mode. See the appropriate CMS manual for a description of the differences between 24-bit and 31-bit mode operation.

Note: These notes do not provide complete descriptions of all internal data structure and algorithm changes that have been made to Model 204; nor are they to be construed as official sanction for the use of a specific interface or of the dependency on any specific data structure or algorithm.

Rocket Software takes no responsibility for user-written \$functions, including the \$functions distributed by the User Group. Please be sure to test carefully any user-written \$functions you use before your system goes into production.

Converting \$functions for z/VM 31-bit machines

The following changes are required to run user \$functions in 31-bit mode. All such interfaces, data structures, and algorithms remain subject to change without notice:

- If needed, convert pointers to data structures, such as servers and PCBs, to 31-bit addresses. If you have 3-byte address constants (for example, AL3(xyz)), you must change them, because they can no longer be used as pointers to such data structures.
- Modify any code that explicitly manipulates Model 204 4-byte string codes to remove the high-order byte from the string code prior to adding KUPVT or KUPST.
- If you invoke a z/OS service that must receive control in 24-bit mode (such as BSAM) use CSAVE or IOSAVE with the AMODE=24 parameter. Code CRSTR or IORSTR with the AMODE=31 parameter. You may not reference the pushdown list between AMODE=24 and AMODE=31 range.
- Use the NOARG parameter for ARG statements in 31-bit mode. NOARG = *label* syntax is described in the *Model 204 System Manager's Guide*. You can also specify NOARG = * to return control to the next sequential instruction after the ARG statement.
- You cannot issue the CCALL macro in 24-bit mode.

Implementing the date/time stamp feature

A date/time stamp (DTS) feature has been added to Model 204 so that you can easily discern when a record in a file was last changed.

Your site may customize:

- The date/time stamp field name
Model 204 is shipped to you without a default date/time stamp field name. You can define the date/time stamp field name in the CUST source code or on User 0's parameter line by setting the new DTSMFN parameter.
- The field definition
With file manager privileges, you can define the date/time stamp field for each file you want enabled, and subsequently you must set the FOPT parameter to include X'10'. Please note the FOPT=X'10' value cannot be set until the date/time stamp field has been defined to the file.
- The data that is inserted into this field
Model 204 is shipped to you with a default date/time stamp format that combines the system DATE and TIME fields. If your site prefers a different system field or a different format, see "Customizing the date/time stamp field name in CUST source code" on page 37.

Installing the date/time stamp feature

Installation of the date/time stamp (DTS) feature does not require a decryption key.

The CUST source member is located in CUST ASSEMBLE on the 194 disk, where it can be edited to make changes such as those described in “Customizing the date/time stamp field name in CUST source code” on page 37.

Customizing the date/time stamp field name in CUST source code

The DTSFN value is established by the DTSINIT routine in the CUST object code. As shipped, this routine does not provide a default field name. You may customize the CUST source code to establish a date/time stamp field name. Edit DTSINIT to change the following line:

```
CUSTFN DC 0C
```

to the desired field name, for example,

```
CUSTFN DC C'MY DATE TIME STAMP FIELD NAME CONTAINING SPACES'
```

The maximum length for a field name is 255 characters.

Default instructions in the routine move the system DATE and TIME into the field in the format:

```
CCYYMMDDHHMMSSMMM
```

Table 3-5. Date/time stamp format

Where	Specifies	For example: 20110915163510345
<i>CC</i>	Century	20 for 21st century
<i>YY</i>	Year	11 for 2011
<i>MM</i>	Month	09 for September
<i>DD</i>	Date	15 for the fifteenth day of the month
<i>HH</i>	Hour on a 24-hour clock	16:00 hours or 4:00 PM
<i>MM</i>	Minutes past the hour	35
<i>SS</i>	Second past the minutes	10
<i>MMM</i>	Milliseconds	345 past the seconds

Making modifications to the default date/time stamp value

If the default data that is supplied in the DTSUB routine in CUST is not what you want or require, the source code for the DTS field value is available for modification. This field area of up to 255 bytes may be completely formatted as your site desires.

Assembler statements to build the data in Table 3-5 that comprise the date/time stamp field values are placed in the DTSUB entry point in the CUST module. This entry point contains instructions that build a field value in the area pointed to by T3 (Rocket register notation) upon input. Register T1 contains packed decimal date in form of 0CYDDDF and register T2 contains the time of day in milliseconds since midnight.

- You are not required to use the *ccyymmddhhmmssmmm* format for your DTS file date. You may change the format of the date and time data to suit your site requirements.
- You may customize the date/time stamp field to include flags or other system data you deem necessary for your site.
- You could add a suffix to the default data, such as USERID or ACCOUNT.
- The default format and data content is DISPLAY; however, you could use Binary or Float instead—up to 255 bytes of data.
- There is no requirement for any position to have a special meaning to Model 204 internal routines.

Note: Each site is responsible for whatever changes they make to the DTSUB entry point. Rocket does not support changes beyond those documented in “Installing the date/time stamp feature” on page 36.

Assembling CUST, CDTB, FUNU, and MSGU

To assemble the CUST, CDTB, FUNU, or MSGU modules under z/VM, use VMFHLASM EXEC supplied by IBM and the NCMS204 CNTRL file from the Model 204 installation software.

For example, you can issue this command:

```
VMFHLASM FUNU NCMS204
```

Preparing FLOD exits

You can modify the record image upon which FLOD or FILELOAD commands act. For additional information, refer to the FLOD exit feature in the *Model 204 File Manager's Guide*.

Modifying the ONLINE loadlist

To generate the ONLINE module with the user-written FLOD routines, for example, FLODXT0, FLODXT1,... FLODXT10, an entry for each routine must exist in the ONLINE loadlist, as shown here:

```
&1 &2 FLODXT0  
&1 &2 FLODXT1
```

Choosing module or saved segment

The ONLINE loadlist that you need to modify depends on whether you are generating ONLINE as a saved segment or module; refer to “Loadlists” on page 172 for information about loadlists:

Modify loadlist...	To generate ONLINE as...
LOADONLN	Module
LSXAONLN	Saved segment in a z/OS environment

Using save segments

If you are using saved segments and the exits are reentrant, insert the entry in the shared portion of the segment; otherwise, insert the entry in the nonshared portion. Refer to “Reentrant and non-reentrant code” on page 171 for more information on reentrant code.

Adjusting the %SLC statement

If you increase the size of the non-reentrant portion of a saved segment to accommodate FLOD exits, you must adjust the %SLC statement in the ONLINE loadlist and the saved segment definitions.

See “%SLC statement syntax” on page 173 for more information on adjusting the %SLC statement.

Setting FLOD exits OI15CMS and OI35CMS

For the OI15CMS and OI35CMS FLOD exits to work, you must set the file type to TEXT for each exit.

Customizing parameter values (M204USER PARMS)

Two parameter files, M204CCA PARMS and M204USER PARMS, are included with the Model 204 installation software; these files supply configuration defaults for Model 204. To modify configuration defaults for your site:

Step	Action to take...	Purpose...
1	Review M204CCA PARMS file	Determine if the defaults are appropriate to your site.
2	Edit M204USER PARMS file	Modify configuration defaults for your site.

Note: You *must not modify* the M204CCA PARMS file.

The FASTLOAD, M204, and ONLINE EXECs must have access to these parameter files.

M204CCA PARMS file

The M204CCA PARMS file contains:

- Default values used by the M204GEN and M204SAVE commands
- Defaults for TXTLIBs
- Default destination for module generation

The DEFAULTS section is used in generating or saving modules and segments and is read by the M204, FASTLOAD, and ONLINE EXECs when bringing up an Online or linking to one.

The configuration sections, such as ONLINE, are used to configure Model 204 modules. Each section contains the name of the loadlists for modules and segments, origins for modules, and so on.

The TXTLIB section contains distributed default names of TXTLIBs. See “Examples of modifying M204USER PARMS” on page 44 for information on how to change these values.

M204CCA PARMS file example

This sample includes default values distributed with the commercial release of Model 204.

```
M204GEN:
DEFAULTS:
  ldrtbls = 35
  prefix.mod.prod=M204      Prefix for PROD module
  prefix.mod.test=T204     Prefix for TEST module
  prefix.mod4seg.prod=X204  Prefix of module saved as PROD seg.
  prefix.mod4seg.test=Y204  Prefix of module saved as TEST seg.
  prefix.seg.prod=M204     Prefix of PROD segment
  prefix.seg.test=T204    Prefix of TEST segment
  dflt.cntrl = ncms204    Standard CMS CNTRL filename
  destination = 193 C     Disk on which to generate modules
END.DEFAULTS:
.
.
.
TXTLIB:
  fortlib = VSF2FORT  FORTRAN textlib
  sortlib = SYNCSORT  SORT textlib
END.TXTLIB
```

M204USER PARMS file

You can modify the values in the M204USER PARMS file. The values in the M204USER PARMS file take precedence over the M204CCA PARMS values. The contents of the M204USER PARMS file are used by these EXECs:

- FASTLOAD
- M204
- M204GEN
- M204SAVE
- ONLINE

M204USER PARMS file example

The M204USER PARMS file contains three main sections:

- M204SSEG.XA
- M204 Operations
- PREV

Each section begins with a section indicator, for example, M204GEN:, and terminates with an end indicator, for example, END.M204GEN:.

Comments are indicated by two asterisks (**). Comments can begin in any column and can be coded to the right of parameter statements.

This sample file, M204USER PARMS, establishes the PREV family, which is an alternative to PROD or TEST. This sample includes default values distributed with the commercial release of Model 204 and should be modified according to your CMS environment.

```
M204GEN:
DEFAULTS:
    prefix.mod.prev=P204          Prefix for PREV module
    prefix.mod4seg.prev=V204     Prefix for module saved
                                as PREV segment.
    prefix.seg.prev=P204        Prefix for PREV segment
END.DEFAULTS:
.
.
.
M204SSEG.XA:
M204ONLN:
    origin = 310000    31 values used to save ONLINE
                    segment for PROD
```

```

key      = 0
rent     = 310-5FF
nonrent  = 600-6FF
rstd     = N
spname   = ONLINE
END.M204ONLN:
.
.
.
P204ONLN:
origin   = 310000    31 values used to save ONLINE segment
key      = 0         for PREV
rent     = 310-5FF
nonrent  = 600-6FF
rstd     = N
spname   = PONLINE
END.P204ONLN:
.
.
.
PROD:           Model 204 PROD configuration
linechan=m204vmio  Line mode channel name
                  (IODEV 39)
fschan=m204vmfs   Full screen channel name
                  (IODEV 41)
subset=          String to get into SUBSET
login=login      Attempt login with CMS ID
disco=disconnect Disconnect command string
cmd=            M204CMND filename
END.PROD:

PREV:           Model 204 PREV (Previous
                release)
linchan=prevvmio Line-by-line channel name
user=m204prev    Service machine name
fschan=prevvmfs  Full screen channel name
subset=*cms      Command to invoke CMS subset
login=login      Attempt login with CMS ID
disco=disconnect Disconnect command string
cmd=            M204CMND filename
END.PREV:
END.M204:

```


M204SSEG.XA section

The M204SSEG.XA section defines the attributes of Model 204 saved segments—M204CMS, M204IFM1, M204ONLN, and M204USR—for use in a z/VM environment.

Note: For the PREV configuration, it is necessary to create entries to specify the saved segment attributes for CMS, IFM1, ONLN, and USR.

Saved segment definitions include the attributes listed in Table 3-6.

Table 3-6. Attributes of saved segment definitions

Attribute	Specifies...
Key	Storage key for the reentrant section of the segment. This must always be zero.
Nonrent	Page range of the non-reentrant section of the segment. Specify the page numbers in hexadecimal. Use one 3-digit number or two 3-digit numbers, separated by a hyphen, if there are two numbers, with no embedded blanks. If there is no non-reentrant code, specify a zero.
Origin	Load point of the saved segment: the virtual storage address where the segment is loaded.
Rent	Page range of the reentrant section of the segment. Specify the page numbers in hexadecimal. Use two 3-digit numbers, separated by a hyphen, with no embedded blanks.
Rstd	Whether use of the saved segment is restricted. Virtual machines require the NAMESAVE directory control statement to use a restricted saved segment. Specify a Y, N, or leave it blank (equivalent of an N).
Spname	SPACE names, separated by blanks, that include this saved segment, thus defining it as a member saved segment (MSS). Space names must be fewer than eight characters long.

M204 Operations section

The M204 Operations section contains the defaults for the M204 EXEC. The “M204USER PARMS file example” on page 41 demonstrates the use of the parameter file for establishing the values required by the familiar PROD family of saved segments. Each statement corresponds to an option you can specify on the M204 command. For more information on the meaning of each option, refer to the *Model 204 System Manager’s Guide*.

PREV section

Entering the command:

```
M204 PREV LINE
```

is equivalent to issuing:

```
M204 USER M204PREV CHAN PREVVMIO LINE SUBSET *CMS -  
DISCO DISCONNECT LOGIN
```

Entering the command:

```
M204 PREV DISPLAY
```

is equivalent to issuing:

```
M204 USER M204PREV CHAN PREVVMFS DISPLAY SUBSET -  
*CMS DISCO DISCONNECT LOGIN
```

Precedence for resolution of option conflicts

You can specify M204 options on the command line, in the M204USER PARMS file, or in the M204 EXEC.

It is recommended that you modify M204USER PARMS. The order of precedence for option resolution is:

1. Command options override any other options.
2. M204USER PARMS file options override options set in M204 EXEC.
3. The `init_opts` subroutine in M204 EXEC establishes basic defaults.

Examples of modifying M204USER PARMS

If you want to generate modules on the 205 disk accessed as G, then add this line to the DEFAULTS section:

```
destination = 205 g
```

Disk must have at least one resident CMS file

The disk specified by the optional destination parameter must have at least one resident CMS file. If the disk is empty, all generated files are placed on the disk accessed as A.

BETA as a saved segment

In addition to TEST and PROD, you can permit the use of BETA as a saved segment to refer to prerelease versions of Model 204 software. To do so, perform these steps:

1. Add three lines to the DEFAULTS section to specify the prefix for the BETA module to be saved as a segment (the order is important):

```
prefix.mod.beta=b204
prefix.mod4seg.beta=a204
prefix.seg.beta=b204
```

2. Generate B204ONLN, B204CMS, B204USR, and B204IFM1 entries in the saved segment section for XA (M204SSEG.XA). To do this, duplicate and modify existing lines.
3. Add an entry in the M204 Operations section, using the labels `BETA:` and `END.BETA` to delimit the values for `linechan`, `fschan`, and so on.

Changing TXTLIB defaults

If your site uses different file names from the defaults for any of these TXTLIBs, you must modify the M204USER PARMS file.

In the TXTLIB section near the beginning of the M204USER PARMS file, change any of the following defaults, if necessary:

```
TXTLIB:
  fortlib  = VSF2FORT
  sortlib  = SYNCSORT
END.TXTLIB:
```

For example, you might change `VSF2FORT` to the name of the CMS file for the FORTRAN TXTLIB at your site. If you are including the LE mathematics functions, change the `fortlib` value to the name of the CMS file for the LE TXTLIB at your site.

Generating modules and saved segments

Installing the Model 204 installation software loads a number of text decks onto the MAINT204 (maintenance) virtual machine. Then you must generate modules or saved segments from those text decks by using the M204GEN command. M204GEN syntax is described in Appendix B. The M204GEN command performs these tasks:

- Applies the CPUIDZAP zap.
- Concatenates text decks.
- Generates MODULE and MAP files on the MAINT204 virtual machine's 193 disk.
- Loads those concatenated files into storage.

Using the M204GEN command defaults

The files loaded from the Model 204 installation software include several files that provide default values used by the M204GEN command to generate Model 204 modules and segments that work in most z/VM environments. For details about running M204GEN, see "M204GEN command" on page 151.

If you choose to use those defaults, then simply issuing the following commands generate all the modules and saved segments:

Command	Generates all the configurations...
M204GEN ALL	Except IFAM1 as modules. The modules generated are production modules for an IBM environment for the version of CMS in which you issued the M204GEN command.
M204GEN ALL (SEG	That can be generated as saved segments. To generate a specific configuration as a segment and a module, if applicable, or for a test or production environment, issue separate M204GENs for each specific configuration.

Saving segments

Use the M204SAVE command to save a segment for which a module was previously generated. The M204GEN command automatically calls the M204SAVE command, if you specify the SAVE option for M204GEN. If you do not specify SAVE for M204GEN, then you must issue the M204SAVE command explicitly for the modules or segments specified. For details on using the M204SAVE command, see “M204SAVE command” on page 154.

Use the NOSAVE option of M204GEN for those segments that you want to generate as member saved segments (MSSs); this is the default for M204CMS, M204IFM1, and M204ONLN. By generating the module files without saving segments, you avoid redefining the saved segments.

You *must* resave segments with M204SAVE after you apply early warnings.

Procedure to generate and save segments

1. Issue the M204GEN command for segments.
2. Issue these commands:

```
DEF STOR 16M
IPL CMS
ACC (NOPROF
SET LDRTBLS 35
DEFSEG temp 310-6FF SR
SEGMENT RESERVE temp
SEGMENT RELEASE temp
Q NSS NAME temp
PUR NSS NAME temp
```

where *temp* is any name used to identify space temporarily saved.

3. Issue the M204SAVE command.

Copying modules to the 195 disk

To make them available to all users, copy these modules from the 193 disk to the 195 disk:

- M204BAT2
- M204INFO
- M204USR
- M204USR LOADLIB
- M204XFER

Preparing the GCS server component

The GCS server component contains the following elements:

GCS Service Machine Directory Entry	See "Sample GCS service machine directory definition" on page 16.
M204VMVT LOADLIB	This file contains the program that runs the VT204 command. The VT204 command controls the CMS/VTAM Interface GCS service machine.
PROFILE GCS	GCS service machine uses this profile to start the VT204 command automatically at IPL time.

VTAM Interface

This section applies only if your site is installing the Model 204 VTAM Interface. After using M204GEN to generate the ONLINE module, perform these steps to complete the installation of the optional VTAM Interface.

Defining the VTAM network

Make sure that the VTAM network administrator at your site has completed the VTAM network definitions, consistent with the discussion below.

An APPL statement in VTAMLST for each IODEV type (IODEV=7, IODEV=37) is required for direct VTAM terminal support by Model 204. The one- to eight-character APPL names are used as the values for the VTAMNAME (full screen) and VTAMNTO (line-at-a-time) parameters within the User 0 CCAIN lines.

Model 204 has no further requirements regarding the APPL statements or other network definition statements. For example, network concerns alone should determine the setting of logmode table entries.

Outbound messages, inbound messages

- Model 204 supports both definite response and exception response protocols on messages outbound to the terminal.
- Requests for SNA definite responses to messages coming inbound from the terminal are not supported.
- For definite response protocol on outbound messages, the '02' value can be removed from the TERMOPT parameter setting on each IODEV=7 statement that refers to the terminal.

Horizon Interface

This section applies only if your site is installing the Horizon/Limited Edition (Horizon/LE) Interface.

After you have used M204GEN to generate the ONLINE module, complete Horizon/LE installation by defining the Horizon/LE network. To do this, use DEFINE commands to specify links, processgroups, and SQL processes.

See the *Model 204 Horizon: Intersystem Processing Guide* for information on defining Horizon connections. See the *Model 204 SQL Connectivity Guide* for information on defining Horizon/LE connections.

Parallel Query Option/204

If you are using the distributed processing module of Model 204, known as Parallel Query Option/204 (PQO), consult the *Parallel Query Option/204 User's Guide*.

4

Completing the Installation

In this chapter

- Overview
- Allocating Model 204 files
- Allocating the demonstration database
- Modifying EXECs and CCAIN streams
- Creating password and group data sets
- Applying early warnings
- Bringing up the Model 204 Online
- Bringing down the Model 204 Online
- Accessing the Model 204 Online
- Using standalone Model 204 utilities
- Error tracking

Overview

This chapter describes steps you perform to complete the Model 204 installation. This chapter also explains how to operate Model 204 in different modes, to make sure that the installation successfully supports any of these modes used at your site.

If you are upgrading from an earlier release of Model 204

If your site has already installed an earlier release of Model 204, you probably already have established test procedures. The methods for operating Model 204 in different modes have not significantly changed with this version; see the *Model 204 Release Notes* for details. Therefore, if you have experience installing earlier releases of Model 204, you can skip over the testing-related section of this chapter, “Accessing the Model 204 Online” on page 60 and perform testing based on your experience.

Model 204 operating modes

Model 204 can be run in several different modes, including:

- IFAM1 and IFAM2 Host Language Interface
- Multiuser Online
- Single-user Online (interactive or batch)
- Single-user IFDIAL
- Standalone utilities

Defaults assumed

The procedures described in this chapter assume that you are using:

- M204DEF EXEC, the SAMPDEF EXEC from the installation
- Default CCAIN stream, the SAMPONLN CCAIN from the installation

ONLINE command

The ONLINE command creates a Model 204 multiple-user Online environment. The ONLINE EXEC performs these functions:

- Executes a user-created EXEC procedure to define the file recovery environment (optional)
- Executes a user-created EXEC procedure to define the Online environment
- Invokes Model 204 to establish the Online environment
- Invokes Model 204 to perform file recovery (optional)

The ONLINE command is used in several of the testing modes discussed in this chapter. See the *Model 204 System Manager's Guide* for more detailed information about the ONLINE command.

Backup facilities

Use the ONLINE command to create a single-user environment in which the appropriate commands to perform backup functions are executed. Alternatively, perform backup functions with a production Online system as required. To perform a backup, use the Model 204 DUMP command, as documented in the *Model 204 Command Reference Manual*. If your site writes backups to magnetic tape, see the *Model 204 File Manager's Guide* for information on performing multitape backup.

Allocating Model 204 files

After you load the Model 204 installation software and generate modules and segments, use the following steps to allocate Model 204 files associated with the MODEL204 service machine:

1. Use the M204UTIL command with the INITIAL parameter to initialize variable-format minidisks.
2. Access the minidisks with the appropriate file modes.
3. Use the M204UTIL command with the ALLOCATE parameter to allocate operational work files on the initialized disks; see Appendix B.
4. Allocate the operational files and Model 204 demonstration database.
5. Check allocations.

M204UTIL ALLOCATE and the ZBLDTAB EXEC

Use the M204UTIL ALLOCATE function to preallocate a Model 204 password table (CCASTAT) on a z/OS-format disk before running the ZBLDTAB EXEC to initialize CCASTAT. The initialization procedure is described under "Creating the password data set" on page 56.

Access minidisks with appropriate file modes

After executing M204UTIL INITIAL for each minidisk, access it as the file mode you will use for future reference by issuing this command:

```
ACCESS cuu fm
```

where:

- *cuu* denotes virtual address
- *fm* denotes file mode

Allocating operational work files

Use the following format of the M204UTIL ALLOCATE function to allocate the Model 204 system files, which are needed to operate the Online and Model 204 utilities.

```
M204UTIL ALLOC dsn fm (PRIMARY n units
```

where:

- *dsn* denotes data set name
- *fm* denotes file mode
- *n units* denotes a number followed by CYL, BLK, or TRK, for example, 10 CYL.

Refer to Table 2-2 on page 14 for a list of the files you need to allocate.

Note: You do *not* need to allocate the SORT5 and the TAPEI–TAPE5 files.

If you use M204UTIL to allocate the space on a minidisk that is defined to a guest operating system such as z/OS or z/VSE, the VTOC entry is defined, but the catalog (z/OS) is not updated. You might, therefore, want to allocate those data sets with the facilities of the guest system.

Allocating the demonstration database

The Model 204 installation software contains a demonstration database, which is used for training, product demonstrations, and documentation examples. Table 4-1 lists the demonstration database files.

Table 4-1. Demonstration database files

File	Model 204 pages (6184 bytes/page)	Description
DAILY	200	Updateable VEHICLES and CLIENTS
VEHICLES	128	Record for each insured auto
CLAIMS02	48	Accident claims for 2002
CLAIMS03	48	Accident claims for 2003
CLIENTS	216	Policyholder and driver records
DUMMY1	120	Dummy string table
MEMBERIN	104	Member information
MONTHIN	104	Month number and abbreviation
PRODUCTN	104	Product information
SALESDAT	104	Sales data

Table 4-1. Demonstration database files

File	Model 204 pages (6184 bytes/page)	Description
SCATTER	32	Scatter chart data
STAFF	216	Employee information
STATES	1016	States and mapping data
STOCKDAT	104	Stock data
STOCKDET	216	Stock details
STOCKIN	216	Stock information
STOCKMON	216	Stock monthly information
TEAMINFO	104	Sales team information

The correct allocation estimates depend on the device type you intend to use, as shown in Table 4-2.

Table 4-2. Model 204 pages per device type

Device type	Maximum block size	Blocks per M204 page	M204 pages per track
FBA	512	13	--
3380	47476	--	7
3390	56664	--	8
9345	46456	--	6

Calculate the space requirements for each file from the number of pages as shown in Table 4-2. To allocate these databases, use M204UTIL ALLOC. The CLIENTS file, for example, requires 32 tracks for a 3380 device type. You allocate the CLIENTS file by entering:

```
M204UTIL ALLOC M204 CMS CLIENTS fm (PRIMARY 32 TRK
```

where:

fm indicates the file mode

Checking allocations

To check allocations for the operational work files and demonstration databases, use the M204UTIL LIST function as follows:

```
M204UTIL LIST * fm
```

where:

- LIST function displays all the files allocated on the minidisk with the specified file mode
- *fm* indicates the file mode

Modifying EXECs and CCAIN streams

The Model 204 installation software includes samples of EXECs used to run Model 204. You need to tailor these EXECs to your site's requirements.

Note: Do *not* modify any Model 204 EXECs with the prefix of CCA. Those EXECs are used for internal processing by Model 204.

Modifying sample EXECs

Modify sample EXECs using the following procedure.

Note: M204CRYPT makes these disks READ ONLY. You might need to access them again as READ/WRITE.

1. Copy the SAMPDEF EXEC and SAMPONLN CCAIN files from the 193 (C) disk to the 195 (E) disk.

```
COPYFILE SAMPDEF EXEC C M204DEF EXEC E
COPYFILE SAMPONLN CCAIN C ONLINE CCAIN E
```

The M204DEF EXEC file defines the files that are used in an Online execution. The ONLINE CCAIN file defines User 0 parameters for Online execution.

2. In the M204DEF EXEC make these changes:
 - Search for all occurrences of *ACCESS ???*, and change the question marks to appropriate disk addresses.
 - Also change file modes *O*, *P*, or *Q* if you are using different ones.
 - The FILEDEFs for the work files and database files contain a question mark (?) for the file mode of the minidisk on which files reside. In each case, change the ? to the appropriate file mode.
 - Define CCASNAP as *one* of the following:

```
PRINTER
A CMS disk file
DUMMY
```

3. Copy SAMPREST EXEC and SAMPRECV CCAIN files from the C (193) disk to the E (195) disk. The EXEC and CCAIN files define the environment and parameters for recovery execution:

```
COPYFILE SAMPREST EXEC C M204REST EXEC E
COPYFILE SAMPRECV CCAIN C RECOVERY CCAIN E
```

4. The M204REST EXEC file defines the files used in an Online recovery

execution. The RECOVERY CCAIN file defines the User 0 parameters and recovery command for Online recovery execution. In the M204REST EXEC, change the question marks as you did in the M204DEF EXEC.

5. The M204 EXEC, by default, assumes the use of a saved segment version of the Model 204 TP Interface program. If you want to use nucleus extension or the user area (MODULE), then add the following lines to the M204USER PARMS file.

```
M204 :
PROD:
    run_location = location_name
END.PROD:
END.M204:
```

where *location_name* is either NUCEXT or UAREA. You can use either of these two keywords to override the default.

6. You might also want to modify the CCAIN files, which define User 0 parameters for Online operation and recovery. For more information, refer to the *Model 204 System Manager's Guide*.

M204MOUN EXEC

You might need to tailor the sample M204MOUN EXEC supplied with the Model 204 installation software to your installation's environment.

M204SPL EXEC

You can tailor the sample M204SPL EXEC supplied with the Model 204 installation software to your site's environment.

M204UNLD EXEC

You can tailor the sample M204UNLD EXEC supplied with the Model 204 installation software to your site's environment.

Creating password and group data sets

You need to create passwords for Model 204 users. If your site plans to use file groups in its Model 204 applications, then you need to create group data sets. If your site has password and/or group data sets already defined for an earlier release of Model 204, you do not need to perform the tasks in this section.

For more information about CCASTAT or CCAGRP, see the *Model 204 System Manager's Guide*.

Support for expiring passwords

Optionally, you can create an alternate CCASTAT file to support expiring passwords. You may choose to run job ZCTLTAB to create a second CCASTAT data set, which would be used if this support is desired. A standard CCASTAT data set is used as input to create this second file.

Parameters exist to set the number of days till expiration, start of warning messages, length of grace period before the password is purged, minimum password length, maximum password length, number of previous passwords to check and whether users are required to enter previous passwords.

z/OS-format disk

For a z/OS-format disk, use the M204UTIL ALLOCATE function to preallocate a password data set (CCASTAT) before running the ZBLDTAB EXEC.

Example

```
M204UTIL ALLOC M204 SYSTEM CCASTAT fm (PRIMARY 5 TRK
```

where:

fm indicates the file mode

Creating the password data set

Run ZBLDTAB (refer to Appendix B) to initialize the password data set CCASTAT with the user SUPERKLUGE and password=PIGFLOUR:

```
ZBLDTAB M204CMS CCASTAT fm
```

Creating the CCAGRP file

If you plan to use file groups, create a CCAGRP file with these steps:

1. Allocate a data set for CCAGRP, for example:

```
M204UTIL ALLOC M204 SYSTEM CCAGRP fm (PRIMARY 5 TRK
```

2. Create a CMS file named CREATEG CCAIN. Insert these commands into CREATEG CCAIN:

```
PAGESZ=6184  
LOGON SUPERKLUGE  
PIGFLOUR  
CREATEG  
LOGOFF  
EOJ
```

3. Create an EXEC from the SAMPDEF EXEC and name it CREATEG EXEC. CCAJRNL, CCASERVR, and CHKPOINT files are not necessary; therefore, remove the FILEDEFS.

Change the file name on the FILEDEF for the CCAIN ddname to CREATEG CCAIN. Change the name and file mode on the CCAGRP FILEDEF.

4. Change the SYSOPT parameter from 154 to 152.
5. CREATE a CCAGRP file by executing the command:

```
ONLINE NODCSS BYPASS CREATEG
```

Applying early warnings

The Autofix system includes maintenance early warnings for Model 204, as well as associated products, such as Dictionary/204. You might need to apply the early warnings for Model 204 when you install it.

The Autofix system is available for download.

1. Go to the following web site:

```
http://www.rocketsoftware.com/m204/support
```

to access the Technical Support page.

2. Navigate to: Technical Support > Model 204
3. Enter your username and password.
4. From the list of choices presented, select Autofix, Early Warnings, and Late Breaking News > Autofix EWnnnn: Available for download as of ...
5. Then select the operating system at your site: z/OS, z/VM, or z/VSE.
6. On the operating system page you can select to download the Autofix application, Autofix documentation, and backup files.

For details on how to install and execute the Autofix system, refer to the *Autofix Installation and Operations Guide*.

Applying early warnings with Autofix

Maintenance changes generally are distributed through the Autofix system as either a zap, a source update, or an object replacement.

1. Before changing a module, create a backup copy.
2. To apply an early warning, access the 193 disk in read/write mode by using MAINTACC RW.
3. After you apply the maintenance, use the M204SAVE command to save any affected DCSS.

Do not use the IBM ZAPTEXT command to apply Model 204 early warnings to text files, unless instructed to do so by Technical Support or by an early warning.

Table 2-5 on page 153 shows the correspondence between Model 204 load module identifiers, CMS MODULE files, and saved segment names.

Bringing up the Model 204 Online

When you first install Model 204, you must bring up the Model 204 ONLINE module (or saved segment) using the following procedure. Refer to the previous section for information about the ONLINE command, which you use to bring up the Online. Once you have created the environment by doing this, the Online can be accessed (connected to) using the methods described in the rest of this chapter.

Procedure

To bring up the Model 204 ONLINE module, perform the following steps.

It is assumed that you have modified the SAMPDEF EXEC and renamed it M204DEF EXEC.

1. From MAINT204, restore the demonstration database files—add FILEDEF statements to the M204DEF EXEC to define the dump files. Verify that you are linked to the disk that contains the Model 204 DUMP files from the installation. Then add FILEDEFs in this format:

```
FILEDEF DUMPCLNT DISK CLIENTS M204DUMP F
FILEDEF DUMPCL02 DISK CLAIMS02 M204DUMP F
FILEDEF DUMPCL03 DISK CLAIMS03 M204DUMP F
FILEDEF DUMPVEH DISK VEHICLES M204DUMP F
FILEDEF DUMPDLY DISK DAILY M204DUMP F
FILEDEF DUMMYDMP DISK DUMMY1 M204DUMP F
FILEDEF MEMDMP DISK MEMBERIN M204DUMP F
FILEDEF MONTHDMP DISK MONTHIN M204DUMP F
FILEDEF PRODDMP DISK PRODUCTN M204DUMP F
FILEDEF SALES DMP DISK SALES DAT M204DUMP F
FILEDEF SCATDMP DISK SCATTER M204DUMP F
FILEDEF STAFFDMP DISK STAFF M204DUMP F
FILEDEF STATEDMP DISK STATES M204DUMP F
FILEDEF STOCKDMP DISK STOCKDAT M204DUMP F
FILEDEF STOCKDDP DISK STOCKDET M204DUMP F
FILEDEF STOCKIDP DISK STOCKIN M204DUMP F
FILEDEF STOCKMDP DISK STOCKMON M204DUMP F
FILEDEF TEAMDMP DISK TEAMINFO M204DUMP F
```

2. Log on to MODEL204, the service machine:

```
LOGON MODEL204
<password>
```

3. If not performed by the MODEL204 service machine profile, access the 193 and 195 disks of the maintenance machine as B and C, respectively:

```
ACCESS 193 B
```



```
ACCESS 195 C
```

4. To bring up the Online, enter one of two commands:

```
ONLINE BYPASS M204DEF (if using saved segments)
```

```
ONLINE NODCSS BYPASS M204DEF (if using modules)
```

Use the BYPASS operand to avoid running recovery in the initial run, because there are no CCAJRNL and CHKPOINT files to use in the recovery step.

5. At this point, you can disconnect from the MODEL204 service machine by entering the command:

```
#CP DISCO
```

Note: Do *not* log off. If you do, the Online terminates and you must run it again.

Running recovery in subsequent runs

To run recovery in subsequent runs, enter one of the following commands. This assumes you have modified the SAMPREST EXEC and renamed it M204REST EXEC:

- If you are using saved segments:

```
ONLINE M204REST M204DEF
```

- If you are using modules:

```
ONLINE NODCSS M204REST M204DEF
```

Bringing down the Model 204 Online

You need to bring the Online down whenever you switch Model 204 operating modes. For example, if you access the Online in multiuser mode and then want to access the Online in single-user mode, you must bring the Online down before accessing it in single-user mode.

Procedure

To bring down the Model 204 Online:

1. Connect or reconnect to the MODEL204 service machine.
2. Be sure that other users are logged out.
3. Enter the following HALT reply sequence to stop ONLINE operation:

```
END OF DAY  
EOJ
```

Accessing the Model 204 Online

You can test the Model 204 installation by accessing it in a variety of ways. This section describes how to access the Model 204 Online in these modes:

- Multiuser
- Interactive single-user
- BATCH2
- IFAM1 and IFAM2
- Single-user IFDIAL

Multiuser Online

Model 204 can serve multiple users simultaneously. One copy of Model 204 supplies central database services to a number of users who can access those services in various ways.

Interactive access to Model 204 from z/VM permits a CMS user to communicate with a Model 204 that is executing in another CMS virtual machine.

Figure 4-1 illustrates the architecture of the Model 204 multiuser environment.

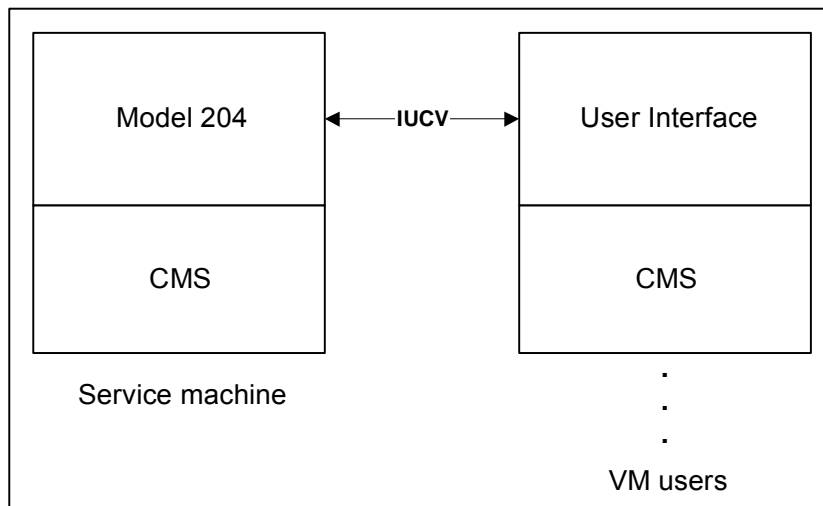


Figure 4-1. Model 204 multiuser service machine architecture

The user interface executes in a virtual machine and accesses Model 204 services interactively.

Procedure

After you have brought up the Online, complete these steps:

1. Log on to a z/VM user ID.

2. Enter this Model 204 command to access the Online that is running in the MODEL204 (service) virtual machine:

```
M204 USER MODEL204    (if using a saved segment)
```

3. Log on to the Online as SUPERKLUGE to add new users with the appropriate privileges and to create database files:

```
LOGON SUPERKLUGE
PIGFLOUR
```

4. Issue the CREATE command to create a file and the RESTORE command to restore the file from the dump file. Do this step for each demonstration database file as illustrated here:

```
CREATE FILE filename
END
OPEN filename
RESTORE 128 FROM dumpdd
```

using the values for *filename* and *dumpdd* in this list:

filename	dumpdd
CLIENTS	DUMPCLNT
CLAIMS02	DUMPCL02
CLAIMS03	DUMPCL03
VEHICLES	DUMPVEH
DAILY	DUMPDLY
DUMMY1	DUMMYDMP
MEMBERIN	MEMDMP
MONTHIN	MONTHDMP
PRODUCTN	PRODDMP
SALESDAT	SALESDMP
SCATTER	SCATDMP
STAFF	STAFFDMP
STATES	STATEDMP
STOCKDAT	STOCKDMP
STOCKDET	STOCKDDP
STOCKIN	STOCKIDP
STOCKMON	STOCKMDP
TEAMINFO	TEAMDMP

5. Most of the distributed files are public and updated by default with no password. The following files are semipublic, with default privileges of read-only access:

CLAIMS02, CLAIMS03, CLIENTS, VEHICLES.

To open a semipublic file for update after it has been restored, the system manager must create passwords with update privileges for each file.

Example

In this example, MANAGER is established as a password for the VEHICLES file with a privilege setting of X'BFFF'. Since no values are provided for CLASS, SELECT, READ, UPDATE, and ADD, the default value of 0 will be used:

```
LOGCTL A :VEHICLES 1
*** ENTER FILE/GROUP PASSWORD, PRIVILEGES, CLASS, SELECT, READ, UPDATE, ADD
MANAGER, x' BFFF '
*** ENTER TERMINAL LIST, ALL, NONE, ADD, DEL, OR RETURN
ALL
```

The absence of spacing around punctuation is important; remove spaces around punctuation as shown in this example.

For more information about passwords, privileges, and the LOGCTL command, refer to the *Model 204 System Manager's Guide*.

6. When you are finished using the Model 204 Online, issue the LOGOUT or LOGOFF command.

Interactive single-user mode

In single-user mode, several users running Model 204 at the same time are independent of each other and of other users. In this mode, Model 204 uses the User 0 input from CCAIN, and CCAPRINT for output.

At the end of the CCAIN stream, Model 204 switches to the console (set by the ALTIODEV parameter) for both input and output.

Figure 4-2 illustrates the architecture of the single-user environment.

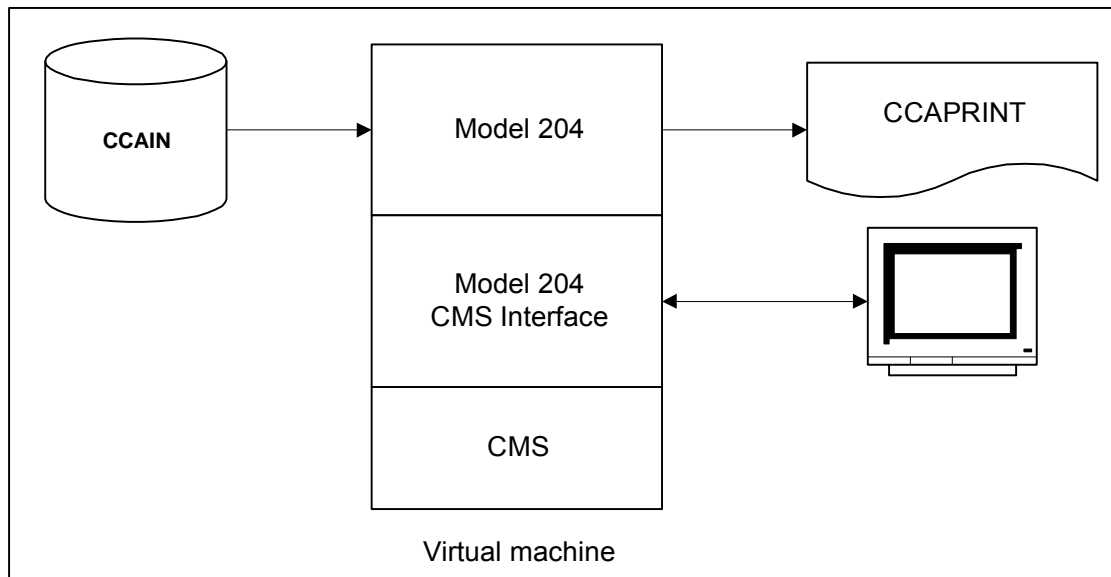


Figure 4-2. Model 204 single-user architecture

Model 204 executes in an individual user's virtual machine and accesses private and shared public databases interactively.

Procedure

To interactively access shared public databases:

1. Log on to a valid z/VM user ID.
2. Copy the SAMPDING EXEC and CCAIN files and name them SINGLUSR.
3. Customize the files as you did earlier for SAMPDEF. If NUSERS=1, then you cannot include any IODEV statements.
4. While you might want to share the CCASTAT file with the service machine, allocate separate files for recovery.

You might also want to allocate temporary disk space for CCATEMP.

5. Execute this M204 command:

`M204 USER * (if using a saved segment)`

or one of the following commands:

`ONLINE SINGLUSR (if using a saved segment)`

`ONLINE NODCSS BYPASS SINGLUSR (if using a module)`

6. After successful testing, issue the following commands to bring down the single-user operation:

`LOGOUT`

`EOJ`

M204 USER * command

The M204 USER * command performs two actions:

- The ALTIODEV parameter is automatically set to 47 (if the hardware supports full-screen mode), which causes the console to be treated as a full-screen device. If you are using a line-at-a-time device, then ALTIODEV is automatically set to 45.
- When the end of the CCAIN stream is reached, assuming that no HALT or EOJ command was specified in the CCAIN stream, Model 204 switches to the console for input. To end the session, you must enter an EOJ command at the console.

BATCH2 mode

Model 204 does not provide batch facilities directly under z/VM. You can use the service virtual machine environment in a non-interactive single-user mode to run Model 204 batch application programs. This mode is called BATCH2. For more discussion about the BATCH2 facility refer to the *Model 204 User Language Manual* "BATCH2 facility" section.

In most batch applications, Model 204 reads its input from a disk file, rather than interacting with a user at the terminal. The primary difference between this mode of operation and the more traditional batch environment is that when simultaneous Model 204 environments are active, multiple virtual machines are in use.

Figure 4-3 depicts this environment.

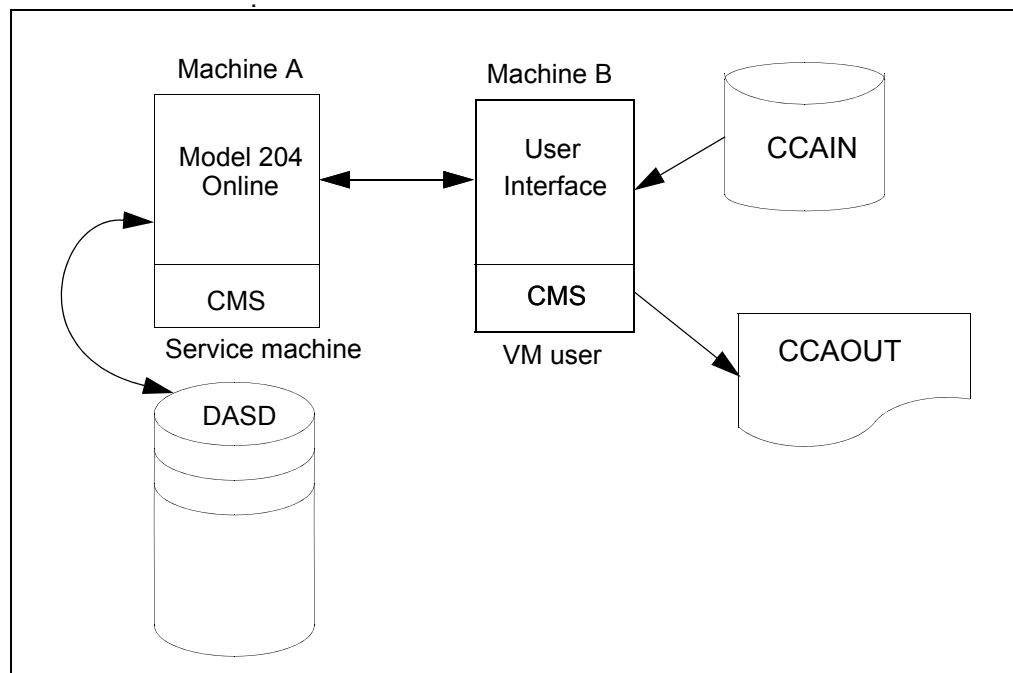


Figure 4-3. BATCH2 application architecture

Procedure To start batch processing, see the BATCH2 section in the *User Language Manual* for information on how to invoke it.

IFAM1 and IFAM2 modes

This section describes the implementations of the IFAM1 and IFAM2 host language environments under z/VM. For both environments, you need to supply a host language program to test these modes.

Using IFAM1 mode

IFAM1 accesses Model 204 files via a single thread from a host language program. Model 204 and the application program execute together in a single virtual machine. The application program module is loaded dynamically by Model 204.

After link-editing the application program(s), execute a host language application by using the IFAM1 command.

Syntax IFAM1 [TEST | PROD] *execname*

Where

- TEST specifies that a test version of the Model 204 IFAM1 shared segment (T204IFM1) is to be invoked. If you do not specify TEST, the production version (M204IFM1) is used.

Running an IFAM1 job with the test saved segment is useful:

- To test zaps before applying them to the production version of Model 204
- If your site has two different versions of Model 204 (for example, Version 6.1.0 and Version 7.1.0), and one uses the M204IFM1 saved segment while the other uses the T204IFM1 saved segment
- *execname* specifies the name of an EXEC procedure that provides the file definitions and other logic required to execute Model 204 with the IFAM1 host language application. You must provide the IFAM1 program module name by the specified EXEC procedure and place it in the stack, via the &STACK statement for EXEC2 or PUSH statement for REXX.
 - A return code of zero from the EXEC invokes the application program.
 - A return code of one (1) bypasses the invocation of the program.
 - Any other return code is considered an error condition report and terminates the IFAM1 EXEC.

Refer to the *Model 204 Host Language Interface Reference Manual* for detailed information about running IFAM1 jobs.

Using IFAM2 mode

IFAM2 provides multithread access to Model 204 files from a host language program. The Model 204 Online that is executing in a service virtual machine communicates with host language programs running in one or more other z/VM virtual machines. The Model 204 files accessed by the host language programs are defined to the service machine just as other database files are. The application program files (if any) are defined in the user's z/VM virtual machine.

Refer to the *Model 204 Host Language Interface Reference Manual* for detailed information about running IFAM2 jobs.

Single-user IFDIAL mode

A single-user IFDIAL environment looks similar to IFAM1.

You need to supply a host language program to test this mode.

Procedure

To create a single-user Model 204 IFDIAL environment:

1. A single-user IFDIAL EXEC procedure, SAMPDIAL, is supplied as part of the distributed material. Tailor SAMPDIAL EXEC, which is loaded on the 193 disk, to the specifications of your site and install it on an accessible minidisk. The M204 command expects the IFDIAL EXEC to be named SINGDIAL.
2. Use the ONLINE command to create a single-user Model 204 IFDIAL environment with the IFDIAL keyword as the connection type:

```
ONLINE IFDIAL
```

Using the IFDIAL mode

The user program name must be placed in the z/VM stack before returning to the ONLINE EXEC. Use the IFSETUP function (refer to the *Model 204 Host Language Interface Reference Manual*) to send the CCAIN parameters via the user program.

Neither CCAIN nor CCAPRINT are used for IFDIAL connections.

Using standalone Model 204 utilities

You can use Model 204 utilities in an Online environment.

You can also execute utilities in standalone service virtual machine environments. Figure 4-4 illustrates the standalone environment.

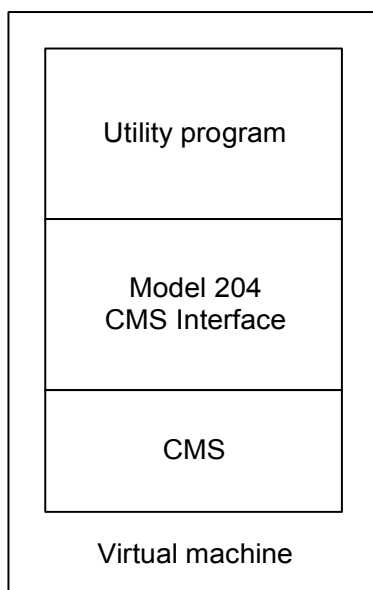


Figure 4-4. Model 204 utility architecture

Types of standalone utilities

In some cases, these utilities are Model 204 facilities that are not appropriate to the interactive environment.

In others, they are related to, but separate from, Model 204 and execute in the z/VM environment. For example, the MERGEJ and AUDIT204 utilities (described in “Model 204 Utility commands” on page 135) normally run independently of a Model 204 Online.

The following utilities are applicable only when preparing to run an Online:

- M204APND
- M204FDEF
- M204LDEF

Error tracking

When problems occur with Model 204, certain information is available to identify and correct errors. Under z/VM, Model 204 provides several problem determination aids.

Using the CCAPRINT file

The CCAPRINT file contains output for User 0 and indicates parameter or other initialization errors.

Using console messages

Some errors are noted on the service virtual machine console. Spooling this console and ensuring that the file is closed and examined on a regular basis can provide useful information about any unusual Model 204 activity.

Using audit trail information

You can obtain audit trail information in the following ways:

CCAAUDIT file	If you have selected the auditing option, the CCAAUDIT file provides problem-related information.
AUDIT204 command	If you are not using the auditing option, use the AUDIT204 command to display the audit trail information recorded in the CCAJRNL or CCAJLOG file.

Using automatic dumps

If a serious Model 204 error is detected, a dump of the Model 204 service virtual machine is created dynamically by using the z/VM Control Program VMDUMP command. The dump file is sent to the virtual machine that is defined to the z/VM Control Program as the system dump recipient.

Dump production is controlled by the presence or absence of the CCASNAP file. Unless this file is defined, dumps are not produced. Otherwise, dumps are generated depending on the setting of the Model 204 SNAPCTL option of the MSGCTL parameter. DUMMY can be specified in the file definition for CCASNAP and is adequate to cause dump production. If CCASNAP is defined as other than DUMMY, Model 204 generates formatted dump information that augments the VMDUMP.

See the *Model 204 Command Reference Manual* and the *Model 204 System Manager's Guide* for more information about formatting snaps.

Using the VMDUMP command

Some Model 204 problems require a manual dump of the service virtual machine. If requested by Technical Support, use the VMDUMP command for this purpose.

The command format is:

```
VMDUMP 0-END DSS SYSTEM FORMAT M204/CMS
```

Model 204 executes the VMDUMP command when a dump is requested by Model 204. You can omit DSS if no saved segments are associated with the virtual machine. To enter a VMDUMP command from the service virtual machine console, prefix it with the Control Program escape sequence (#CP).

After Model 204 dump files are created, read the dump files into a CMS data set using IPCS or DUMPSCAN. Then FTP the resulting CMS files and any supporting documentation (for example, an M204 audit trail), to Technical Support for analysis. For instructions on how to FTP customer files and documentation, please contact Technical Support, or visit our website for FTP upload information.

When sending VMDUMPs, remember to include either the MAPs (modules) or SEGMAPs (saved segments) of the Model 204 modules involved, usually M204ONLN and M204CMS.

5

Preparing to Install Dictionary/204

In this chapter

- Overview
- Overview of EXEC and CCAIN installation files
- Customizing EXECs
- Required privileges
- Modifying LOGIN CCAIN and LOGINSM CCAIN
- Files allocated and used by EXECs
- Customizing common FILEDEF statements

Overview

Dictionary/204 is a set of facilities provided to all installations as part of Model 204. It is used to create and maintain information about:

- Model 204 files, fields, field groups, user accounts, records, and subsystems
- Site-specified applications, reports, and facilities
- Internal system definitions

This chapter describes the steps to prepare for installing Dictionary/204 for the first time or for upgrading your site's version of Dictionary/204 to the current commercial release.

Dictionary/204 facilities

Dictionary/204 consists of these standard facilities:

- DDGEN (to populate the Dictionary/204)
- Dictionary/204
- Dictionary/204 Administration
- Dictionary/204 Migration
- Dictionary/204 Reports
- Documentation
- File Management
- Subsystem Management
- User Language Cross-Reference

Your site might also have purchased Access/204 View Management, an optional Dictionary/204 facility.

For more information about Dictionary/204

After you have installed Dictionary/204, refer to the *Model 204 Dictionary/204 and Data Administration Guide* for information on how to:

- Start, stop, and test Dictionary/204 subsystems
- Set Dictionary/204 facility parameters
- Define Dictionary/204 users
- Customize the Dictionary/204 structure
- Use the Dictionary/204 facilities

Overview of EXEC and CCAIN installation files

To install Dictionary/204, you must use several of the EXECs and CCAIN files supplied with the installation software. The D204 EXEC is the main EXEC used in Dictionary/204 installation. In addition to the D204 EXEC, several utility EXECs and CCAIN streams perform functions on M204PROC and M204TEMP for Dictionary/204 and other products, such as Workshop/204.

Using EXECs

You must use several EXECs to install Dictionary/204. The type of privileges that you need varies depending on the EXEC. Before you use the installation EXECs, modify them to meet your site's specific requirements (as discussed in

this chapter). The EXECs allocate and use several files that are provided with the installation software.

D204 EXEC

The D204 EXEC performs the actual installation and most related functions. Using installation parameters, D204 issues calls to the D204FDEF EXEC and other EXECs as necessary. The D204FDEF EXEC contains the necessary FILEDEF statements for any function of D204. The other EXECs use corresponding CCAIN files to supply account names, file names, file sizes, and other information. Run D204, specifying as parameters the function and the file you need to manipulate. For example, to back up the METADATA file, enter:

```
EXEC D204 BACKUP METADATA
```

In this case, only the FILEDEFs need to be changed in the D204FDEF EXEC. Do not change the information in D204 and the other EXECs and CCAIN files.

The D204 EXEC first executes the process equivalent to D204 BACKUP DICTIONARY prior to executing the reinstallation process. You can easily restore the databases if you encounter a problem with the reinstallation of Dictionary/204.

Printing a copy of a Model 204 EXEC or CCAIN

You can use your local printer to print a copy of any EXEC or CCAIN input stream used for Model 204 installation. To print an EXEC or CCAIN file, issue the CMS PRINT command. The format of the PRINT command is:

```
PRINT filename filetype filemode
```

For example, to print a copy of the D204FDEF EXEC, enter this command:

```
PRINT D204FDEF EXEC E
```

Required storage

D204 DDGEN, D204 INSTALL, and D204 REINSTALL require defined storage of at least 1536K. Other D204 functions and EXECs require at least 1024K.

Customizing EXECs

The EXECs supplied with the installation software are in a generalized form that you must customize for the particular installation in order to run them. The two types of required modifications are described in this section.

FILEDEF modifications

The utility EXECs, as well as virtually all of the EXECs used by D204, contain FILEDEF statements that describe files used during Dictionary/204 installation.

The parameters in these statements are site-dependent. They must be modified wherever necessary to specify values appropriate to the site standards and requirements. Parameter modification is explained below.

Some FILEDEF statements in the D204FDEF EXEC are used by all D204 functions. These statements are customized as the first step in installing or upgrading Dictionary/204. The other FILEDEFs are customized as required by each step of the installation or upgrade.

Additional modifications to EXECs and CCAIN files

Many of the EXECs and CCAIN files supplied with the installation software contain additional statements that must be modified for the particular function that you are performing. For example, when backing up a file, change the CCAIN file to specify the name of the file being backed up. CCAIN modification is explained in later chapters of this manual.

CMS-format FILEDEF statement syntax

A CMS-format FILEDEF statement has the following form:

Syntax FILEDEF { *ddname device* | *filename filetype mode* }

- Where**
- *ddname* is the Model 204 DDNAME of the file being described, for example, DUMPDATA.
 - *device* is the device on which that file resides, for example, DISK.
 - *filename* is the CMS file name of the file being described, for example, METADATA.
 - *filetype* is the CMS file type of the file being described, for example, M204DUMP.
 - *mode* is the CMS file mode of the file being described, for example, A.

Changes to CMS-format FILEDEF statements

The following statement is included in several EXECs to describe the backup file for the Dictionary/204 file named METADATA:

```
FILEDEF DUMPDATA DISK METADATA M204DUMP MODE
```

Change the MODE and possibly the ddname or file name, or both, as follows, to suit installation requirements:

```
FILEDEF DUMPDATA DISK METADATA M204DUMP A
```


z/OS-format FILEDEF statement syntax

Dictionary/204 also uses some z/OS-format files primarily for performance purposes. FILEDEF statements for z/OS-format files have a slightly different form, as follows:

Syntax FILEDEF *ddname mode DSN dsname*

- Where**
- *ddname* is the Model 204 DDNAME of the file being described, for example, DUMPDATA.
 - *mode* is the CMS file mode of the file being described, for example, A.
 - DSN is a keyword, indicating that the data set name follows.
 - *dsname* is the data set name of the file being described with periods replaced by blanks, for example, M204 CCATEMP.

Changes to z/OS-format FILEDEF statements

The following example statement is included in several EXECs to describe the z/OS-format Dictionary/204 file named METADATA:

```
FILEDEF METADATA MODE DSN D204 METADATA
```

Change the MODE and possibly the *ddname*, file name, or both, to suit installation requirements, as follows:

```
FILEDEF METADATA I DSN D204 METADATA
```

Required privileges

EXECs that create files require superuser login privileges. EXECs in which certain Model 204 commands are issued, such as INITIALIZE or DUMP, require file manager login privileges. Whenever CCASYS is opened to read or update subsystem definitions, system manager login privileges are required.

Table 5-1 shows required login privileges by each function of D204 and by each utility EXEC.

Table 5-1. Login privileges required for EXEC functions

D204 function or EXEC	Ordinary user	Super-user	File manager	System manager
D204 BACKUP			X	X
D204 DDFIMCMD				X
D204 DDGEN	X			X (for CCASYS)
D204 DDGENSET	X			
D204 DDPROCI		X	X	

Table 5-1. Login privileges required for EXEC functions (Continued)

D204 function or EXEC	Ordinary user	Super-user	File manager	System manager
D204 DDPROCP		X	X	
D204 DDTINIT		X	X	
D204 INSTALL		X	X	
D204 REINSTALL		X	X	X
D204 REORGANIZE		X	X	X (for CCASYS)
D204 RESTORE			X	
DDBKPROC			X	X (for CCASYS)
DDRSPROC			X	X (for CCASYS)
RESPROC		X	X	
TRANSFR	X			

Modifying LOGIN CCAIN and LOGINSM CCAIN

The LOGIN CCAIN and LOGINSM CCAIN files specify User 0 parameters and log the user in to Model 204:

- D204 EXEC uses the LOGIN CCAIN file to log in the user for D204 functions that require privileges for superuser, file manager, or both.
- D204 EXEC uses LOGINSM CCAIN to log in the user for D204 functions that also require system manager privileges.
- D204 EXEC functions that require nonstandard User 0 parameters do not use either of these files. They specify User 0 parameters and log in the user in a CCAIN file specific to the function. For example, the REINSTALL function logs in the user in the DDRINST CCAIN.

Before you use the D204 EXEC, modify the LOGIN CCAIN and LOGINSM CCAIN files so that they contain the correct user login accounts and passwords; these modifications are described in this section. Modifications to other CCAIN files used by the D204 EXEC are described later in this manual. You can secure Dictionary/204 files during file creation so that passwords are required for subsequent EXECs that open these files. Check the OPEN statements in each CCAIN file to determine which files are being used and to supply passwords.

Linking 193 disk as C

The 193 disk contains the CCAIN files. Be sure that the 193 disk:

- Is accessed as C

- Has the CCAIN files
- Is linked read/write

Modifying LOGIN CCAIN

Modify the LOGIN CCAIN file by replacing:

- *USERID* with a login account with superuser and file manager privileges
- *PASSWORD* with the password for that login account

Modifying LOGINSM CCAIN

Modify the LOGINSM CCAIN file by replacing:

- *USERID* with a login account with superuser and file manager privileges
- *PASSWORD* with the password for that login account

Files allocated and used by EXECs

During the installation process, the EXECs summarized in the previous section allocate and use the files listed in Table 5-2.

Table 5-2. Files allocated and used by EXECs

File	Usage
CCASYS	Data file containing a site's subsystem definitions; D204 INSTALL creates CCASYS. For an explanation of how to use CCASYS when running subsystems, refer to the <i>Model 204 System Manager's Guide</i> .
D204RPT	Procedure file containing a site's user-defined Dictionary/204 reports; D204 INSTALL creates D204RPT.
D204SYS	An intermediate file used for importing or exporting Dictionary/204 data from one Model 204 environment to another. This file needs to be created only once; it can then be used repeatedly by different Model 204 environments. D204SYS is created in D204 INSTALL and also in D204 REINSTALL.
DATALINK	Data file containing cross-references between a site's Dictionary/204 entries; D204 INSTALL creates DATALINK.
DICDIST	Procedure file for standard Dictionary/204 subsystems. DICDIST procedures are copied into the installation's M204PROC procedure file by restoring it (via RESPROC) to an intermediate Model 204 file and then transferring it from the intermediate Model 204 file to M204PROC.
M204DCTL	M204DCTL contains user profile information, help and error messages, and system EXECs. M204DCTL is created in D204 INSTALL and is recreated in D204 REINSTALL.

Table 5-2. Files allocated and used by EXECs

File	Usage
M204INST	Procedure file for installing Dictionary/204 and Workshop/204. M204INST is provided with the installation software in dump format.
M204PROC	Procedure file for Dictionary/204, Workshop/204, and other optional facilities, that is built by restoring and copying the procedure files supplied with the installation software; also contains data records indicating which fixes have been applied to the procedures.
M204TEMP	Temporary data file used for storage of Dictionary/204 and Workshop/204 data. M204TEMP contains lock records that control updating. It also contains records used for scrolling. M204TEMP is created by D204 INSTALL and is recreated in D204 REINSTALL.
METADATA	Data file containing a site's Dictionary/204 entries; D204 INSTALL creates METADATA.
OUTDDGN	File used to run DDGEN online.
OUTFILE	Default file for reports from Dictionary/204.

File usage

File usage is shown by the standard Dictionary/204 subsystems in Table 5-3. For information about the Dictionary/204 subsystems, refer to the *Model 204 Dictionary/204 and Data Administration Guide*.

Table 5-3. File usage by subsystem

File name	DOC	FIM	SUM	DIR	DIA	DIC	XRF	MIG
CCASYS			D					
D204RPT				D	D			
D204SYS			D					D
DATALINK	D	D	D	D	D	D	D	D
M204DCTL		D	D				D	D
M204PROC	P/D	P/D	P/D	P/D	P/D	P/D	P/D	P/D
M204TEMP	D	D	D	D	D	D	D	D
METADATA	D	D	D	D	D	D	D	D
Total files/subsystem	4	5	7	5	5	4	5	6

The abbreviations in Table 5-3 have the following meanings.

Abbreviation	Meaning
DOC	Documentation facility

Abbreviation	Meaning
FIM	File Management facility
SUM	Subsystem Management facility
DIR	Dictionary/204 Reports facility
DIA	Dictionary/204 Administration facility
DIC	Dictionary/204 facility
XRF	User Language Cross-Reference facility
MIG	Dictionary/204 Migration facility
P	Procedure file
D	Data file

Customizing common FILEDEF statements

Use this step to customize the FILEDEF statements that all D204 functions use.

Change the D204FDEF EXEC by finding the *-Common* label and changing:

- Definitions for CCAPRINT, CCAAUDIT, CCASNAP, CCATEMP, and CCASTAT
- *MODE* (or filemode) to reflect the correct file mode for each file
- File name and file type or DATASET NAME, as necessary

6

Installing or Upgrading Dictionary/204

In this chapter

- Overview
- Common procedures
- Allocating, creating, and restoring M204INST
- Preparing M204PROC
- Restoring distribution procedure files
- Overview for new Dictionary/204 installations
- Calculating file sizes
- Installing the standard Dictionary/204 facilities
- Backing up files
- Bringing up the Model 204 Online
- Authorizing Dictionary/204 access for all users
- Upgrading Dictionary/204
- Backing up Dictionary/204 files
- Upgrading the standard Dictionary/204 facilities
- Backing Up Dictionary/204 files
- Preparing existing files with DDGEN

Overview

This chapter explains the steps to install Dictionary/204 for the first time or to upgrade an existing version. The last section of this chapter presents the procedure for installing the Access/204 View Management facility.

Before you begin installing or upgrading Dictionary/204, complete the tasks presented in Chapter 5.

Task list

To install or upgrade Dictionary/204, complete these steps.

Common procedures

1. Allocate, create, and restore M204INST.
2. Prepare the M204PROC file: calculate size, if required; allocate; and create.
3. Restore distribution procedure files.
4. Transfer procedures from the distribution files.

With a new installation

1. Calculate file sizes.
2. Install the standard Dictionary/204 facilities.
3. Back up Dictionary/204 files.
4. Bring up the Model 204 Online.
5. Authorize Dictionary/204 access for all users.

Upgrade

1. Back up Dictionary/204 files.
2. Upgrade the standard Dictionary/204 facilities.
3. Modify the CCASYS file.
4. Back up Dictionary/204 files.

Common finishing tasks

1. Prepare existing files with DDGEN.
2. Install the Access/204 View Management facility.

Common procedures

The first four steps for installing Dictionary/204 apply whether you are installing Dictionary/204 for the first time or upgrading an existing version.

Step	Task	Page
1.	Allocate, create, and restore M204INST.	page 83
2.	Calculate the size of M204PROC; if required, allocate and create M204PROC.	page 84
3.	Restore distribution procedure files.	page 86
4.	Transfer procedures from the distribution files.	page 88

Allocating, creating, and restoring M204INST

Make sure that you access the disk containing the dump file from the MODEL204 service machine. Use the D204 EXEC with the DDPROCI option to:

- Allocate space for the M204INST procedure file
- Create M204INST
- Restore M204INST from the dump data set unloaded to the 200 F disk

Modifying D204FDEF EXEC to allocate space for M204INST

Change the D204FDEF EXEC by completing these steps:

1. Find the *-ALLCINST* label:

Calculate the number of cylinders needed for M204INST. The cylinders required vary depending on the type of device being used. M204INST requires at least 600 pages. Refer to Table 6-1.

Table 6-1. Calculating cylinders needed

CKD device	Pages per cylinder	FBA device	Blocks per page
3380	105	All	13
3390	120		
9345	90		

Change *n* to the cylinder number you calculate.

Change the data set name and mode for M204INST.

2. Find the *-M204INST* label:

Change the data set name and mode for M204INST.

Change the mode for DUMPINST. This is the dump file unloaded to MAINT204's F disk.

3. Find the *-COMMON* label:

Change the data set name and mode for CCATEMP.

Change the data set name and mode for CCASTAT.

Running D204 DDPROCI EXEC

Run the D204 EXEC with the DDPROCI option by entering this command:

```
EXEC D204 DDPROCI
```

Preparing M204PROC

Use this step to determine the appropriate size for your site's M204PROC file. Determine the size for this file before modifying the file size specification in the DDPROCP CCAIN in the next step.

Procedure file

A procedure file is available for each product and optional Dictionary/204 facility. For example, if you order Workshop/204, you receive the WORDIST procedure file. Your site's M204PROC file must be large enough to accommodate the standard Dictionary/204 facilities, the optional Dictionary/204 facilities, and Workshop/204.

Calculating M204PROC size

To calculate the size of M204PROC, determine the number of TABLE D pages required by your installation by summing the values of all the products that are being installed.

Add 13 pages for Tables A, B, C, and FCT combined.

DICDIST is the name of the Dictionary/204 distribution file and it contains 890 Table D pages.

The minimum size for M204PROC is 903 pages.

Allocating and creating M204PROC

If you have an existing M204PROC file, skip to "Increasing M204PROC size for additional products" on page 85.

If you are creating a new M204PROC, use the D204 EXEC with the DDPROCP option to:

- Allocate space for the M204PROC procedure file

- Create M204PROC

Procedure

To execute the D204 EXEC, modify the D204FDEF EXEC with these steps:

1. Modify D204FDEF EXEC by changing the information at these labels:

-ALLCPROC	Allocation for M204PROC. Change MODE to reflect the file mode of the file. If necessary, change the DATASET NAME.
-----------	-------------------------------------------------------------------------------------------------------------------

-M204PROC	File definition for M204PROC. Change MODE to reflect the file mode of the file. If necessary, change the DATASET NAME.
-----------	------------------------------------------------------------------------------------------------------------------------

2. Modify DDPROCP CCAIN file by changing DSIZE to 890 or to the total number of TABLE D pages calculated in “Calculating M204PROC size” on page 84.
3. Run the D204 EXEC with the DDPROCP option by entering:

```
EXEC D204 DDPROCP
```

Increasing M204PROC size for additional products

If you are adding more facilities, increase M204PROC in one of the following ways:

- If there are enough FREESIZE pages, use the INCREASE TABLED command to move the pages to Table D.
- Allocate a separate data set with the additional pages and use the INCREASE DATASETS command to make the multiple data sets available as one Model 204 file.
- Back up M204PROC using the DDBKPROC EXEC. Allocate and CREATE a larger file, restore into it, and use the INCREASE TABLED command to add FREESIZE pages to Table D.

Back up M204PROC

Modify D204FDEF EXEC by changing the following:

M204PROC File definition for M204PROC

- Change MODE to reflect the file mode of the file.
- If necessary, change the DATASET NAME.

If you have an existing M204PROC file, you should back it up at this point. For instructions, see “Backing up M204PROC” on page 105.

Restoring distribution procedure files

The software programs for Dictionary/204, Workshop/204, and other optional facilities are stored on the M204PROC file. New versions of the Dictionary/204 and Workshop/204 software are delivered as dump files with the installation software.

To install the newest version, execute RESPROC and TRANSFR for each facility you are installing (or reinstalling):

- RESPROC creates a Model 204 file, called DICDIST or WORDIST, from a dumped version of the file loaded from the installation software.
- TRANSFR copies the procedures on the DICDIST or WORDIST file to your M204PROC file. At the conclusion of this step, your software is at the newest release level.

If you are installing Dictionary/204 or Workshop/204, your data files, such as METADATA, DATALINK, and so on, are created during New Installation procedures. If you are upgrading Dictionary/204 or Workshop/204 from a prior release, your existing data files are modified for use with the new software during Upgrade procedures.

Execute RESPROC and TRANSFR for DICDIST to install the Dictionary/204 facilities. Execute RESPROC and TRANSFR again for WORDIST to install the Workshop/204 facilities.

Using the RESPROC EXEC

The RESPROC EXEC restores the distribution procedure file for each product ordered into a temporary Model 204 file. The RESPROC and TRANSFR utilities combine all the procedures distributed for the Dictionary/204 and Workshop/204 products, and optional Dictionary/204 facilities, into a single procedure named M204PROC.

Procedure

Follow these steps to modify and run RESPROC and TRANSFR for each distribution file to be restored:

1. Change the RESPROC EXEC by modifying the FILEDEF statements to reflect your site's data sets:

Replace...	With...
DUMPPROC	Name of the dumped file from the installation software for the procedure file to be restored to, for example, DICDIST M204DUMP.

Replace...	With...
FILENAME	Name of the procedure file for the product or optional facility to be restored, for example, DICDIST for the Dictionary/204 product. Then insert the data set name for DICDIST in the ALLOC statement and in the FILEDEF statement. Change <i>N</i> to the number of cylinders you calculate for the file size. Use Table 6-1 to calculate the number of cylinders required. <ul style="list-style-type: none"> DICDIST requires 910 Model 204 pages WORDIST requires 800 Model 204 pages. Repeat this file name in the RESPROC CCAIN.
	Do not run RESPROC for M204INST; the M204INST file is restored by using D204 DDPROCI and its procedures are not copied to M204PROC.
MODE	File mode for each file. Change it in all FILEDEF statements. If necessary, change the file names and file types.

2. Change the RESPROC CCAIN:

Replace...	With...
USERID	Login account with superuser and file manager privileges.
PASSWORD	Password for the login account.
FILENAME	Name of the procedure file to be restored (the same as the file name specified in the RESPROC EXEC, for example, DICDIST or WORDIST). If necessary, add a password to OPEN filename statements.

3. Modify the D204FDEF EXEC as follows:

Change the FILEDEF statement for M204PROC to indicate the correct data set name and mode.

4. To run RESPROC, enter:

```
EXEC D204 RESPROC
```

Transferring procedures from the distribution files using TRANSFR

Procedure

Follow these steps using the TRANSFR EXEC to copy procedures from each temporary Model 204 distribution file into M204PROC:

- Change the TRANSFR EXEC by changing the FILEDEF for FILENAME to reflect the distribution file you are using, for example, DICDIST, WORDIST, and so on. This file was restored in the RESPROC step. Change the TRANSFR CCAIN mode to indicate its location.
- Change the TRANSFR CCAIN by modifying the following FILEDEF state-

ments to reflect your site's data sets:

Replace...	With...
USERID	Login account with ordinary user privileges.
PASSWORD	Password of the USERID account.
FILENAME	Name of the Model 204 file that contains the procedures to be copied into M204PROC; the same as the file name in the TRANSFR EXEC.

- If necessary, add a password to OPEN M204PROC statement.
- To run TRANSFR, enter:

```
EXEC D204 TRANSFR
```

You can use this job to reorganize any procedure file. In this more general usage, change the FILENAME parameter in the CCAIN to the name of the file from which you are transferring procedures and change M204PROC in the CCAIN to the name of the file to which procedures are transferred. Replace the FILENAME and M204PROC FILEDEF statements.

If you have an existing Dictionary, skip to section “Upgrading Dictionary/204” on page 98.

Overview for new Dictionary/204 installations

This section lists the steps necessary to install Dictionary/204 for the first time. Additional information, as appropriate, follows the list.

Before performing the procedures listed here, complete the steps listed in “Common procedures” on page 83.

Step	Task	Page
1.	Calculate file sizes for: <ul style="list-style-type: none"> • CCASYS • D204RPT • D204SYS • DATALINK • M204DCTL • M204TEMP • METADATA 	page 89
2.	Define standard Dictionary/204 facilities.	page 94
3.	Back up Dictionary/204 files (do not delete M204INST).	page 96
4.	Bring up the Online.	page 96
5.	Authorize Dictionary/204 access for all users.	page 98

Calculating file sizes

This section tells how to calculate appropriate sizes for these installation files:

- CCASYS
- METADATA
- DATALINK
- D204RPT
- M204TEMP
- M204DCTL
- D204SYS

Determine the size before modifying the file size specifications for the files created by D204 INSTALL. When computing file sizes, take into consideration the size requirements of all the products you are installing, including products other than Dictionary/204. For more information about the Model 204 parameters, for example, ATRPG and BSIZE, refer to the *Model 204 Command Reference Manual* and the *Model 204 File Manager's Guide*.

The calculation for the total number of pages required is:

ASIZE + BSIZE + CSIZE + DSIZE + ESIZE + 8 (for the FCT)

CCASYS file

The CCASYS file, which contains your site's subsystem definitions and internal procedures, has a minimum size of 25 pages. Use these steps to determine the right size for your installation:

1. Estimate:
 - Number of subsystems. Count all the subsystems you are installing, including Dictionary/204 subsystems and Workshop subsystems, and other CCA and user-defined subsystems. Call this number *NS*.
Dictionary/204 has nine standard subsystems and three optional subsystems. To obtain the number of subsystems for other products being installed, see the product's installation guide for your operating system.
 - Average number of files per subsystem. Call this number *NFS*. The number used for CCA subsystems is six.
 - Average number of users who are not in a default subsystem class. Call this number *NUS*.
 - Average number of sclasses per subsystem. Dictionary/204 subsystems use two. Call this number *NSC*.
2. Use the numbers from Step 1 in the following formula:

```
ATRPG = 1, FVFPG = 1, MVFPG = 1
BRESERVE = 88
BRECPPG = PGSIZE/BRESERVE
BSIZE = 2 * (NS*.03) + (NS * NFS *.03)
        + (NS * NUS *.022)
CSIZE = ((14*NS*NSC*NUS)
        + (7*((NS*(2 + NSC + NUS)/49152) +1) *
        (2 * NS(2 + NS(2 + NSC + NUS))))/6144) * 1.2)
DSIZE = 10 + NS
ESIZE = 0
```

The following sizes are suggested for calculating CCASYS. Depending on your data and applications, you may need to change these sizes at a later date.

```
ASIZE = 3 BRECPPG = 70
BSIZE = 40
CSIZE = 10
DSIZE = 40
ESIZE = 0
FCT = 8
```

Total = 101

METADATA file

The METADATA file contains your installation's Dictionary/204 entries. Use this formula to calculate its size:

```
ATRPG = 2, FVFPG = 1, MVFPG = 1
BRESERVE = average record length (225 suggested)
BRECPPG = page size/BRESERVE
BSIZE = number of expected entries/BRECPPG
CSIZE = BSIZE *.1
DSIZE = BSIZE *.2
ESIZE = 0
```

The following sizes are suggested for calculating METADATA. Depending on your data and applications, you may need to change these sizes at a later date.

```
ASIZE = 3 BRECPPG = 31
BSIZE = 300 BRESERVE = 225
CSIZE = 30
DSIZE = 60
ESIZE = 0
FCT = 8
```

Total = 401

For information about Dictionary/204 entries, refer to the *Model 204 Dictionary/204 and Data Administration Guide*.

DATALINK file

The DATALINK file contains relationships between your installation's Dictionary/204 entries. Use this formula to calculate its size:

```

ATRPG = 1, FVFPG = 1, MVFPG = 1
BRESERVE = 120 (suggested)
BRECPPG = PGSIZE/BRESERVE
number DATALINK records =
    (5 * number METADATA records expected)
BSIZE = number of DATALINK records/BRECPPG
CSIZE = BSIZE * .2
DSIZE = BSIZE * .1
ESIZE = 0

```

BRESERVE is based on an average name length of 30 characters. If the average name length for your installation is different, increase or decrease BRESERVE by twice the difference between 30 and the average name length.

The following sizes are suggested for calculating DATALINK. Depending on your data and applications, you may need to change these sizes at a later date:

```

ASIZE = 3    BRECPPG = 31
BSIZE = 300  BRESERVE = 200
CSIZE = 30
DSIZE = 60
ESIZE = 0
FCT = 8

Total = 401

```

D204RPT file

The D204RPT file contains your installation's procedures for all user-defined Dictionary/204 reports. Use this formula to calculate its size:

```

ATRPG = 1, FVFPG = 1, MVFPG = 1
BSIZE = 5
CSIZE = 1
DSIZE = number of procedures * (average pages per procedure) + 6
ESIZE = 0

```

The following sizes are suggested for D204RPT. Depending on your data and applications, you may need to change these sizes at a later date:

```
from above 6
ASIZE = 3
BSIZE = 5
CSIZE = 1
DSIZE = 90
FCT = 8

Total = 113
```

M204TEMP file

The M204TEMP file is a data file that stores temporary Workshop/204 and Dictionary/204 data. Use the following values to calculate its approximate size:

```
BRESERVE = 140
BRECPPG = 75
BSIZE = 100
CSIZE = 10
DSIZE = 30
ESIZE = 0
```

Multiply the values for BSIZE, CSIZE, and DSIZE by 1.1 for every 10 users. For a more exact calculation, refer to Appendix B.

The following sizes are suggested for M204TEMP. Depending on your data and applications, you may need to change these sizes at a later date:

```
from above 140
ASIZE = 3
FCT = 8

Total = 151
```

M204DCTL file

The M204DCTL file is the control file for the User Language Cross-Reference facility.

This file also stores error messages and help information for other Dictionary/204 facilities. As a result of this additional information, Table B requires 25 pages and Table D requires 12 pages. These values may be set during the job that recreates M204DCTL to redefine fields and reset BRECPPG and BRESERVE parameters. The data set space may need to be increased if there is not enough free space available in the file for the increased table sizes.

Use the following values for an approximate calculation of the size of this file:

```
BRESERVE = 130
BRECPPG = 52
FCT = 8 pages
Table A = 3 pages
```

Table B = 25 pages for installation records (leaves a cushion for growth) + (the number of XREF users with profile records / 5)

Table C = 5 pages (ample for most situations)

Table D = 12 pages (for installation data) + number of XREF users.

Table E = 0 pages

Set the values for BREERVE and BRECPPG in the DICN.DCTL.PARMS procedure. You may need to reset the values if a large number of users have profile records. The following sizes are suggested for M204DCTL. Depending on your data and applications, you may need to change these sizes at a later date:

```
ASIZE = 3
BSIZE = 30
CSIZE = 5
DSIZE = 30
ESIZE = 0
FCT = 8
```

```
Total = 76
```

D204SYS file

The D204SYS file is an intermediate file used for importing or exporting data from one Model 204 environment to another. Therefore, this file only needs to be created once; subsequent Model 204 environments can then use the same file for importing and exporting. The size of this file depends on the expected number of files and subsystems to be exported.

To estimate the size of D204SYS, perform these steps:

1. Estimate export activity:

- The maximum number of files that will be exported to D204SYS at one time. Call this number *NF*.
- The maximum number of subsystems that will be exported to D204SYS at one time. Call this number *NS*.
- The average number of Dictionary/204 records per file, including both METADATA and DATALINK. Call this number *AF*.
- The size of the average file. Multiply *AF* by the average record length for METADATA and DATALINK records. Call this number *SF*.
- The size of the average subsystem. Call this number *SS*. Use the following formula as a guide:

$$SS = 1.2 * (\text{Number of files in subsystem} * 55 * \text{number of sclasses}) + (\text{Number of users} * 34)$$

2. Compute sizing parameters. Using the values you have estimated in Step 1, set parameters as follows:

```
BRESERVE = average record length
BRECPPG = (PGSIZE - 44)/BRESERVE
BSIZE = (2.3 * (NF*SF/(PGSIZE - 44)) + (NS*SS/(PGSIZE -
44)))
CSIZE = 10
DSIZE = 15
ESIZE = 0
```

The following sizes are suggested for calculating D204SYS. Depending on your data and applications, you may need to change these sizes at a later date:

```
ASIZE = 3 BRECPPG = 35
BSIZE = 300 BRESERVE = 125
CSIZE = 30
DSIZE = 60
ESIZE = 0
FCT = 8
```

```
Total = 401
```

Installing the standard Dictionary/204 facilities

D204 INSTALL command

Use the D204 EXEC with the INSTALL option to:

- Allocate space for and create the CCASYS, D204RPT, D204SYS, DATALINK, M204DCTL, M204TEMP, and METADATA files.
- Define initial entries in the Dictionary/204 and the CCASYS file for:
 - Dictionary/204 administrator account
The Dictionary/204 administrator is authorized to use every Dictionary/204 facility, including the Subsystem Management facility, provided that this administrator also has system manager privileges. The Dictionary/204 administrator can then use the Dictionary/204 Administration facility to authorize access to facilities for other users.
 - Standard Dictionary/204 entity types
 - Entries for Dictionary/204 subsystems
The login account of the system manager running D204 INSTALL is added to the ADMIN class of each Dictionary/204 subsystem. The system manager can start, stop, and test each subsystem. The system manager is also authorized to use the Subsystem Management facility and can start, stop, and test each subsystem.
 - Internal CCASYS procedures
 - Initial M204TEMP field definitions

Procedure

1. Change the FILEDEF statements of the D204FDEF EXEC to reflect your site's file names, file modes, file types, and space allocations, as follows:

Label	Change
-ALLCDICT	Gives space allocations for the files CCASYS, D204RPT, D204SYS, DATALINK, M204DCTL, and METADATA.
-ALLCTEMP	Gives space allocation for the file M204TEMP.
-CSYS	Customizes FILEDEF for CCASYS.
-D204SYS	Customizes FILEDEF for D204SYS.
-DATA	Customizes FILEDEF for METADATA.
-DDTEMP	Customizes FILEDEF for M204TEMP.
-LINK	Customizes FILEDEF for DATALINK.
-M204DCTL	Customizes FILEDEF for M204DCTL.
-REPORT	Customizes FILEDEF for D204RPT.

2. Change the DDINST CCAIN as follows:

Replace...	With...
USERID	Login account with system manager, superuser, and file manager privileges.
PASSWORD	Password for the USERID account.
DICTIONARY.DBA	Initial login account of your installation's Dictionary/204 administrator.
OUTFILE	Name of the file to be used as the default output file (OUTFILE) for Dictionary/204. The default is used by the Dictionary/204 administrator and the PUBLIC account. If you do not know the name, use OUTFILE.
COMMAND.PROCFILE	Name of the file that is to be used as the default procedure file for commands generated through FILEMGMT.
XX	Appropriate file table sizes for installation requirements. Change the PARAMETER settings in the CREATE statements for the CCASYS, METADATA, M204TEMP, M204DCTL, DATALINK, D204SYS and D204RPT files. Refer to the previous calculations. Set the OPENCTL and PRIVDEF parameters to secure Dictionary/204 files. For more information about file security, refer to the <i>Model 204 Command Reference Manual</i> and the <i>Model 204 File Manager's Guide</i> .

3. Run the D204 EXEC with the INSTALL option for Dictionary/204 by entering:

EXEC D204 INSTALL DICTIONARY

Backing up files

After installing the standard Dictionary/204 facilities, back up the files by using D204 BACKUP and DDBKPROC EXEC (refer to Chapter 7).

Do *not* delete M204INST. It is used by the RESTORE and REORG EXECs.

Bringing up the Model 204 Online

ONLINE EXEC

The ONLINE EXEC performs these functions:

- Executes a user-created EXEC procedure to define the file recovery environment
- Invokes Model 204 to perform file recovery
- Executes a user-created EXEC procedure to define the Online environment
- Invokes Model 204 to establish the Online environment

Modifying the user-created EXEC

Modify the user-created EXEC that you are using to define the Online environment to include the Dictionary/204 files. These modifications are for standard Dictionary/204 facilities only. Other necessary changes for Model 204 are described in the *Model 204 System Manager's Guide*.

Setting up the Online

To set up the Online, follow these steps:

1. Include FILEDEF statements for these Dictionary/204 files:

```
CCASYS
METADATA
DATALINK
D204RPT
M204TEMP
M204PROC
M204DCTL
OUTFILE
OUTDDGN
D204SYS
```

2. OUTFILE is the default file for reports from Dictionary/204. Add FILEDEF statements for other defined Dictionary/204 report files. OUTFILE must be a sequential file. If it is a disk file, set the DISP parameters to (MOD,

PASS) so that output from multiple requests is concatenated, rather than overwritten.

3. OUTDDGN is required to run DDGEN Online. It must be a sequential file. Set DISP parameters to OLD, because it is a work file, and RECFM FBA.
4. Allocate 800 pages in CCATEMP for Dictionary/204 procedures.
5. Change the CCAIN input stream as follows:

Parameter	Setting
LENQTBL	Include 250 entries for Dictionary/204.
LGTBL	Set to at least 500.
LOUTPB	Set to at least 3000.
LPDLST	Set to at least 1760; the default value is 2600.
NFILES, NDCBS, NDIR	Specify 7 for Dictionary/204 files. This number includes D204RPT, D204SYS, DATALINK, M204DCTL, M204PROC, M204TEMP and METADATA. These parameters are automatically incremented 1 for CCASYS, if SYSOPT is odd.
NORQS	Set to at least 10.
SERVSZ	Set to at least 206000 for each server to support Dictionary/204 usage. If User 0 does not use any Dictionary/204 subsystems, only LGTBL must be set to at least 500 for User 0. The recommended SERVSZ assumes the default values for server tables FTBL, ITBL, TTBL and XTBL. If your values for these tables are larger than the default, adjust your SERVSZ accordingly.
SPCORE	Increase by 3507+ (6*NFILES) for Dictionary/204 system control blocks.
SYSOPT	The 4 option restricts the use of Model 204 data definition commands within a particular run. If SYSOPT includes this option, Model 204 requires that file creation, definition, and subsequent maintenance of the file definitions in the run be accomplished through the File Management facility of Dictionary/204. For more information about the 4 option, such as the commands it affects, refer to the <i>Model 204 Command Reference Manual</i> . The 1 option enables the CCASYS file and must be set in order to open and use CCASYS.

Recovery

Dictionary/204 files must participate in your installation's recovery scheme, just as other Model 204 files do; refer to Chapter 7. If any one of the Dictionary/204 or Subsystem Management files is defined as a transaction backout file (the default), all the files must be defined as transaction backout files.

Bringing up the system

At this point, you can bring up the system.

Authorizing Dictionary/204 access for all users

Before Dictionary/204 users can log in to any Dictionary/204 facility, the Dictionary/204 administrator (the account name used when Dictionary/204 was installed) must add the user account names to the Dictionary/204 security function.

Authorize access for users through the Dictionary/204 Administration facility by selecting the Security Administration function. Refer to the *Model 204 Dictionary/204 and Data Administration Guide* for an overview and for details on using the Dictionary/204 Administration facility.

Applying early warnings

Apply all appropriate Dictionary/204 early warnings using the Autofix system.

Refer to Chapter 4 for more information about using the Autofix system.

Upgrading Dictionary/204

This section lists the steps necessary to upgrade Dictionary/204. The steps are summarized in the following checklist. Additional information, as appropriate, follows the checklist.

Complete the steps listed in “Common procedures” on page 83 before you perform the procedures listed below.

Step	Task	Page
1.	Back up Dictionary/204 files.	page 98
2.	Upgrade the standard Dictionary/204 facilities.	page 100
3.	Use DDGEN to prepare existing files for file management.	page 101

Backing up Dictionary/204 files

Using D204 BACKUP

Use D204 BACKUP, customized in a previous release, to back up these Dictionary/204 files, which occur in the following order:

- CCASYS
- D204RPT
- D204SYS

- DATALINK
- M204DCTL
- METADATA

If you have already customized a backup exec, use it. Otherwise, do the following:

For each dump file:

1. Determine if you want to use z/VM or z/OS formatted disks for backup:

- z/VM formatted dump file:

```
FILEDEF DUMPDCTL DISK d204 dumpdctl mode
```

- z/OS formatted dump file:

```
FILEDEF DUMPDCTL mode DSN d204 m204dctl dump
```

If you are using z/OS formatted dump files, allocate them with the following commands. Use the same space allocation as their corresponding M204 database files:

```
M204UTIL ALLOC d204 CCASYS DUMP mode (PRIMARY n CYL)
M204UTIL ALLOC d204 D204RPT DUMP mode (PRIMARY n CYL)
M204UTIL ALLOC d204 D204SYS DUMP mode (PRIMARY n CYL)
M204UTIL ALLOC d204 DATALINK DUMP mode (PRIMARY n CYL)
M204UTIL ALLOC d204 M204DCTL DUMP mode (PRIMARY n CYL)
M204UTIL ALLOC d204 METADATA DUMP mode (PRIMARY n CYL)
```

2. Change the D204FDEF EXEC at the following labels:

```
-M204DCTL
-D204SYS
-DATA
-LINK
-REPORT
-CSYS
```

For each Model 204 file, change the data set name and mode.

For each dump file, if you are using z/VM formatted files for backup, change the file name, file type, and file mode for the dump data sets. If you are using z/OS formatted files for backup, change the data set name and mode.

3. Check BKPDICT EXEC — check CCAIN file modes.
4. Check BKPDICT EXEC CCAIN files for file passwords.
5. Check BKPDSYS EXEC — check CCAIN file modes.
6. Check BKPDSYS CCAIN for file passwords

To back up the Dictionary/204 files, enter:

```
EXEC D204 BACKUP DICTIONARY
```

EXEC D204 BACKUP D204SYS

Using D204 RESTORE

Use D204 RESTORE to restore the files from these backups and begin the upgrade process again, if there are any problems while upgrading. The restore makes use of the changes you made for D204 BACKUP.

To restore the files

1. Edit RSTDICT EXEC
 - Check CCAIN file modes
 - For each CCAIN file listed, check file passwords
2. Edit RSTDSYS EXEC
 - Check CCAIN file modes
 - Check file passwords in RSTDSYS CCAIN
3. Then enter:

```
EXEC D204 RESTORE DICTIONARY
EXEC D204 RESTORE D204SYS
```

Upgrading the standard Dictionary/204 facilities

Use the D204 EXEC with the REINSTALL option to:

- Add and modify control entries for Dictionary/204 architecture
- Recreate the existing M204TEMP file

Procedure

1. Change D204FDEF EXEC to reflect your site's file data set names and modes for each Model 204 database file. Changes appear in the following order:

At label ...	Change MODE DSN to local values in statement...
-CSYS	FILEDEF CCASYS
-D204SYS	FILEDEF D204SYS
-DATA	FILEDEF METADATA
-DDTEMP	FILEDEF M204TEMP
-LINK	FILEDEF DATALINK
-M204DCTL	FILEDEF M204DCTL
-M204PROC	FILEDEF M204PROC
-REPORT	FILEDEF D204RPT

2. Change the DDRINST CCAIN as follows:

Replace...	With...
USERID	Login account with system manager and superuser privileges.
PASSWORD	Password for the USERID account.
DICTIONARY.DBA	Initial login account of your installation's Dictionary/204 administrator.
OUTFILE	Name of the file to be used as the default output file OUTFILE for Dictionary/204. The default is used by the Dictionary/204 administrator and the PUBLIC account. If you do not know the name, use OUTFILE. Set the OPENCTL and PRIVDEF parameters to secure M204TEMP. For more information about file security, refer to the <i>Model 204 File Manager's Guide</i> .

3. If necessary, add passwords for Dictionary/204 files to the OPEN statement.

For more information about Dictionary/204 administration and facilities, refer to the *Model 204 Dictionary/204 and Data Administration Guide*.

4. Run the D204 EXEC with the REINSTALL option for Dictionary/204 by entering:

```
EXEC D204 REINSTALL DICTIONARY
```

Backing Up Dictionary/204 files

After upgrading the standard Dictionary/204 facilities, use D204 BACKUP and DDBKPROC EXEC to back up the files. For complete instructions, refer to Chapter 7.

Preparing existing files with DDGEN

Bringing up the Online

At this point, you can bring up the system.

Preparing existing files for file management

You must run existing files through D204 DDGEN before you can update them through Dictionary/204 File Management. All system-controlled attributes of files must be updated through Dictionary/204 File Management.

If any of the existing files are already in the Dictionary/204, use D204 DDGENSET to set up a D204 DDGEN run for these files. For complete instructions, see Chapter 7.

7

Dictionary/204 Operation and File Maintenance

In this chapter

- Overview
- Backing up Dictionary/204 files
- Backing up M204PROC
- Restoring Dictionary/204 files
- Restoring M204PROC
- Reorganizing Dictionary/204 files
- Populating Dictionary/204 from a Model 204 file
- Modifying the DDGEN CCAIN file
- Setting up D204 DDGEN based on Dictionary/204 files
- Updating files using delayed processing
- Reinitializing M204TEMP

Overview

In addition to the EXECs that must be run during installation, the installation software contains the EXECs and CCAIN files for a number of other functions that are used in the daily operation of Dictionary/204 and Dictionary/204 file maintenance. The following sections describe the functions and provide information necessary for the changes that must be made to the EXECs and CCAIN files.

When to modify EXECs

Although you do not need to run the functions described in this chapter during installation, it is recommended that you modify the EXECs and CCAIN files when installing Dictionary/204, so that they contain the appropriate values for your site. When running the functions of D204 and the separate utility EXECs, be sure that you have defined storage of at least 1024K.

Backing up Dictionary/204 files

Files to be backed up

Use D204 BACKUP to back up the Dictionary/204 files including:

- CCASYS
- D204RPT
- DATALINK
- M204DCTL
- METADATA

You do not need to back up M204TEMP, because D204 DDTINIT can reinitialize the file at any time. M204PROC is backed up separately, because it must be backed up after installing a new product, after upgrading to a new release, or after applying maintenance.

The D204SYS file is not included in the backup options, because it is an intermediate file that holds data temporarily. If you wish to back up D204SYS, modify D204 and D204FDEF accordingly.

See the *Model 204 File Manager's Guide* for information on performing multitape backups and restores.

Using D204 BACKUP

Include an argument after BACKUP that specifies the files to be backed up as follows:

Argument	Backs up...
ALL	Dictionary/204 and Workshop/204 files
CCASYS	CCASYS only
D204RPT	D204RPT only
DATALINK	DATALINK only
DICTIONARY	All of the above
M204DCTL	M204DCTL only

Argument	Backs up...
METADATA	METADATA only

For example, to back up the METADATA file, enter:

```
D204 BACKUP METADATA
```

Modifying D204FDEF EXEC and CCAIN files

To modify D204FDEF EXEC and CCAIN files:

1. Change the D204FDEF EXEC:

Label	Change to customize
-CSYS	DUMPCSYS FILEDEF for CCASYS backup.
-DATA	DUMPDATA FILEDEF for METADATA backup.
-LINK	DUMPLINK FILEDEF for DATALINK backup.
-M204DCTL	DUMPDCTL FILEDEF for M204DCTL backup.
-REPORT	DUMPRPT FILEDEF for D204RPT backup.

For information about customizing the Workshop/204 FILEDEFs before using the ALL option, refer to the *Workshop/204 Installation and Operations Guide*.

2. If a file to be backed up is secured, modify its corresponding CCAIN to specify a password after the OPEN statement.

To back up this file...	Modify this file...
CCASYS	BKPCSYS CCAIN
D204RPT	BKPRPT CCAIN
DATALINK	BKPLINK CCAIN
M204DCTL	BKPDCTL CCAIN
METADATA	BKPDATA CCAIN

For information about customizing the Workshop/204 CCAIN files before using the ALL option, refer to the *Workshop/204 Installation and Operations Guide*.

Backing up M204PROC

Use the DDBKPROC EXEC to back up the M204PROC file, which holds the Dictionary/204 and Workshop/204 procedures.

You must back up M204PROC each time you install a product that transfers procedures into M204PROC or when an early warning is applied to a procedure

in M204PROC. The Dictionary/204 and Workshop/204 products do not update M204PROC.

Procedure

1. Change the DDBKPROC EXEC:
 - Replace MODE with the file mode of each file.
 - Change all FILEDEF statements. If necessary, change the file names and file types.

2. Change the DDBKPROC CCAIN:

Replace...	With...
USERID	Login account with file manager privileges
PASSWORD	Password for the login account

3. If M204PROC is secured, modify the DDBKPROC CCAIN to specify a password after the OPEN statement.
4. To back up M204PROC, enter:

```
ONLINE NODCSS BYPASS DDBKPROC
```

Restoring Dictionary/204 files

Using D204 RESTORE

Use D204 RESTORE to restore Dictionary/204 files from the dumped version created with D204 BACKUP.

The argument after RESTORE specifies the files to be restored. The values and meanings are the same as for D204 BACKUP.

For example, to restore the METADATA file, enter:

```
D204 RESTORE METADATA
```

Modifying D204FDEF EXEC and CCAIN files (if necessary)

The D204FDEF modifications already made to run D204 BACKUP are all that are required for D204 RESTORE.

If a file to be restored is secured, modify its corresponding CCAIN file to specify a password after the OPEN statement.

To restore this file...	Modify this file...
METADATA	RSTDATA CCAIN
DATALINK	RSTLINK CCAIN

To restore this file...	Modify this file...
D204RPT	RSTRPT CCAIN
CCASYS	RSTCSYS CCAIN
RSTDCTL	RSTDCTL CCAIN

Restoring M204PROC

Use the DDRSPROC EXEC to restore the M204PROC file from a DDBKPROC dump.

Procedure

1. Change the DDRSPROC EXEC:
 - Replace MODE with file mode of each file.
 - Change all FILEDEF statements. If necessary, change file names and file types.
2. Change the DDRSPROC CCAIN:

Replace...	With...
USERID	Login account with file manager privileges
PASSWORD	Password for the login account

3. To restore M204PROC, enter:

```
EXEC ONLINE NODCSS BYPASS DDRSPROC
```

Reorganizing Dictionary/204 files

The D204 REORGANIZE command unloads, reallocates, recreates, and uses the Model 204 FASTLOAD program to reload the specified file.

For information about FASTLOAD and file reorganization, refer to the *Model 204 File Manager's Guide*.

Files to be reorganized

Run D204 REORGANIZE when the file to be reorganized is not in use. Make a current backup before attempting to reorganize the file. Refer to the D204 BACKUP command description on page 104. Use D204 REORGANIZE to reorganize any of these Dictionary/204 files:

- METADATA
- DATALINK
- D204RPT
- CCASYS

- M204DCTL

Using D204 REORGANIZE

The argument following REORGANIZE specifies the file to be reorganized. For example, to reorganize the METADATA file, enter:

```
D204 REORGANIZE METADATA
```

The EXEC works on one file at a time. Neither the Dictionary/204 nor the ALL option is available when reorganizing.

Reorganizing D204RP

To reorganize D204RPT, follow these steps:

1. Run the TRANSFR EXEC, described earlier, to temporarily transfer procedures from D204RPT (replace FILENAME with D204RPT) to another Model 204 procedure file.
For example, transfer TEMPRPT (replace M204PROC with TEMPRPT).
2. Run D204 REORGANIZE for D204RPT to reorganize the data and change file parameters.
3. Run the TRANSFR EXEC again to transfer procedures from the temporary file (replace FILENAME with TEMPRPT) to the D204RPT file (replace M204PROC with D204RPT).

Including site-defined Dictionary/204 extensions

If you have extended Dictionary/204 to include additional entity types, or additional entity type attributes, or additional fields, you must be sure to include these when you reinstall Dictionary/204. Use one of the following techniques:

- Use the DISPLAY command to display all the field definitions into an OUTFILE. Use this file as input for defining the fields again.
For more information, refer to the *Model 204 Command Reference Manual*.
- Run D204 DDGEN for the METADATA file.

In Dictionary/204 File Management, use the RECREATE command and execute the commands with the inactive option, which puts the DEFINE commands into a procedure or a sequential OUTFILE. Use this procedure or file as the input to the reorganization.

EXECs and CCAIN for files that might require reorganization

The following chart lists the EXECs and CCAINs for each file that you might want to reorganize:

File name	EXEC or CCAIN to unload data	EXEC or CCAIN to load data
METADATA	PAIDATA	LOADDATA
DATALINK	PAILINK	LOADLINK
D204RPT	PAIRPT	LOADRPT
CCASYS	PAICSYS	LOADCSYS
M204DCTL	PAIDCTL	LOADDCTL

Modifying the D204FDEF EXEC and CCAIN file

To modify the D204FDEF EXEC and CCAIN file:

1. Change the D204FDEF EXEC:

Label	Change...		
	MODE in all statements	File name and file type (if necessary)	Space allocation for OUTPAI (if necessary)
-ROGDATA	X	X	X
-TAPEDATA	X	X	
-ROGLINK	X	X	X
-TAPELINK	X	X	
-ROGRPT	X	X	X
-TAPERPT	X	X	
-ROGCSYS	X	X	X
-TAPECSYS	X	X	
-ROGDCTL	X	X	X
-TAPEDCTL	X	X	

2. Change the CCAIN files for loading data by replacing *XX* with the appropriate file table sizes for your installation. Change the PARAMETER settings in the CREATE statement for the file being reorganized.

For example, the LOADLINK CCAIN file specifies information necessary to create the DATALINK file. In this CCAIN file, *replace* the following statements:

```
CREATE DATALINK
PARAMETER ATRPG=XX, FVFPG=1, MVFPG=1
PARAMETER BSIZE=XX, BRECPPG=XX, BRESERVE=XX
PARAMETER CSIZE=XX, DSIZE=XX
```

with these statements:

```
CREATE DATALINK
PARAMETER ATRPG= 1, FVFPG=1, MVFPG=1
PARAMETER BSIZE=50, BRECPPG=119, BRESERVE=52
PARAMETER CSIZE=50, DSIZE=25
```

Note: The above values are *recommended* values only. CSIZE and DSIZE must be calculated by sizing DATALINK. If you have defined many new long record types, you might need a larger BRESERVE.

If the files are secured, specify the password after the OPEN statement.

Populating Dictionary/204 from a Model 204 file

Using D204 DDGEN

D204 DDGEN populates your installation's Dictionary/204 from existing Model 204 files.

Run D204 DDGEN when creating your Dictionary/204 for the first time or to update the Dictionary/204 to include definitions of new files. You can use D204 DDGENSET as a front end to D204 DDGEN (see page 111).

You can run DDGEN in Online mode, if you prefer.

You can specify up to 100 Model 204 files.

Modifying D204FDEF EXEC

Note: File name and password specifications are read as input by DDGEN. If you do not need passwords for the specified files, supply blank lines.

Change the D204FDEF EXEC:

Change	By...
-DDGEN	Replacing the FILEDEFS FILNM001-FILNM100 with the FILEDEFS for the first three files to be used to populate the Dictionary/204. Add a FILEDEF for each file mentioned in the CCAIN. Delete any unneeded FILEDEFS.
MODE	Altering the file mode for OUTFILE and OUTDDGN. If necessary, change the file name and file type.

Modifying DDGEN EXEC

Check the file mode for the DDGEN CCAIN file.

Change SYSOPT from 176 to 177.

Modifying the DDGEN CCAIN file

Change the DDGEN CCAIN:

Replace...	With...
USERID	Login account with ordinary user privileges.
PASSWORD	Password of the login account.
ENTER FILENAME 001	File name for the first Model 204 file used to populate the Dictionary/204.
ENTER FILENAME 001 PASSWORD	Password (if one exists) for the first Model 204 file used to populate the Dictionary/204. Leave a blank line if no password has been assigned for the file.

For each additional file, insert in the CCAINs, after the first file's name and password and before the END statement, the next file name. On the next line insert the appropriate password or leave a blank line.

Repeat as needed. You can specify up to 100 files.

Add the following additional User 0 parameter:

```
LRETBL=4000
```

Running DDGEN

To run DDGEN, enter:

```
EXEC ONLINE NODCSS BYPASS D204 DDGEN
```

Setting up D204 DDGEN based on Dictionary/204 files

Use D204 DDGENSET to select a set of files that are already defined in the Dictionary/204 to run through D204 DDGEN. This may be necessary if the attributes of files in the Dictionary/204 have not been filled in, or if Dictionary/204 has not been kept synchronized with the actual Model 204 files.

Using D204 DDGENSET

D204 DDGENSET reads your installation's Dictionary/204 and automatically creates CCAIN and FILEDEF statements from it. They are used to update the files and corresponding fields and procedures in the Dictionary/204 based on the actual Model 204 files.

D204 DDGENSET builds a list of file names from the file entries found in Dictionary/204. That file list is then passed as input to D204 DDGEN to update or load the full file information into your Dictionary/204.

Each entry must contain a valid DDname attribute value. No other attributes are required. If the entry contains valid data set information, a valid FILEDEF statement is generated; otherwise, the FILEDEF statement must be edited to fill in the data set name. The data set name is required in the Dictionary/204 to use this file through the Dictionary/204 File Management Facility.

If more than 100 files are in your Dictionary/204, which is the limit for D204 DDGEN, D204 DDGENSET creates multiple sets of CCAIN and FILEDEF statements.

D204 DDGENSET generates FILEDEF statements and CCAIN into the OUTDDG1 file.

Modifying the D204FDEF EXEC

Change the D204FDEF EXEC to indicate the correct file mode for OUTFILE and OUTDDG1:

Replace...	With...
MODE	File mode for OUTFILE and OUTDDG1.

Modifying the DDGENSET EXEC

Check the file mode of the CCAIN file.

Modifying the DDGENSET CCAIN

Change the DDGENSET CCAIN:

Replace...	With...
USERID	Login account with ordinary user privileges.
PASSWORD	Password for the login account.
METADATA PASSWORD	Password, if one exists, for the METADATA file. Enter a blank line, if a password is not assigned to the file.
DATALINK PASSWORD	Password, if one exists, for the DATALINK file. Enter a blank line, if a password is not assigned to the file.
M204PROC PASSWORD	Password, if one exists, for the M204PROC file. Enter a blank line, if a password is not assigned to the file.

The file name and password specifications are read as input by D204 DDGEN. If the specified files do not need passwords, supply blank lines.

In the User 0 parameters, remove MINBUF=10 and add LQTBL=800.

Running DDGENSET

To run DDGENSET, enter:

```
EXEC ONLINE NODCSS BYPASS D204 DDGENSET
```

Editing the OUTDDG1 file

Edit the OUTDDG1 file as follows:

1. Edit the FILEDEF statements to specify any missing data set information and to delete files that do not actually exist as Model 204 files.
2. If any Model 204 files have more than six data sets, only the first six data sets have FILEDEF statements. Add FILEDEF statements for the additional data sets.
3. Edit the CCAIN to supply passwords for secured files.

This file can now be used as input for DDGEN.

Updating files using delayed processing

Use D204 DDFIMCMD to update your Model 204 files based on pending commands specified in the Dictionary/204 File Management facility. Pending commands can be executed for one or more files.

Listing files that have pending commands

Enter the following commands online to get a list of the files that have pending commands. The process will generate a prompt for an output data set. If no data set name is entered, the list appears on the terminal:

```
OPEN METADATA  
OPEN M204PROC  
INCLUDE FIMO.PENDFILE
```

For information on the File Management facility, see the *Model 204 Dictionary/204 and Data Administration Guide*.

Modifying D204FDEF EXEC and DDFIMCMD CCAIN files

1. Change the D204FDEF EXEC by modifying *-DDFIMCMD* to replace the sample FILEDEF statements and add FILEDEF statements for each Model 204 file to be updated by pending commands. There is no limit to the number of files to be processed.

2. Change the DDFIMCMD CCAINs:

Label	Change
USERID	Login account with system manager privileges and privileges to start subsystem File Management. You can set up the account to run only in batch without the authority to run online by setting the terminal list option to NONE. For information about the LOGCTL command, refer to the <i>Model 204 Command Reference Manual</i> .
PASSWORD	Password for the login account.
FILENAME1 FILENAME2 FILENAME3	File names of the Model 204 files that have commands executed against them by the File Management facility. Delete or add occurrences as needed.

3. To execute all pending commands against all files, specify the string *ALL instead of listing the file names. The line with *** must follow the file names or *ALL.

LOGADD parameter

Set the LOGADD parameter to the number of files in order to have commands executed against them. Passwords are added for these files and are then deleted during the run. This action is performed because users who request that these commands be executed are not logged in during this particular run. Their privileges are checked when they request the commands online.

Reinitializing M204TEMP

When to use D204 DDTINIT

Use D204 DDTINIT to reinitialize the M204TEMP file when the file has become physically inconsistent or to change parameters that cannot be reset, such as CSIZE. D204 DDTINIT creates, initializes, defines fields, and rebuilds the necessary data records in M204TEMP.

If M204TEMP becomes logically inconsistent or extraneous records are left in M204TEMP when a user is restarted, the data records are rebuilt automatically by the Dictionary/204 and Workshop/204 subsystems. The data records can also be rebuilt Online by the Dictionary/204 administrator. For more information, refer to the *Model 204 Dictionary/204 and Data Administration Guide*.

Modifying DDTINIT CCAIN

Change the DDTINIT CCAIN:

Replace...	With...
USERID	Login account with superuser and file manager privileges.
PASSWORD	Password for the USERID account.
XX	Appropriate parameter values for your installation. Change in the CREATE statement for M204TEMP. Refer to the DDINST CCAIN or DDRINST CCAIN modified with installation or upgrade for the initial parameter settings for your installation.

Invoke the D204 DDTINIT EXEC by issuing the following command:

```
D204 DDTINIT
```


8

Installing Connect★ Support

In this chapter

- Overview
- Installing SQL
- Installing SQL catalog facilities and optional subsystems
- Reinstalling SQL catalog files and optional subsystems
- Loading the demonstration database DDL
- Verifying SQL installation

Overview

This chapter describes tasks required to install Connect★ on the mainframe server. It is assumed you are installing this product as part of Model 204 installation or reinstallation.

“Installing SQL” on page 118 contains the main installation steps for Connect★. From there, you are referred to other sections in this chapter, other sections in this manual, or to other manuals. The installation tasks required for the client part of the installation are documented separately.

Additional Connect★ client installation instructions

For Connect★ installation, some of the information required is contained in this manual; some additional information is in the *Model 204 SQL Connectivity Guide* and the *Model 204 SQL Server User's Guide*; and most of the additional information is in the appropriate Model 204 Connect★ manual.

Before you begin

Dictionary/204 must be installed before the optional SQL catalog subsystems are installed.

Installing SQL

You install the mainframe component of Connect★ as part of Model 204 installation or reinstallation.

Product and feature dependencies

Connect★ itself requires the installation of other Model 204 products and features; see Table 8-1. The SQL installation steps in Table 8-2 include directions and pointers for the installation of these products and features.

Table 8-1. Model 204 SQL product and feature dependencies

If installing...	You must install...
Connect★	Horizon/LE SQL catalog
SQL catalog subsystems	CCACATREPT, CCATSF, and Dictionary/204

SQL installation steps

Follow the SQL-specific steps and reminders in Table 8-2 in your Model 204 installation or reinstallation.

Table 8-2. SQL installation steps

Step	Action
1.	Install the Model 204 nucleus. See Chapter 3.
2.	Install Dictionary/204 if you are installing the optional SQL catalog subsystems. Follow the Dictionary installation information in Chapter 5 and Chapter 6.
3.	Review the SQLALLOC EXEC's file-specific parameters for the SQL catalog and optional catalog subsystem support. You may want to specify values other than the defaults.
4.	Install or reinstall the SQL catalog and, optionally, the SQL catalog subsystems. See page 119 through page 122 for more information.
5.	Install Connect★. See the Connect★ <i>Suite Installation and Programming Guide</i> .
6.	Verify your installation. See page 124.

Installing SQL catalog facilities and optional subsystems

Connect★ clients require the Model 204 SQL catalog to access their data. The catalog is a Model 204 file, CCACAT, that maps Model 204 files to SQL tables and Model 204 fields to SQL columns. Before you can run an SQL application against an existing Model 204 file, you need to use a PC/Client utility to define the file and its field descriptions to the catalog. For details, see "Populating the SQL Catalog" in the Connect★ *Suite Installation and Programming Guide*.

The Model 204 SQL Table Specification facility, CCATSF subsystem, is a tool for defining Model 204 files to the SQL catalog. The Model 204 SQL Catalog Reporting facility (CCACATREPT subsystem) provides reports of the contents of the SQL catalog. Unlike the SQL catalog itself, these catalog facilities are not required for SQL processing.

If you have already installed the SQL catalog or subsystems

If you installed SQL catalog facilities or subsystems as part of installing a prior release of Model 204, skip to the section "Reinstalling SQL catalog files and optional subsystems" on page 122.

Using CCACAT

CCACAT, the SQL catalog file, is populated and updated using SQL DDL. It is also maintained under Model 204 as a Model 204 file.

Access to CCACAT for purposes other than normal SQL installation, operation, and reporting, and other than normal Model 204 operations like sizing, reorganizing, and recovery is highly discouraged.

TSFPROC: File organization changed

TSFPROC, the procedure file for the CCATSF application subsystem, has occasionally filled due to its use as both the procedure file and as a temporary work file. To avoid this problem TSFPROC has been redefined with FILEORG bit X'24', meaning Unordered, Reuse Record Number, and also BSIZE has been increased.

It is recommended that you edit and run the SQLINST job to restore the new version of TSFPROC from the installation software, although you may do this update at a later time.

SQL catalog and subsystem installation steps

The SQL catalog and subsystem support described in the following sections assume that you have completed the installation steps in Table 8-2 up to the instruction to review the SQLALLOC EXEC.

Follow the steps in Table 8-3 to install the SQL catalog and catalog subsystems. Some steps apply only to the catalog subsystems. You can skip these, if you are not installing the subsystems.

Table 8-3. SQL catalog and subsystem installation steps

Step	Applies to...	Action
1.	SQL files	Use the SQLALLOC EXEC to allocate CCACAT, the SQL catalog file, and allocate the SQL catalog subsystem procedure files. Run the SQLALLOC EXEC after reviewing and modifying it as described in the job comments. Issue this command: SQLALLOC
2.	SQL catalog and subsystems	Run the SQLINST EXEC after reviewing and modifying it as described in the job comments. Use the SQLINST EXEC to create CCACAT and the SQL catalog subsystem procedure files. Issue this command: ONLINE NODCSS BYPASS SQLINST
3.	SQL catalog	Bring up your SQL-configured Model 204 Online. The online job configuration requirements for SQL, described in the <i>Model 204 SQL Connectivity Guide</i> , include the following: <ul style="list-style-type: none"> • FILEDEF for the SQL catalog (CCACAT file) • Additional or modified CCAIN parameters for SQL processing • Thread definitions for IODEV 19 (SQL threads) and/or IODEV 49 (RCL threads) • DEFINE LINK, DEFINE PROCESSGROUP, and DEFINE PROCESS commands for Horizon/LE link • OPEN LINK command for Horizon/LE link If you are installing the catalog subsystems, include the items from Step 4.
4.	Subsystems	Make sure that the SQL-configured Model 204 Online EXEC includes: <ul style="list-style-type: none"> • FILEDEF for a sequential output data set to which generated DDL is written. You may want to define a file for each subsystem to avoid mixing the two outputs. The file name must begin with <i>OUT</i>, and you must set LRECL to 80 and RECFM to FM or FS. Using FS (fixed standard) prevents the printing to the file of carriage control characters. The DISP parameter should be MOD to permit multiple writes to the file before processing. • FILEDEFs for TSFDATA, TSFPROC, and CATPROC. • FILEDEFs for the DICTIONARY subsystem management files: METADATA, DATALINK, and CCASYS. Back up these files before installing the SQL subsystems. • The SYSOPT parameter is set to one, that is, the X '01' bit is on, to support Model 204 subsystem operation.
5.	Subsystems	Install the SQL subsystems.
6.	SQL catalog	Populate the SQL catalog to prepare it for SQL processing. For information about loading the catalog with the installation software sample DDL that defines the Model 204 demonstration database, see page 124. For information about loading DDL into the CCACAT SQL catalog, see the Connect* Suite Installation and Programming Guide, Page 2-6 (Populating the SQL Catalog).

Installing SQL catalog subsystems

Install the SQL subsystems, the Table Specification facility, CCATSF, and the Catalog Reporting facility, CCACATREPT, as described in the following sections. Be sure that Dictionary/204 is installed and a SQL-configured Model 204 Online is running.

Installing CCATSF

To install CCATSF:

1. Make sure that the Online is up.
2. Enter:


```
OPEN TSFPROC
```
3. Enter:


```
IN TSFPROC INCLUDE INSTALL.CCATSF
```
4. Specify a CCATSF subsystem Administrator ID and passwords, if any, for the CCASYS, METADATA, and DATALINK files. If you want, you can change the Administrator ID later through the Dictionary.
5. See the *Model 204 SQL Server User's Guide* for information about using the TSF. For example, CCATSF must first be started with the START SUBSYSTEM command before users can log in.

Installing CCACATREPT

To install CCACATREPT:

1. Make sure that the Online is up.
2. Enter:


```
OPEN CATPROC
```



```
IN CATPROC INCLUDE INSTALL.CCACATREPT
```
3. Specify a CCACATREPT subsystem Administrator ID and passwords, if any, for the CCASYS, METADATA, and DATALINK files. If you want, you can change the Administrator ID later through the Dictionary.
4. See the *Model 204 SQL Server User's Guide* for information about using CCACATREPT. For example, CCACATREPT must first be started with the START SUBSYSTEM command before users can log in.

Enabling Connect★ access

To provide your users with access to Connect★:

1. Install Horizon or Horizon/LE as described in Chapter 3.
2. Install Connect★, as directed in the Connect★ Suite *Installation and Programming Guide*.

Reinstalling SQL catalog files and optional subsystems

This section contains instructions for rebuilding or updating your existing CCACAT SQL catalog file.

SQLRINST job for restoring CATPROC and TSFPROC

The SQLRINST job handles the restore of CATPROC and TSFPROC. This job defines two fields, MINIMUM_SCALE and MAXIMUM_SCALE, in CCACAT, and executes ODBCTABLES.INSTALL to update the System Tables in CCACAT. It also updates a previously existing SQL Catalog, CCACAT file, that was created in a prior release.

If you want to install the optional subsystems, CCATSF and CCACATREPT, see "Installing SQL catalog subsystems" on page 121.

At this time, you might want to rebuild the SQL Catalog to take advantage of newly installed System Table features. (See "Rebuilding the SQL catalog" on page 122.) However, this is not necessary; previously defined tables will function as they did before.

Rebuilding the SQL catalog

To rebuild your catalog, you must perform these tasks:

- Recreate the CCACAT file and reprocess all user DDL into the catalog using the new ONLINE module and the Connect★ CVI, PC/Client application. See "Populating the SQL catalog" in the Connect★ Suite *Installation and Programming Guide*.
- Use the new CATPROC file, which contains procedures to create the CCACAT file and store queryable catalog definitions in it.

To rebuild the catalog:

1. Start CCACATREPT and use it to generate DDL based on the existing data within your catalog. At the Model 204 command prompt enter:

```
START SUBSYS CCACATREPT
CCACATREPT
```

Use option 1 from the main menu and supply the following parameter values:

```
SCHEMA NAME :          *
AUTHORIZATION ID :     *
TABLE/VIEW NAME :     *
```



```
DDL STATEMENT TYPES: TVG
USE COMMAND ARG:      (to a machine readable file)
```

Route the output to a machine-readable file.

Note: Verify that you have correctly obtained a machine readable file of DDL before proceeding with the recreation of the CCACAT file.

2. Dump your existing CCACAT file to ensure that you have a backup of the old file in case any problems occur.
3. Run SQLRINST.
4. Review your DDL and the rules for updateable tables for ODBC. Make any desired changes (you might, for example, want to assign primary keys).
5. Use this DDL as input to the DDLUTIL utility (which you can run from the PC).

Note: If you are defining large schemas, you might want to break the schema definition into multiple transactions. To do this, add SET SCHEMA statements as shown in the following sample.

```
Before SET 1)  CREATE SCHEMA FOO ...
SCHEMA      CREATE TABLE ...
added       CREATE TABLE ...
              CREATE TABLE ...
              CREATE VIEW ...
```

```
After SET  1)  CREATE SCHEMA FOO ...
SCHEMA      CREATE TABLE ...
added       2)  SET SCHEMA FOO
              3)  CREATE TABLE ...
              4)  CREATE TABLE ...
              5)  CREATE VIEW ...
```

Without a SET SCHEMA statement, the entire schema definition starting at line 1 is treated as one large transaction. With the SET SCHEMA statement, there are five separate transactions as identified by the line numbers above.

Updating your catalog without rebuilding

If you choose not to completely rebuild your catalog, then you must update your catalog by running SQLRINST.

1. Be sure that your SQL catalog facilities, CCATSF and CCACATREPT, are not active.
2. Edit SQLRINST EXEC to modify the FILEDEFS and SQLRINST CCAIN to modify the userid and password.

3. Run SQLRINST by entering:

```
ONLINE NODCSS BYPASS SQLRINST
```

Note: This procedure does not enable potential performance improvements of the full ODBC migration. Therefore, if you want to use ODBC in the future, you *must* convert the catalog using the instructions beginning on page 122.

Setting SQL runtime parameters

The following default and recommended parameter values have been modified to improve SQL and RCL processing. For more information, see Table 2-2 in the *SQL Connectivity Guide*.

Parameter	New value
LIBUFF	5000
LPDLST	32760
LTTBL	2000
LQTBL	2000
SERVSZ	350000
SQLBUFSZ	100000
SQLIQBSZ	32752

Loading the demonstration database DDL

If SQL clients want to use the Model 204 demonstration database, you must first define, or map, it to the SQL catalog. Rocket provides a sample SQL DDL stream that maps the demonstration database to SQL views and tables.

For information about loading the catalog from a Connect★ Workstation, see the Connect★ *Suite Installation and Programming Guide*.

Renaming your schema

Most of the examples for the Connect★ online help are based on the demonstration database and use DEMO as the schema name. You can either use this name or choose your own. If you do use a different schema name, be sure to edit the DEMODDL job and any example files you use to replace DEMO with your own schema name.

See the Connect★ *Suite Installation and Programming Guide* for information about renaming the schema.

Verifying SQL installation

Connect★ clients can immediately test their SQL installation.

You can test your Connect★ installation using the CATALOG2 program that is provide in the Connect★ Suite ODBC installation. The CATALOG2 program issues an SQL query of the SQL catalog table, the Model 204 CCACAT file.

For more information on testing your installation, see the *Connect* Suite Installation and Programming Guide*.

A

Object Modules

In this appendix

- Object modules

Object modules

The following table lists the object modules that are link-edited into the ONLINE, IFAM1, or BATCH2 configurations of Model 204.

A name in brackets, such as [Horizon], in the Description column indicates the optional feature with which a module is associated.

Module	O N L I N E	I F A M 1	B A T C H 2	Description
ANXV	X			ANALYZE command processor
APSY	X			Application subsystem support
APSZ	X			Application subsystem support
ARTH	X	X		Arithmetic compiler
ATIO	X	X		Audit trail formatting routines
AUTHCMS	X	X		Authorization controller
BEGN	X	X		MODEL 204 initialization
BKUP	X			File DUMP/RESTORE commands

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
BOUT	X	X		Transaction backout feature
BXCI	X	X		Ordered index support
CDTB	X			\$CODE and \$DECODE functions
CHKP	X	X		Checkpoint/restart
CMFS	X			Full screen console I/O
CMIF			X	IFAM/IUCV interface
CMIO	X			Line-by-line console I/O
CMIS	X	X		Server communication interface
CMPB	X	X		Boolean compiler
CMPF	X	X		File maintenance compiler
CMPI	X	X		Compiler sequential I/O
CMPJ	X	X		Miscellaneous routines compiler
CMPL	X	X		Compiler control routine
CMPR	X	X		Retrieval compiler
CMPS	X	X		General compiler subroutines
CMPU	X			Tprocess statement compiler
CMPY	X			External record I/O image compiler
CMPZ	X			User Language screens and menus compiler
CNST	X	X	X	Support for RMODE 31 processing
CRFS	X			Full screen remote User Language routine
CRIO	X			Line-by-line remote User Language routine
CRSQ	X			CRAM remote SQL thread manager
CTCK	X	X		Password Expiration
CTTC	X	X		Account file security routines
CUST	X	X		Date/time stamp field data
DBCS	X	X		Double-byte character set subroutines

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
DEBUG	X	X		Debugging facilities
DDLMM	X	X		Extended field definition, displays
DEFN	X	X		USE command support
DFIL	X			Distributed file handling [PQO]
DKBM	X	X		Disk buffering monitor
DKIO	X	X		Disk I/O routines (Model 204 files)
DSPS	X	X		Model 204 dataspace support
DTME	X	X		Date and time routines
DTSR	X	X	X	Date/time stamp routine
EDFS	X			Full screen editor routine
EDIT	X			Line editor
EVAR	X	X		Arithmetic evaluator
EVBT	X	X		Ordered index
EVFM	X	X		File maintenance evaluator
EVIO	X	X		Sequential I/O evaluator
EVLU	X			Evaluator subr. for interprocess communication
EVMI	X			External record I/O evaluator
EVNM	X	X		Numeric range evaluator
EVNU	X	X		Evaluator control and booleans
EVNV	X	X		Evaluator control and booleans
EVSO	X	X		Sorted and hash key files evaluator
EWDP	X	X		Early warning recording and display module
FENQ	X	X		File enqueueing routines
FILE	X	X		File OPEN/CLOSE routines
FLIO	X	X		FLOD and deferred update routines
FLL3	X	X		Z command processor

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
FLOD	X			Fast file load utility
FLXT	X			FLOD exit controller
FUNB	X			PQO functions
FUNC	X	X		Basic functions
FUND	X			Model 204 math functions and FORTRAN subroutines
FUNDLE	X			Model 204 LE mathematical functions and subroutines (with LENV module)
FUNE	X			Editing functions
FUNF	X	X		Additional functions
FUNR	X			Sockets module
FUNU	X			User functions
FUNV	X	X		User functions used by Dictionary
FUNW	X	X		Functions for new products
FUTL	X	X		File utility routines
GLOB	X	X		Globals
GRUP	X	X		File groups features
IFAM	X	X		IFAM function routines
IFAS	X	X		IFAM function routines
IFCF	X	X		Interface to call C functions
IFCM			X	IFAM interface routines
IFDM	X	X		IFAM interface routines
IFET	X			PQO support
IFID	X			IFDIAL single user support
IFIF			X	IFAM outboard interface subroutines
IFII	X			IFAM2 support
IFII1		X		IFAM1 dynamic loading
IFSQ	X	X		Routines to call IFAM from SQL

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
IF6C	X			Client side of LU 6.2 [Horizon]
IF6S	X			Server side of LU 6.2
IOFS	X	X		I/O interface — full screen device
IOIF	X	X		I/O interface — device independent
IOPC	X			I/O interface — pc device
IPIC	X			Interprocess Inbound Conversation manager
IUCV			X	IUCV interrupt exit routines
JRIO	X	X		Journal I/O routines
KOMM	X	X		Common storage and parameter values
LARB	X	X		Logical I/O arbitration routines
LBUF	X	X		Logical I/O buffer handling
LCCTOS	X	X		Concatenation for logical I/O-OS
LCIO	X	X		Logical I/O module for streams
LCMD	X	X		COPY and OFFLOAD command processor
LENV	X			Initialize LE environment to allow mathematical \$functions to use LE mathematical services (with FUNDLE)
LIOS	X	X		Logical I/O subsystem
LPIO	X	X		Parallel I/O streams
LRIO	X	X		Ring I/O streams
LSUB	X	X		Logical I/O common routines
LUCO	X			Control operator program for CNOS Horizon
LU62	X			Horizon controller
MAIN	X	X		Command loop, error routines
MLNK	X	X		CCA debugging features
MSGP	X	X		Error message routines
MSGS	X	X		Model 204 error message table

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
MSGU	X			User error message table
M204END	X	X		DCSS generation
M204ONLN	X			DCSS generation
M204RENT		X		DCSS generation
M204XIF1		X		DCSS generation
M204XONL	X			DCSS generation
NLANG	X	X		US and other language tables
ONLM	X			Online monitor package
PARM	X	X		Parameter VIEW and RESET routines
PARS	X	X		CCAIN and command parsing
PROC	X			Procedures subroutines
PROD	X	X		Procedures subroutines
PRPR	X			Process-to-process communications
PSEC	X			Procedure security
PTCH	X	X		Early warning code fix area
PTCN	X	X		Patch Number table
RCLK	X	X		Record Locking manager
RCVY	X	X		Recovery RESTART command
REMQ	X	X		Record and resource enqueueing routines
RM24	X	X		Implementation of support for RMODE 31 processing
RSQL	X	X		Remote SQL and Pattern Matcher support
RUTL	X			Media recovery
SBDS	X			PQO support
SBD2	X	X		Common SQL or PQO routines
SBFM	X	X		File maintenance subroutines
SBIO	X	X		Sequential I/O subroutines

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
SBNU	X	X		Miscellaneous subroutines
SBNV	X	X		Conversion subroutines
SBPT	X	X		Subroutines for pattern matcher
SBSQ	X	X		BAL routines call from C
SBXM	X	X		Ordered index maintenance subroutines
SCHD	X	X		Priority scheduler
SDIO	X	X		Sequential data sets I/O routines
SESM	X			Session mgmt for interprocess communication
SM62	X			Horizon implementation of LU session management
SORT	X	X		Sort functions
SORU	X	X		UL sort compiler
SORX	X	X		Sort user exits
SPCL	X	X		Support for specials
SQIO	X	X		BSAM, QSAM sequential I/O
STAT	X	X		Statistics and accounting routines
SUPF	X	X		System independent supervisor functions
SUSE	X			Directed output (USE) features
SVIO	X			Server feature
SWDB	X			Server Swapping Debug Facility
SYIOOS	X	X		System dependent I/O functions
SYSFXA	X	X		XA system dependent supervisor functions
SYSXI	X	X		SYSFXA extension (IBM)
TCIU	X			TCP/IP socket programming interface
TC62	X			Logical driver for TCP/IP communication
TCSE	X			TCP/IP sockets extended programming interface

Module	O	I	B	Description
	N	F	A	
	L	A	T	
	I	M	C	
	N	1	H	
	E		2	
TFRM	X			TRANSFORM command
ULIFOS			X	BATCH2 utility module
UTIOCMS			X	I/O subroutines for utilities
VMIF	X			IUCV interface (Directed Output under z/VM)
VTIU	X			VTAM interface routines
VT62	X			Interprocess communication with LU 6.2 [Horizon]
VT75	X			VTAM 3270 support (full screen)

B

Utilities and EXECs

In this appendix

- Model 204 Utility commands
- z/VM utility commands
- M204UTIL functions
- Internal EXECs
- M204GEN command
- M204SAVE command
- Maintaining the MAINT204 machine

Model 204 Utility commands

This section discusses these Model 204 utility commands:

- AUDIT204
- MERGEJ
- ZBLDTAB
- ZCTLTAB

Model 204 also provides the UTILC utility, which provides information about the Model 204 checkpoint process associated with recovery, and UTILJ utility, which interprets a Model 204 journal file. Refer to the *Model 204 System Manager's Guide* for more information about them.

AUDIT204 utility program

The AUDIT204 utility program prints a Model 204 journal file or produces a statistical report from information written to the journal during a run. If a statistics report is produced, it requires a SORT utility that can be invoked dynamically. AUDIT204 output is produced either as a printer spool file or in a CMS disk file. Use the AUDIT204 EXEC procedure to run the AUDIT204 utility under CMS. The input to AUDIT204 is the AUDIT204 CCAIN file. A sample for installation on the Model 204 maintenance virtual machine's 193 disk is provided with other distributed material.

Syntax The format of the AUDIT204 command is:

```
AUDIT204 {datasetname | filename filetype} filemode
```

- Where**
- *datasetname* specifies the name of the journal data set on a variable-format disk with qualifiers separated by blanks. If the specified disk is variable-format and no data set name is specified, the default journal data set name is M204.JOURNAL.
 - *filename* and *filetype* specify the name and type of the journal file on a CMS-format disk. If not specified for a disk in CMS format, the default journal file name is M204 JOURNAL.
 - *filemode* specifies the mode of the disk holding the specified journal file.

For more information about the AUDIT204 utility and its options, refer to the *Model 204 System Manager's Guide*.

MERGEJ utility program

The MERGEJ utility program merges several journal files into a single file. It is generally used when performing Model 204 media recovery using the REGENERATE command. Under CMS, the MERGEJ command requires a SORT command that can be invoked dynamically. Use the MERGEJ EXEC procedure to run the MERGEJ utility under CMS.

Syntax The format of the MERGEJ command is:

```
MERGEJ fileid / fileid [●●●] / outfileid
```

- Where**
- *fileid* specifies the file identifier and mode of a journal file. A file identifier can be either the name of a data set on a variable-format disk or the name, type, and mode of a file on a CMS-format disk. If you specify a data set, separate the qualifiers with blanks. Use a slash (/) to separate two file IDs.
 - *outfileid* specifies the identifier and mode of the output file that is to contain the merged journal file. The output journal file must reside on a CMS-format minidisk.

Example MERGEJ JOURNAL DAY1 F / JOURNAL DAY2 F / MERGED JOURNAL A

The number of journal files that can be merged with one MERGEJ command is determined by the capacity of the available SORT/MERGE utility.

ZBLDTAB utility program

Use the ZBLDTAB EXEC procedure to initialize a Model 204 password table.

Syntax The format of the ZBLDTAB command is:

```
ZBLDTAB {datasetname | filename filetype} filemode
```

Where

- *datasetname* specifies the name of the password table data set on a variable-format disk with the qualifiers separated by blanks. If the specified disk is variable-format and no *datasetname* is specified, the default password table data set is M204.CCASTAT.
- *filename* and *filetype* specify the name and type of the password table file on a CMS-format disk. If not specified for a disk in CMS format, the default password table file name is M204 CCASTAT.
- *filemode* specifies the mode of the disk where the password table file to be processed resides.

ZCTLTAB utility program

Use the ZCTLTAB utility program to invoke the M204ZCTL module to

- initialize the password security feature in an existing MODEL 204 password table, or
- update the password security parameters in a password table that was previously transformed.

Syntax The format of the ZCTLTAB command is:

```
ZCTLTAB {oldfileid [newfileid [outfileid]]} [(parms)
```

Where

- *oldfileid* specifies the file identifier of the existing CCASTAT file. If no *datasetname* or *filename filetype* is entered, it defaults to M204 CCASTAT. The filemode is required.
- *newfileid* specifies the file identifier of the new CCASTAT file that ZCTLTAB will create when the file is first transformed to the new format. If the old CCASTAT file was previously transformed, the old CCASTAT file is updated and this parameter is ignored. If *newfileid* is omitted, it defaults to M204 NEWSTAT with the same filemode as *oldfileid*.

- *outfileid* specifies the file identifier of the output message file on a CMS disk. If *outfileid* is not entered, it defaults to ZCTLTAB CCAOUT A. The filemode must represent a CMS-format disk.

The format for a file identifier is

{datasetname | filename filetype} filemode

where:

datasetname specifies the name of the password table data set on a variable-format disk, with the qualifiers separated by blanks.

filename specifies the name of the password table file on a CMS-format disk.

filetype specifies the type of the password table file on a CMS-format disk.

filemode specifies the mode of the disk where the password table file to be processed resides. If *datasetname* or *filename filetype* is entered, *filemode* must also be entered.

- *parms* indicates one or more of these parameters:

[EXP *n1*] [WARN *n2*] [PURGE *n3*]

where:

n1 is the number of days until a password expires. Valid values are from 0 to 255.

n2 is the number of days before expiration that a warning message is issued to the user. Valid values are from 0 to 255.

n3 is the number of days after expiration that a user's entry remains in the password table. After this grace period, the entry is deleted. Valid values are from 0 to 255.

Restrictions

- At least one parameter must be entered.
- The default parameter value is the current value in the file. If this is the first time ZCTLTAB has been run on a file, the default value is 0.
- If a previous version of CCASTAT is being converted to the new format, EXP is required.
- WARN must be 0 if EXP is 0.

Usage Notes

If a variable-format disk is specified for NEWSTAT, the file must have first been created with M204UTIL ALLOC.

When running ZCTLTAB for the first time to transform an old-format file, any existing file with the same fileid as NEWSTAT will be overwritten.

Examples

1. Take all the default filenames and filetypes; all files are on disk A.

```
ZCTLTAB A (EXP 60 WARN 5 PURGE 90
```

2. Take all the default filenames and filetypes, but all files are on disk B.

```
ZCTLTAB B / B / B (EXP 60 WARN 5 PURGE 90
```

3. CCASTAT is M204 CCASTAT A, NEWSTAT is MYNEW CCASTAT D, and the messages file is ZCTLTAB MSG D.

```
ZCTLTAB A / MYNEW CCASTAT D / ZCTLTAB MSG D (EXP 60 WARN 5  
PURGE 90
```

4. CCASTAT is PROD CCASTAT B. It has already been converted to the new format and you just want to change the expiration period to 120 days.

```
ZCTLTAB PROD CCASTAT B (EXP 120
```

z/VM utility commands

This section discusses the following Model 204 for IBM z/VM utility commands:

- M204APND
- M204CMS
- M204CRYP
- M204FDEF
- M204LDEF
- M204XFER

Model 204 also provides the VT204 command, which is used to examine and control the GCS service machine of the optional Model 204 VTAM Interface. It is discussed in “VTAM Interface” on page 47 and in the *Model 204 Horizon: Intersystem Processing Guide*.

M204APND utility program

The M204APND program concatenates file definitions for the Model 204 environment so that several files, including sequential files, can be used as one file. The M204APND command defines a file that is to be concatenated to other files with a specified name. Refer to the *Model 204 File Manager's Guide* for information on using M204APND and M204LDEF to back up and restore multtape Model 204 files.

Syntax

The format of the M204APND command is:

```
M204APND DDname parameters [(options [ ])]
```

Where

- *DDname* specifies the DD name to which the file being defined is to be concatenated.
- *parameters* and *options* specify standard FILEDEF or M204FDEF command parameters and options.
If a device address is specified, a 4-digit address (390 subchannel address) can be specified.
A FILEDEF command that has the CLEAR option for a *DDname* representing a concatenation causes all file definitions in the concatenation to be cleared.

The files in a concatenation are processed in the order that the associated FILEDEF, M204APND, and M204FDEF commands are issued.

M204CMS command

The M204CMS program is the interface between Model 204 and CMS. It provides system services in the CMS environment during execution of a variety of Model 204 facilities.

The M204CMS command usually is used in the EXEC procedures that provide higher level interfaces to Model 204 (for example, FASTLOAD or ONLINE).

Syntax

The format of the M204CMS command is:

```
M204CMS [module] [[(keyword value) [,•••] ( )]]
```

Where

- *module* specifies the name of the Model 204 program module that is to be executed. The default is M204ONLN.
- *keyword* and its associated *value* specify parameters that are passed to the Model 204 module in the format of an OS PARM field. Separate the keyword and its associated value with a blank and insert the keyword and value in the PARM field. Separate successive keyword-value pairs with commas.
An exception is recognized when the value is specified as a percent sign (%). In this case, only the keyword is inserted in the PARM field separated with commas from neighboring operands.

M204CRYP command

The Model 204 installation contains all the object code required to use the separately purchased DBCS option:

The object modules that enable the use of optional features are distributed in encrypted form and must be decrypted before running the M204GEN

command. You need to invoke the M204CRYP command for each option you are decrypting.

Syntax

The format of the M204CRYP command is:

```
M204CRYP [DECODE] feature-name key ([NOREPLACE | REPLACE])
```

Where

- DECODE causes the Rocket XENCDEC program to decrypt vital elements of a specified feature, enabling the feature.
- *feature-name* is the name of the feature, as defined in the CMS table M204CRYP FEATTBLE. The valid feature names are:

Feature name	Purpose
DBCS	Double-byte character set subroutines

- *key* is the 8-byte numeric key that decrypts the object modules for the specified feature. The key is provided by Rocket.
- *NOREPLACE*, the default, checks that files having the same file ID as the output file do not already exist. If a file with the same file ID exists, an error message is displayed and the M204CRYP command terminates.
REPLACE causes the output file to replace an existing file with the same file identifier.

The M204CRYP command has restricted use and must be issued directly from the MAINT204 machine (Model 204's maintenance virtual machine).

M204CRYP report

The M204CRYP command generates a report that is identified by a CMS file name equal to a feature name and a file type of D_REPT. The report contains:

- FILE IN—CMS file ID of the input file.
- FILE OUT—CMS file ID of the output file.
- CONTROL PARM—Function used (DECODE) and the key value.
- NUMBER OF RECORDS PROCESSED—Total number of records processed.
- Return codes and messages.

M204CRYP return codes and messages

The return codes issued by M204CRYP are described in Table B-1.

Table B-1. M204CRYP return codes and messages

Code	Message	Description
0	Successful completion	All materials were decrypted successfully.
1	Error in an input argument	The error may be a misspelling, an invalid option, or invalid decryption key. Check the command you issued, correct the error, and re-execute it.
2	Error in M204CRYP	This internally detected error return code is not used.
3	Execution of a host, CMS or CP, command failed	The return code and the text of the command are described in an error message. This usually means that the decryption utility failed for some reason. If this is the case, the D_REPT file shows the reason for the decryption failure. Check the report, correct the error, and rerun the M204CRYP command.

Table B-2 lists the return codes and messages that appear in the report produced by the decryption utility. The appropriate diagnostic action is also suggested.

Table B-2. Return codes for the decryption utility

Code	Message	Diagnostic action
0	None	None—decryption was successful.
8	No Control or Parm provided	PARM= on the EXEC card was invalid or missing. This reflects an internal error in the service routine used by M204CRYP. Contact Technical Support for assistance.
12	Invalid command specified	DECODE was not specified on the PARM= on the EXEC card. This reflects an internal error in the service routines used by M204CRYP. Contact Technical Support for assistance.
16	Invalid decryption key specified	Decryption key specified was not exactly eight bytes in length. Verify that the key you have specified correctly matches the key you were supplied by Rocket.
20	Wrong decryption key specified	Decryption key specified does not match the key used to encrypt modules for the feature that you are installing. Verify that the key you have specified correctly matches the key you were supplied by Rocket.

Table B-2. Return codes for the decryption utility (continued)

Code	Message	Diagnostic action
24	Internal decryption xxxx error detected where xxxx can be: <ul style="list-style-type: none"> • Clear text length • Clear checksum • Cipher checksum 	Decryption key specified does not match the key used to encrypt modules for the feature that you are installing, or the material you are trying to decrypt has been modified or tampered with. Verify that the key you have specified correctly matches the key you were supplied by Rocket.

M204FDEF command

You can create file definitions for files on unaccessed variable-format disks in the Model 204 environment using the M204FDEF program.

File definitions describe the files that a Model 204 service virtual machine is to use. Usually, they are created by an EXEC procedure tailored to the particular requirements of an individual configuration or installation.

The CMS FILEDEF command requires that any variable-format disk used by Model 204 be accessed by the standard CMS ACCESS command. As a result, the total number of disks that can be made available to a particular Model 204 service virtual machine is limited. The M204FDEF command overcomes this limitation.

M204FDEF performs a modified FILEDEF function that permits data sets on z/OS or z/VSE volumes to be defined for use with Model 204 without the need to access the volume. As a result, the number of file definitions established for Model 204 on an arbitrary number of disks can be unlimited.

Syntax

The format of the M204FDEF command is as follows:

```
M204FDEF DDname address DSN dsname [(options [])]
```

Where

- *DDname* specifies the DD name of the file being defined.
- *address* specifies the virtual address of the minidisk on which the file to be processed resides. The address can be up to four digits long.
- DSN is the identifier tag for the data set.
- *dsname* specifies the name of the data set as a series of qualifiers separated by blanks. The same conventions and restrictions associated with the DSN operand of the FILEDEF command apply.
- *options* specify any of the valid options of a FILEDEF command for a z/OS or z/VSE data set definition. Consider using the DSORG, RECFM, BLKSIZE, and LRECL options in the Model 204 environment.

M204LDEF command

By using the M204APND command, you can define a file as a concatenation of magnetic tape volumes. A LABELDEF command must be issued for each volume in the concatenation. For more information about the LABELDEF command, refer to your IBM z/VM documentation.

Since the label definition is associated with a file definition through the DDname parameter, an alternative is required to define labels for the second and subsequent volumes in a multi-reel file. The M204LDEF command provides this function.

M204LDEF commands are associated with the file definitions created by the M204APND command in the sequence in which the file definitions are established.

Refer to the *Model 204 File Manager's Guide* for information on using M204LDEF and M204APND to back up and restore multitape Model 204 files.

Syntax

The format of M204LDEF is as follows:

```
M204LDEF DDname [parameters] [(options ●●● ())]
```

Where

- *DDname* specifies the DD name associated with the file to be defined.
- *parameters* specify the standard LABELDEF command parameters. A LABELDEF command that specifies the CLEAR parameter for a DD name that represents a label definition for a concatenated file causes only the first label definition in the concatenation to be cleared.
- *options* specify the standard LABELDEF command options.

M204XFER command

Use the M204XFER command to invoke Model 204 environments that execute in saved segment areas. M204XFER transfers control to the version of the M204CMS module that executes in a discontinuous saved segment (DCSS).

The M204XFER command is usually used in EXEC procedures that provide higher-level interfaces to Model 204 (for example, ONLINE or IFAM1).

M204XFER first attempts to load a designated program segment. It then attempts to load the segment containing the Model 204 CMS Interface. If this second load attempt fails, M204XFER terminates with an error message. Otherwise, the Model 204 CMS Interface is invoked in its saved segment with the same parameter list that was used to invoke M204XFER.

For this module to function, the specified interface segment (M204CMS by default) must be defined. If this segment is not created, the M204XFER module must not be generated.

If the M204XFER module exists and the interface segment does not, M204XFER abandons any attempt to invoke Model 204.

M204XFER can optionally load a second DCSS containing the Model 204 program that is to be invoked by M204CMS.

M204XFER first attempts to load the specified program segment. Regardless of whether this operation succeeds, it then attempts to load the interface segment. If this attempt fails, M204XFER terminates with an error message. Otherwise, the interface is invoked in the DCSS with the same parameter list that was used to invoke M204XFER.

If the specified program segment could not be loaded by M204XFER, a MODULE that has the same name as the segment is loaded by M204CMS.

Syntax The format of the M204XFER command is:

```
M204XFER [module | *] [interface] [(options ... ())]
```

- Where**
- *module* specifies the name of the Model 204 program module that is to be executed. The default is M204ONLN. Entering an asterisk (*) for this option indicates that no program segment is to be invoked.
 - *interface* is the M204CMS interface or M204USR interface. If *interface* is omitted, the default is M204CMS.
 - *options* are passed directly to the interface saved segment.

M204UTIL functions

M204UTIL utility program

The M204UTIL utility program initializes variable-format volumes and allocates, renames, and erases data sets on such volumes.

Syntax The format of the M204UTIL command is:

```
M204UTIL function operands [(options ())]
```

- Where**
- *function* specifies the operation that M204UTIL is to perform. The following lists the valid values for function. Each of these functions is described in more detail in the rest of this section.

Function	Alias	Description
INITIAL	FORMAT	Initializes a minidisk in variable format
LABEL		Changes a variable-format minidisk label
ALLOCATE	CREATE	Allocates space for a variable-format disk

Function	Alias	Description
ERASE	SCRATCH REMOVE	Erases a data set on a variable-format disk
RENAME	CHANGE	Changes the name of a data set on a variable-format minidisk
LIST		Lists data sets on a variable-format minidisk

- *operands* and *options* are function-dependent. An operand can be a data set name, a virtual address, or whatever represents the object of the function. Options represent operand attributes. Minidisk addresses must be specified as three or four digits.

For example, when M204UTIL INITIAL initializes a minidisk for use as a z/OS or z/VSE volume, it takes the following form:

```
M204UTIL INITIAL address label (options
```

Shared DASD

M204UTIL uses reserve/release logic when it updates the Volume Table of Contents (VTOC) of a variable-format disk. Consequently, it can manipulate a volume used by one or more Model 204 service virtual machines. However, erasing data sets on such volumes while they are being used by Model 204 causes unpredictable results.

ALLOCATE function

The M204UTIL ALLOCATE function creates a data set on an initialized volume.

Syntax

M204UTIL ALLOCATE takes the following form:

```
M204UTIL ALLOCATE datasetname {mode | address}
    [(options ••• [ ])]
```

Where

- *datasetname* specifies the qualifiers that comprise the name of the data set to be allocated; separate qualifiers by blanks.
- *mode* specifies the disk mode of the minidisk where the file allocation is to be performed.
- *address* specifies the virtual address of the mindisk where file allocation is to be performed.
- *options* are the options used when allocating space for the files on variable-format disks. Table B-3 describes the supported options.

You must specify the PRIMARY option. You can use other options, but they are not required. Model 204 under CMS does not allocate Secondary quantities; to do so, use z/OS operating system utilities.

Table B-3. M204UTIL ALLOCATE command options

Option	Description
PRIMARY <i>nnnnnn</i> {CYL TRK BLK}	Primary allocation quantity, in cylinders (CYL) or tracks (TRK) for a count key data device, or in blocks (BLK) for a fixed-block architecture. Required.
DSORG {PS DA}	Data set organization. If this option is omitted, no data set organization is specified.
RECFM <i>format</i>	Data set record format. If this option is omitted, no record format is defined.
BLKSIZE <i>nnnnn</i>	Data set block size. If this option is omitted, a value of 0 (zero) is used
LRECL <i>nnnn</i>	Data set logical record length. If this option is omitted, a value of 0 (zero) is used.

The following example initializes a temporary disk for a temporary file (CCATEMP). Do not use temporary disks for permanent files. This example initializes a temporary minidisk, accesses it as M, and allocates space for a data set named SCRATCH.CCATEMP to be stored on it:

```
CP DEFINE T3390 291 15
M204UTIL INITIAL 291 TMP291
ACCESS 291 M
M204UTIL ALLOCATE SCRATCH CCATEMP M (PRIMARY 5 CYL
```

The data set is defined in the ONLINE with a FILEDEF. For example:

```
FILEDEF CCATEMP M DSN SCRATCH CCATEMP
```

ERASE function

The ERASE function erases a data set from a volume.

Syntax

The format of the M204UTIL command ERASE function is:

```
M204UTIL ERASE dataset {mode | address}
```

Where

- *dataset* specifies the qualifiers (separated by blanks) that comprise the name of the data set that is to be erased. For example, if erasing a data set named M204.CMS.CCASERVER, specify the operand M204 CMS CCASERVER.
- *mode* specifies the mode of the minidisk from which the file is to be erased.

- *address* specifies the virtual address of the minidisk where the data set to be erased resides.

INITIAL function

The M204UTIL INITIAL function initializes Model 204-owned variable-format disk(s).

Syntax

The format of M204UTIL INITIAL is:

```
M204UTIL INITIAL address label (options)
```

Where

- *address* specifies the 3-digit hexadecimal virtual address of the minidisk that is to be initialized.
- *label* specifies the 1- to 6-character volume label assigned to the minidisk.
- *options* specify the location and extent of the VTOC. The supported options are described in the following table.

Option	Description
LOCATION <i>nnnnnn</i>	Starting location of the VTOC, in relative track or block number.
EXTENT <i>nnnnn</i>	Number of tracks or blocks to be reserved for the VTOC

If you do not specify values, the default options for a count-key-data (CKD) device are LOCATION 1 EXTENT 2. For a fixed-block-architecture (FBA) device, default options are LOCATION 2 and an EXTENT value that causes the VTOCs to occupy the remaining portion of the first cyclical group on the device.

Model 204 does not support VTOCs that cross cylinder boundaries on count-key-data devices. Indexed VTOCs are neither created nor manipulated by Model 204. Table B-4 shows the capacities of the default and maximum size VTOCs by device type.

Table B-4. Volume Table of Contents capacities

Device	VTOC size	Available DSCBs	Maximum VTOC size	Maximum DSCBs
3380	2 tracks	106	14 tracks	795
3390	2 tracks	100	14 tracks	750
9345	2 tracks	90	14 tracks	675
9332	60 blocks	210	n/a	--
9335	60 blocks	210	n/a	--
9336	60 blocks	210	n/a	--

Note:

Do *not* use the M204UTIL INITIAL function on volumes that are owned by virtual machines other than the MODEL204 virtual machine, for example, other operating systems, or minidisks that contain data that you want to keep.

The initialization process destroys access to all data on the volumes.

The initialization process deletes the VTOC, which has the pointers to the z/OS data sets on the disk, and Model 204 can no longer find the data sets. Enter YES and not Y in response to the confirmation prompt.

LABEL function

The LABEL function changes a variable-format minidisk label.

Syntax The format of the M204UTIL command LABEL function is:

```
M204UTIL LABEL address label
```

Where

- *address* specifies the virtual address of the minidisk to be labeled.
- *label* specifies the volume label assigned to the minidisk.

LIST function

The LIST function lists information about one or more data sets on a specified volume.

Syntax The format of the M204UTIL LIST function is:

```
M204UTIL LIST datasetname {mode | address}
```

Where

- *datasetname* specifies the qualifiers (separated by blanks) of the name of the data set that is to be listed. For example, if the attributes of a data set named M204.CMS.VEHICLES are to be listed, specify the operand M204 CMS VEHICLES.

If the last qualifier specified is an asterisk (*), all data set names beginning with the prefix defined by the preceding qualifiers are listed. For example, an operand of M204 CMS * lists all data sets that have M204.CMS as their first two index levels.

If the last qualifier specified ends with an asterisk (*), the response includes all data sets whose names begin with the preceding qualifiers and are followed by a qualifier that matches the last one up to the asterisk. For example, an operand of M204 CM* lists all data sets that have M204 as their first index level and whose second index level starts with CM.

Specifying the asterisk (*) as the only qualifier lists all data sets on the volume.

- *mode* specifies the mode of the minidisk that contains the data set(s) to be listed.
- *address* specifies the virtual address of the minidisk where the data set(s) to be listed reside.

RENAME function

The RENAME function changes the name of a data set.

Syntax

The format of the M204UTIL command RENAME function is:

```
M204UTIL RENAME oldname {mode | address} (newname)
```

Where

- *oldname* specifies the qualifiers (separated by blanks) that comprise the name of the data set that is being renamed. For example, if renaming a data set M204.CMS.CCASERVR, specify M204 CMS CCASERVR.
- *mode* specifies the mode of the minidisk on which the file to be renamed resides.
- *address* specifies the virtual address of the minidisk where the data set to be renamed resides.
- *newname* specifies the qualifiers (separated by blanks) that comprise the new name to be assigned to the data set.

Internal EXECs

The following EXECs control spooling and cartridge mounts in the Model 204 for IBM z/VM environment.

M204MOUN EXEC

You might need to tailor the sample M204MOUN EXEC supplied with the Model 204 installation software to your installation's environment.

M204SPL EXEC

You can tailor the sample M204SPL EXEC supplied with the Model 204 installation software to your site's environment.

M204UNLD EXEC

You can tailor the sample M204UNLD EXEC supplied with the Model 204 installation software to your site's environment.

M204GEN command

The M204GEN command generates (constructs) the Model 204 modules and segments.

The M204GEN command:

- Concatenates text decks.
- Loads those concatenated files into storage.
- Generates MODULE and MAP files on the MAINT204 (maintenance) virtual machine's 193 disk.

Input to the M204GEN command

The M204GEN command uses the following files as input for generating modules and segments:

- M204USER PARS (distributed with Model 204)
- Loadlists for each configuration (distributed with Model 204)

You do not need to modify these files unless you want to change saved segment definitions from the default definitions.

M204GEN syntax

```
M204GEN {configuration | ALL} ([Ctl cntrl]
  [PROD | TEST] [MODule | SEGment] [NOSAVE | SAVE]
  {FUND | FUNDLE} [HRZN | NOHRZN]
  [PQO | NOPQO])
```

Where

- *configuration* specifies the name of the module or saved segment to generate. You can specify more than one configuration on a command line:

```
M204GEN CONDENSE APPEND AUDIT204
```

See Table 2-5 on page 153 for a list of M204GEN configurations. Instead of naming each configuration, you can specify *ALL* to generate all configurations of the specified type (MOD or SEG). If you do not include a configuration, no action is taken.

- *Ctl <cntrl>* specifies the name of the control file used by M204GEN. The default is NCMS204.
- *PROD*, the default, creates a production version; *TEST*, optional, creates a test version. As a result of what you specify, M204GEN generates:
 - Modules with names beginning with the prefix *M204* for PRODUCTION

and *T204* for TEST.

- Saved segments, with intermediate modules beginning with the prefix *X204* or *Y204*, and executable segments beginning with the prefix *M204* or *T204*.

```
M204GEN CMS ONLN (TEST
```

You need to specify PROD or TEST only once for a list of configurations when you issue M204GEN.

- *MODule*, the default, causes the named configuration(s) to be generated as modules. *SEGment* causes the configuration(s) to be generated as saved segments. In a z/OS environment, M204GEN generates each saved segment as a DCSS or MSS, depending on the definitions included in the input files used by M204GEN (see “Customizing parameter values (M204USER PARMs)” on page 39).
- *NOSAVE* is the default, if you specify *SEGment*; it is ignored for modules. *NOSAVE* causes M204GEN to generate a module file but bypasses the call to M204SAVE. This is useful for generating modules that you save as MSSs (which is the default for M204CMS, M204IFM1, and M204ONLN). By generating the module files without saving segments, you avoid redefining the saved segments. *SAVE* invokes the M204SAVE command to save the modules as segments for the specified configuration.
- *FUND* specifies to link in support for FORTRAN mathematics functions. *FUNDLE*, the default, specifies to link in support for LE mathematics functions.
- *HRZN*, the default, specifies to link in support for Horizon for TCP. This is required to support IODEV 27 for Horizon-to-Horizon processing and IODEVs 19 and 49 to support Connect ★ SQL and RCL processing. Specify *NOHRZN* to exclude from the module.
- *PQO*, the default, specifies to link in support for Parallel Query Option, which is required to support IODEV 51. Specify *NOPQO* to exclude from the module.

M204GEN configurations

Table 2-5 lists M204GEN configurations. Generate only the modules or segments that your site needs to run Model 204. Minimally, you must specify the CONDENSE, CMS, and either ONLINE or IFAM1 configurations.

The IFAM1 configuration must be generated as a saved segment; the name listed in the table is the saved segment name. The ONLINE, CMS, and M204 configurations can be generated as either:

- Modules with names beginning with the prefix *M204* for PRODUCTION and *T204* for TEST.

- Saved segments, with intermediate modules beginning with the prefix *X204* or *Y204*, and executable segments beginning with the prefix *M204* or *T204*.

Return codes

Return code	Meaning
0	Successful return
1	User error
2	Application or host command error
3	Unexpected (system) error

This message is displayed upon successful generation of a saved segment:

```
HCPNSS440I SAVED SEGMENT <seg_name> SUCCESSFULLY SAVED
```

This message appears after you issue the M204GEN command, with the SAVE option, to generate saved segments. It also appears when you save segments using the M204SAVE command (see next section).

Table 2-5. M204GEN configurations

Name	Module	Description	Module or Segment?	Saved segment
APPEND	M204APND	File definition concatenation utility	M	-
AUDIT204	M204AUDT	AUDIT204 utility	M	-
BATCH2	M204BAT2	BATCH2 utility	M	-
CMS	M204CMS	CMS Interface to Model 204	M	-
	T204CMS		S	M204CMS
	X204CMS			
	Y204CMS			
CONDENSE	M204SQSH	Load map compression utility	M	-
FILEDEF	M204FDEF	Unaccessed disk file definition utility	M	-
IFAM1	X204IFM1	IFAM1 segment	S	M204IFM1
INFO	M204INFO	Obtains terminal information	M	-
LABELDEF	M204LDEF	Cartridge label definition utility	M	-
LIST	M204LIST	M204 Dictionary list module	M	-
MERGEJ	M204MRGE	Journal file merge utility	M	-

Table 2-5. M204GEN configurations (continued)

Name	Module	Description	Module or Segment?	Saved segment
ONLINE	M204ONLN	Model 204 online	M	-
	X204ONLN		S	M204ONLN
USR	M204USR	CMS M204 Terminal Interface	M	-
	X204USR		S	M204USR
UTIL	M204UTIL	Data set manipulation utility	M	-
UTILC	M204UTLC	Checkpoint file dump utility	M	-
UTILJ	M204UTLJ	Journal file dump utility	M	-
VTAM	M204VMVT	VM/VTAM load library	M	-
XFER	M204XFER	Utility to load Model 204 saved segments	M	-
XREF	M204XREF	M204 Dictionary cross-reference	M	-
ZBLDTAB	M204ZBLD	Password table initialization utility	M	-
ZCTLTAB	M204ZCTL	Updates password expiration parameters in CCASTAT	M	-

M204SAVE command

Use the M204SAVE command to save a segment for which a module was previously constructed.

Use the SAVE option to have the M204GEN command call the M204SAVE command.

If you use the default, NOSAVE, for M204GEN, then you must explicitly issue the M204SAVE command for the modules or segments specified. Use the NOSAVE option of M204GEN for those segments that you want to generate as MSSs (which is the default for M204CMS, M204IFM1, and M204ONLN). By generating the module files without saving segments, you avoid redefining the saved segments.

You *must* resave segments with M204SAVE after you apply early warnings.

Storage required for using M204SAVE for saved segments

Syntax `M204SAVE configuration ([PROD | TEST]`

Where • *configuration* specifies the name of the module or saved segment to be generated. See Table 2-5 on page 153 for a list of M204GEN and

M204SAVE configurations. You can specify more than one configuration on a command line; for example:

```
M204SAVE ONLINE CMS USR IFAM1
```

Instead of naming each configuration, you can specify *ALL*, to generate all the segments.

- *PROD* is the default and creates a production version; *TEST* is optional and creates a test version. As a result of what you specify, M204SAVE generates:
 - Modules with names beginning with the prefix *M204* for PRODUCTION and *T204* for TEST.
 - For saved segments, intermediate modules beginning with the prefix *X204* or *Y204*, and executable segments beginning with the prefix *M204*.

You must specify PROD or TEST only once for a list of configurations when you issue M204SAVE.

Using M204SAVE for MSS generation

The following example uses M204GEN to generate the three modules which comprise the M204CMS, M204ONLN, and M204IFM1 member saved segments (MSSs). First, the module files are generated and then M204SAVE is used to define and save the segments:

```
M204GEN CMS IFAM1 ONLINE (SEGMENT NOSAVE
M204SAVE CMS
```

Note that M204SAVE specifies only CMS. Because these three are defined as MSSs, all three segments are defined; you can specify only IFAM1 or only ONLINE. If you specify all three in the M204SAVE command, each MSS is saved multiple times.

Return codes

Return Code	Meaning
0	Successful return
1	User error
2	Application or host command error
3	Unexpected (system) error

Maintaining the MAINT204 machine

Once you have defined the MAINT204 machine and installed Model 204 onto that machine, you can access MAINT204 minidisks in read-only or read/write mode using the MAINTACC command.

MAINTACC command

Run MAINTACC only from the MAINT204 machine. MAINT204 must have the required 193 and 194 minidisks defined as described in the sample directory entry on “Defining the MAINT204 directory entry” on page 9.

Syntax MAINTACC{RR | RW | ?}

Where

- *RR* (default) indicates read-only access to the disks. Each disk is accessed with file mode extensions equal to the primary (first) file mode:

193 C/C

194 D/D

195 E/E

- *RW* indicates read/write access to the disks.

193 C

194 D

195 E

- ? displays help information.

MAINT204 links to database minidisks

The Model 204 maintenance virtual machine (MAINT204) might require access to the service virtual machine’s database minidisk(s) to perform data set maintenance functions. For convenience, the address used by MAINT204 to

access the linked service machine's database minidisk(s) should be the same as the address defined in the service virtual machine's directory. For example,

```
LINK MODEL204 . . . 210 210.
```

If you are installing Model 204 in an environment with multiple versions of CMS, generate a separate copy of the CMS Interface for each CMS version.

C

Decision Making

In this appendix

- Overview
- Choosing CMS-format or variable-format files
- Determining whether to generate modules or segments
- Saved segments
- Determining whether to use default segment definitions

Overview

As you plan your Model 204 installation, you are faced with fundamental decisions about how to implement Model 204. You must determine:

- File format
- Whether to generate your configuration as a segment or a module
- If you generate segment, whether to use default segment definitions

This appendix supplies information about choosing file formats and configurations.

Choosing CMS-format or variable-format files

This section provides background information about choosing CMS-format and variable-format files.

Two file formats are available under z/VM:

- CMS-format
- Variable-format

For each type of format, consider requirements for:

- Performance issues
- File sharing
- File allocation

CMS-format files

Performance issues

Files on CMS disks are allocated using a scheme that places the blocks associated with a single file in a random pattern across the volume. The blocks are fixed length, although the CMS file system maps records of any size onto the underlying physical structure. In particular, logical blocks of data can span multiple physical blocks.

CMS file systems use the DIAGNOSE instruction interface that is supported by the z/VM Control Program (CP) to perform DASD Input/Output operations. Through this interface, I/O operations perform synchronously with virtual machine execution. That is, the virtual machine is suspended until the I/O operation completes.

CMS uses file allocation techniques that can extend the duration of I/O operations because of the additional arm motion and rotational delay required to read multiple disjointed physical blocks. The fixed-block format of its disks can cause some I/O operations to require extra block transfers between the disk and main storage because of the spanning of physical blocks by logical records. This also tends to increase I/O service times. The synchronous I/O execution interface eliminates the possibility of overlap between I/O and processing when CMS-format disks are used.

Note: It is **not** recommended to use CMS format files for any applications in which performance is critical.

File sharing

Multiple virtual or real machines cannot share files in read/write mode on CMS-format disks. The file allocation techniques and the lack of support in CMS for access serialization prevent effective read/write file sharing.

File allocation

Files on CMS disks do not require preallocation. Instead, files increase dynamically as data is added and are restricted in size only by the space

available on the minidisk where they reside or by the XTENT option on the FILEDEF command.

CMS has no explicit file creation function. A new file is created automatically the first time it is referenced. File allocation information is recorded in storage and is highly dynamic in nature. When all the files on a disk are closed, the allocation information is transferred to disk. The master file directory (MFD) records which files exist on a particular CMS volume.

Variable-format files

Performance issues

Variable-format disks permit variable length physical blocks to be written and are not allocated randomly. Instead, allocation is predetermined through a separate process that reserves areas on the volume for particular files. The CMS Interface for Model 204 supports asynchronous I/O operations through SSCH-level logic and associated interrupt handling facilities. As a result, you can acquire significant overlap between I/O and processing when you use variable-format disks.

Note: Files that experience high levels of activity, and thus can benefit significantly from overlapped processing, should be allocated on variable-format disks.

File sharing

Files on variable-format disks can be shared by several virtual machines through the use of virtual Reserve/Release facilities. Reserve/Release permits access to a volume restricted to a particular (real or virtual) access path. Since allocations are static in nature after a file has been located, it can be read and written without further reference to the allocation information, except when secondary allocation functions are required.

Read/write sharing of files on variable-format disks by multiple Model 204 applications is limited to database files (including CCAGRP) and the Model 204 password table (CCASTAT).

File allocation

Files on variable-format disks must be preallocated. A primary allocation must be provided, and secondary extents can be specified to permit limited extension of the file. The file allocation information is recorded on a disk in the Volume Table of Contents (VTOC).

When a file is allocated with the M204UTIL utility that specifies the Secondary option, the secondary extents are not allocated under z/VM. For z/OS and z/VSE operating system compatibility, secondary extents that are allocated by a program executing in that operating system are supported when sharing full

volumes. For more information about file allocation, refer to the *Model 204 File Manager's Guide*.

Determining whether to generate modules or segments

Table C-1 indicates for each M204GEN configuration whether that configuration can be generated as a module, a segment, or both.

The next section provides some background information about saved segments that may be useful in determining whether you want to use saved segments, and if so (in a z/VM environment), what type of saved segment to use.

Summary of configurations

For each M204 command configuration, Table C-1 shows whether the configuration is generated as a module, a segment, or both. Table C-1 also shows which default type of saved segment (DCSS or SSP/MSS) is used in z/VM operating environments.

Table C-1. M204GEN configurations: modules or segments

Configuration name	Module or segment
APPEND	Module
AUDIT204	Module
BATCH2	Module
CMS	Module or segment In z/VM, default is an MSS. Changing it to a DCSS is not recommended.
CONDENSE	Module
DEFINE	Module Generated automatically if any configuration requires it.
IFAM1	Segment In z/VM, default is an MSS. Changing it to a DCSS is not recommended.
INFO	Module
LABEL	Module
LIST	Module
MERGEJ	Module
ONLINE	Module or segment In z/OS, default is an MSS Changing it to a DCSS is not recommended.

Table C-1. M204GEN configurations: modules or segments (continued)

Configuration name	Module or segment
SECURITY	Module
TRACE	Module
USR	Module or segment In z/OS, default is an MSS. Changing it to a DCSS is not recommended.
UTIL	Module
UTILC	Module
UTILJ	Module
VTAM	Module (actually generated as a loadlib)
XFER	Module Generated automatically if any configuration requires it.
XREF	Module
ZBLDTAB	Module
ZCTLTAB	Module

Saved segments

When to use saved segments

You can install some sections of Model 204 as z/VM saved segments instead of, or in addition to (except for IFAM1), as modules.

Saved segments provide advantages in the following cases:

- In a z/VM environment where many virtual machines are running single copies of Model 204, it is appropriate for a program to be shared among all users.
- In multiuser Model 204 environments, it is necessary to load user-written programs along with Model 204. The saved segment capabilities allow separate storage areas for Model 204 and user-written programs.

TPROCESS

If you want to use CMS TPROCESS (Process-to-Process) communications, you must execute the terminal interface, M204USR, in a saved segment or a nucleus extension; use of a module is not supported for the TPROCESS feature. Refer to the *Model 204 System Manager's Guide* for information about using the TPROCESS feature.

Saved segments

The z/VM operating system includes the following types of saved segments:

- Discontiguous saved segment (DCSS) occupies one or more architected segments. A DCSS begins and ends on one megabyte boundaries and is accessed by name.
- Saved segment space (SSP) is composed of up to 64 member saved segments (MSSs). SSPs occupy virtual storage in one megabyte increments, beginning and ending on a megabyte boundary. Access to a SSP provides access to all of the MSSs in that SSP.
- Member saved segment (MSS) is a saved segment that is a member of an SSP. MSSs occupy storage in 4K increments and must begin and end on page boundaries. An MSS can be accessed by its own name or by the SSP name.

When to use SSPs

Small programs defined as a DCSS can waste storage space. To save storage space, pack programs into SSPs.

Saved segments: architectural differences

The architectural differences between saved segments are illustrated in Figure 0-1. Two different configurations of Model 204 are shown:

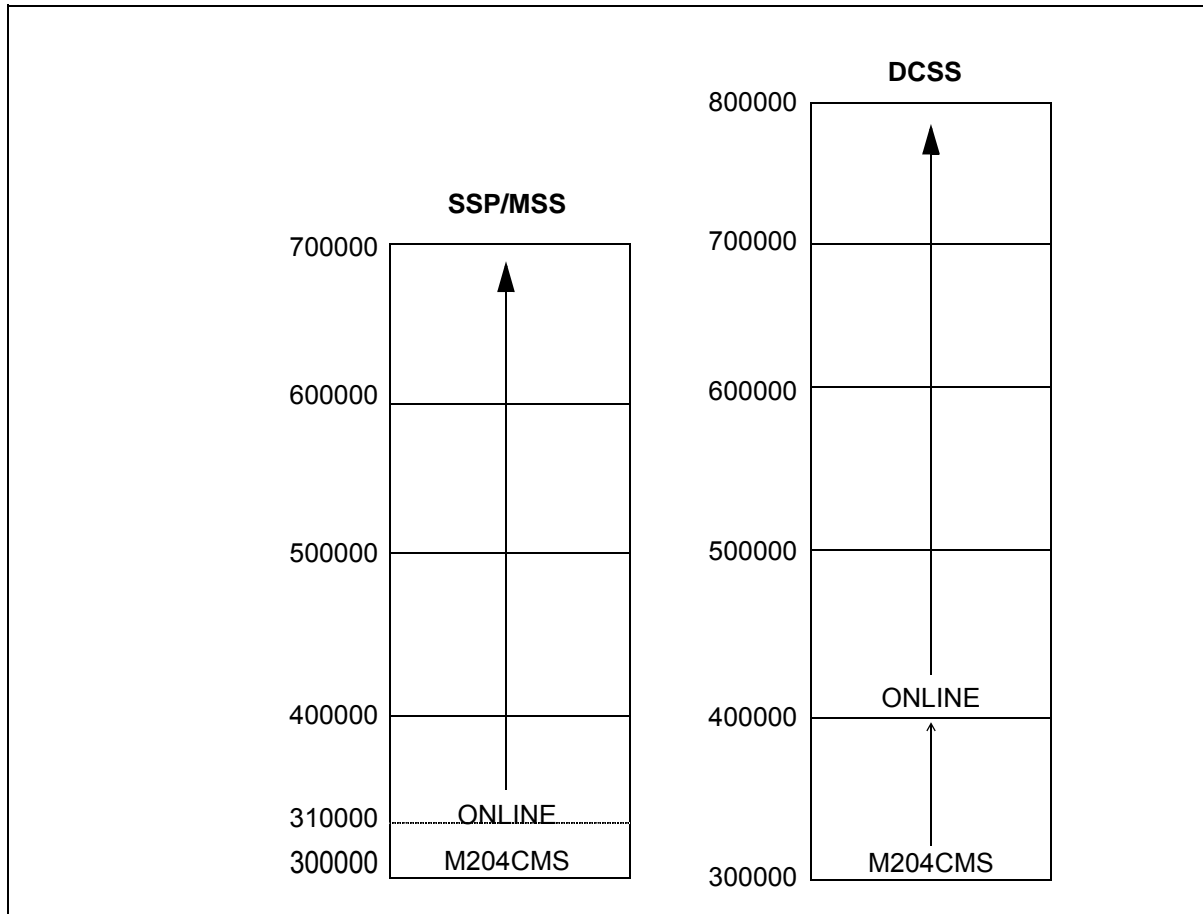
- MSS
- DCSS

The numbers to the left of each configuration are hexadecimal addresses.

Running M204CMS, and M204ONLN as DCSSs requires a total of five MB (one for M204CMS, and four for M204ONLN). Therefore, these saved segments are defined as MSSs in Model 204 and require only four megabytes, as shown in Figure 0-1.

The SSP names are ONLINE and IFAM1.

Figure 0-1. Saved segments



Determining whether to use default segment definitions

Once you decide whether you are going to generate a configuration as a module or as a saved segment, you need to determine whether you want to use the default saved segment(s) definitions or redefine them.

The default definitions, as described below, are stored in the M204USER PARS file.

See Chapter 3 for a description of how to change saved segment definitions.

Saved segment definitions

Saved segment definitions include attributes listed in Table C-2.

Table C-2. Attributes of saved segment definitions

Attribute	Specifies...
Origin	Load point of the saved segment; the virtual storage address where the segment is loaded.
Key	Storage key for the reentrant section of the segment. This should always be zero.
Rent	Page range of the reentrant section of the segment. Specify the page numbers in hexadecimal.
Nonrent	Page range of the non-reentrant section of the segment. Specify the page numbers in hexadecimal.
Rstd	Whether use of the saved segment is restricted. Virtual machines require the NAMESAVE directory control statement to use a restricted saved segment.
Spname	SPACE names that include this saved segment. Entering a value indicates an MSS.

Default saved segment definitions

You can review the default saved segment definitions by examining the M204USER PARMS file, which is on the MAINT204 193 disk. The first major section of the M204USER PARMS begins with the following line:

```
M204SSEG.XA
```

The same default values apply to PRODUCTION and TEST versions of the saved segments for these configurations.

The following example shows the saved segment definitions, in the format they appear for the ONLINE configuration for z/VM.

```
.
.
.
M204SSEG.XA
M204ONLN:
  Origin   = 310000
  Key      = 0
  Rent     = 310-5FF
  Nonrent  = 600-6FF
  Rstd     = N
```

```
Spname = ONLINE  
END.M204ONLN:
```

```
.  
.   
.
```


D

Changing Saved Segment Definitions

In this appendix

- Overview
- Guidelines for changing saved segment definitions
- Loadlists
- z/VM examples

Overview

This appendix describes how to change the definitions for Model 204 saved segments. If your site wants to use the default saved segment definitions provided by Rocket, then you do not need to read this appendix.

Four configurations can be generated with the M204GEN (and possibly the M204SAVE) command as saved segments. Only IFAM1 must be generated as a saved segment.:

- CMS
- IFAM1
- M204
- ONLINE

This appendix also discusses:

- Member saved segments (MSSs)
- Discontiguous saved segments (DCSSs)

- Saved segment spaces (SSPs)

Guidelines for changing saved segment definitions

This section provides important guidelines to bear in mind if you change the saved segment definitions from those supplied by Rocket.

If you are upgrading from an earlier Model 204 release

If you are upgrading from an earlier Model 204 release and your site has redefined saved segments, then you can use the same definitions with this version, provided that those definitions are consistent with the guidelines presented in this appendix.

You must still modify the M204USER PARMS file, and you might need to modify the loadlist(s) to make those definitions available for use by the M204GEN command.

Review M204USER PARMS carefully for the default definitions.

Guidelines for changing the M204USER PARMS file

If you redefine saved segments, you must update the M204USER PARMS file. This is a logical starting point for redefining saved segments; you can base any changes to the loadlist on the information you enter in the M204USER PARMS file.

Note the following requirements when modifying the M204USER PARMS file:

- Page range for reentrant (Rent) must always have a range specified. If there is no non-reentrant section (Nonrent), you must specify a 0 (zero).
- If you define a saved segment as an MSS, you must specify a space name (Spname) (z/OS only).

Where to load saved segments

In most z/VM environments you can load a saved segment either within or outside a virtual machine.

In z/VM environments, it is recommended that you load the saved segment for the CMS configuration at an address lower than you use for loading the ONLINE and IFAM configuration saved segments.

ONLINE and IFAM1 saved segments can have the same virtual address, because they are not used concurrently in the same machine. Similarly, CMS and Model 204 saved segments can share the same virtual address.

Do not overlap the ONLINE and IFAM1 configuration saved segments with CMS saved segments.

TEST and PROD saved segments

When you are changing saved segment definitions, it is recommended that you generate your new definitions as TEST saved segments when you issue the M204GEN command. You can then generate the modified definitions as PROD (production) saved segments once you have tested them.

Test and production versions of a saved segment can use the same definition, as specified in M204USER PARMS and possibly the loadlists.

Reentrant and non-reentrant code

Saved segments allow code to be shared among several users, yet can also allow some code to be stored separately from the shared code. Saved segments are composed of reentrant and possibly non-reentrant architected segments.

Reentrant architected segments can be shared by many users, but cannot be updated by users. They are accessed in read only (R/O) mode.

Non-reentrant architected segments cannot be shared by many users. They are accessed in read/write (R/W) mode.

Reentrant and non-reentrant code can exist in one MSS or DCSS, but each type of code must reside in separate architected segments.

Place reentrant code in shared (R/O) architected segments, and non-reentrant code in exclusive (R/W) architected segments.

The reentrant and non-reentrant csects are positioned in the default saved segment definitions to minimize the amount of storage required. You must preserve the same relative proportion and position for reentrant and non-reentrant code in modified saved segment definitions as are specified in the default definitions.

MSSs and SSPs

MSSs and SSPs are available only in a z/VM environment.

Small programs defined as DCSSs might waste much storage; generally, pack them into SSPs.

It is strongly recommended that you do not redefine saved segments that by default are defined as MSS to become DCSSs. The CMS, ONLINE, and IFAM1 configuration saved segments are defined as MSSs (for z/VM). The USR configuration saved segment is defined as a DCSS, because it is not packed with an SSP with any other Model 204 segments. You can define USR as an MSS packed with other saved segments in use at your site, for example, language processors.

Default SSPs

The default SSPs are:

- ONLINE (made up of the CMS and ONLINE configuration MSSs)
- IFAM1 (made up of the CMS and IFAM1 configuration MSSs)

Guidelines

You can define up to 64 MSSs in one SSP.

SSPs occupy virtual storage in 1-MB increments, beginning and ending on a megabyte boundary.

None of the names used for Model 204 saved segments (see Table C-2 on page 166) can be used as SSP names.

MSSs occupy storage in 4K increments.

MSSs must begin and end on page boundaries.

Loadlists

A *loadlist* is an EXEC file that contains a list of text files that comprise a load module. Each configuration has one or more loadlists associated with it.

If you redefine a saved segment, but it starts on an architected segment boundary (1 megabyte for z/VM), then you do *not* need to change the loadlist.

However, if you redefine a saved segment so that it does not start on a saved segment boundary, you must change the %SLC statement, as the following describes.

Location

The loadlists are loaded onto the MAINT204 (maintenance) machine's 194 disk.

Saved segment loadlist names

Table D-1 indicates the names of the loadlists for saved segments.

Table D-1. Saved Segment Loadlists

M204GEN configuration	z/OS
ONLINE	LSXAONLN
USR	LOADM204
CMS	LOADCMS
IFAM1	LSXAIFM1

Loadlist format

A loadlist contains a list of entries, one for each object module to include in a load module. For example:

```
&1 &2 M204ONLN
```

Each z/OS loadlist contains one %SLC statement. For example:

```
%SLC 1F0000 M204ONLN
```

%SLC statement syntax

The format for the %SLC statement is:

```
%SLC offset entry
```

where:

- *offset* is the amount, in hexadecimal, that is required to position the saved segment at an architected segment boundary.

If you define the starting address for the saved segment to be at an architected segment boundary, then you do not need to change the offset in the %SLC statement.

If you need to change the offset, use the following formula to calculate the new offset if you are loading the reentrant code at a higher address than the default starting:

$$\text{newoffset} = \text{defoffset} + (\text{newasb} - \text{defentry})$$

where:

- *newoffset* is the value for the offset parameter in the %SLC statement you are editing.
- *defoffset* is the default offset value supplied in the loadlist.
- *newasb* is the new architected segment boundary at which the saved segment will be loaded.
- *defentry* is the value of the entry parameter in the default %SLC statement.
- *entry* is the name of the entry point to which the %SLC statement applies. The entry is a symbol or name with which the CMS loader can associate a virtual storage address. The location counter is set to the sum of the offset and the address associated with the entry.

For example:

```
%SLC 1F0000 M204ONLN
```

VMFLOAD command not supported for loadlists

You cannot use the CMS command VMFLOAD. Rocket provides its own load utility, eliminating the need to use the VMFLOAD command for installing Model 204.

FLOD exits

You must add a line to the appropriate ONLINE loadlist for each FLOD exit your site uses. Refer to “Preparing FLOD exits” on page 38 and to the *Model 204 File Manager’s Guide*.

z/VM examples

The default saved segment definitions can be changed in a variety of ways, ranging from simple to very complicated. For Model 204, the following types of changes might be needed:

- You can move origins to a different segment boundary or a nonsegment boundary.
- You can make M204USR an MSS.

Changing the origin of M204USR

To change the origin of M204USR, edit M204USER PARMS, making sure that the origin matches the beginning page number for reentrant range (M204USR is fully reentrant).

Changing the origin of M204CMS/M204IFM1 and M204CMS/M204ONLN

To move the origin of M204CMS to a segment boundary, edit M204USER PARMS:

1. Change the M204CMS origin for the reentrant page range section (M204CMS is fully reentrant).
For example, change 300000 to 400000 and 300-30F to 400-40F.
2. Change M204IFM1 origins and page ranges, starting 64K higher than M204CMS.
For example, change:
 - 310000 to 410000, 310-5FF to 410-6FF for reentrant
 - 600-6FF to 700-7FF for non-reentrant
3. Change M204ONLN origins and page ranges, starting 64K higher than M204CMS.
For example, change:
 - 310000 to 410000, 310-5FF to 410-6FF for reentrant

- 600-6FF to 700-7FF for non-reentrant

You do not need to change loadlists.

Making M204USR an MSS and packing it with other segments

You can pack M204USR with other segments that your site uses, such as high level language compilers, sort utilities, and so on. To do this:

1. Establish the origin based on the other segments with which it will be packed. Specify the origin in M204USER PARMs and make the corresponding change to the reentrant page range.
2. Specify the space names with which M204USR will be packed. See M204USER PARMs editing instructions.
3. Follow the instructions for *defining* other types of saved segments. See your IBM documentation for more information about defining saved segments.
4. Generate and save M204USR with the M204GEN command.
5. Or, if the module has already been generated, and you do not need to regenerate it, use M204SAVE to resave it.
6. Follow the instructions for *saving* the other segments. See the instructions for saving the MSSs of any other program products you are packing with M204USR.

Index

Symbols

\$functions 35 to 38
%SLC statement
 accommodating FLOD exits 39
 syntax 173

A

accounting exits (ACCTEXIT) 31
ALLOCATE function 51, 146 to 147
allocating
 demonstration database files 52
 Model 204 files 51
 password data set 56
 testing allocations 53
allocation exits (ALLOEXIT) 34
ALLOW statement for IUCV 17
ALTIODEV parameter 64
AMODE settings
 accounting exits 32
 allocation exits 35
 invoking MVS services that receive control in
 24-bit mode 36
 security exits 34
APPL statements and VTAM support 47
application database files 11
applying early warnings 57
architecture
 BATCH2 64
 IFDIAL 66
 multiuser 60
 single-user 63
 standalone utilities 66
archive
 identifier 3
 unpacking 27
assembling user-written exits 30
audit trail 68
AUDIT204
 command 68
 utility 136
Autofix
 applying early warnings 57

Autofix files 3

B

backing up
 Dictionary files 96, 98
 M204PROC file 105
backup facilities 51
basic mapping support (BASM) 31
batch facilities 64 to 65
BATCH2 configuration
 object modules 127
 running 64 to 65
BYPASS operand, ONLINE command 59

C

calling conventions
 accounting exits 32
 security exits 33
CCAAUDIT system file 14, 68
CCACAT file
 field attribute change 119
CCACATREPT 121
CCAGRP system file 14, 56
CCAIN files
 D204FDEF 109
 D204FEDF 106
 DDFIMCMD 113
 DDGENSET 112
 DDPROCP 85
 Dictionary installation 72
 LOGIN 76
 LOGINSM 76
 modifying 20
 printing 73
 reorganizing 109
 SAMPONLN 54
 SAMPRECV 54
CCAIN system file 15
CCAJRNL system file 15, 68
CCAPRINT system file 15, 67
CCASERVER system file 15
CCASNAP system file 15

CCASTAT system file	15, 56
CCASYS system file	
sizing	89
CCATEMP system file	
format	15
CCATSF	121
CDTB module	35 to 38
CHKPOINT system file	15
CMS	
assembling user-written exits	30
machine and IPL statement	14
minidisk 105	10
nucleus, storage	13
CMS Interface	1, 140
CMS-format	
disks	20
FILEDEF statements	74
files	159, 161
commands	
IFAM1	65
LABELDEF	144
M204APND	139
M204CMS	140
M204CRYP	140
M204FDEF	143
M204GEN	151
M204LDEF	144
M204SAVE	154
M204SCTY	139
M204USER	64
M204UTIL	21, 52, 53, 145, 148
M204XFER	144
MAINTACC	156
ONLINE	50
VMDUMP	68
VMFLOAD	174
VT204	139
z/VM utility	139
ZAP	57
ZAPTEXT	57
concatenating	
files	139
M204APNDT	139
magnetic tape volumes	144
configuration defaults	39 to 45
Connect★	
free threads	4
installation	118
support, installing	121
console messages	68
CPUID operand	13
CPUIDZAP ZAP file	27
Cross-Reference facility	18 to 20, 92
CUST module	35 to ??

CUST source code	
date/time stamp field name	37

D

D204 BACKUP	104
D204 EXEC	
DDGENSET	111
DDPROCP	26, 85
DDTINIT	114
defined	73
INSTALL	94
modifying LOGIN and LOGINSM CCAIN files	
76	
REORGANIZE	108
required storage	73
RESTORE	106
D204 RESTORE	100
D204FDEF EXEC	
backing up Dictionary files	105
customizing FILEDEF statements	79
populating Dictionary from Model 204 files	110
relationship to D204 EXEC	73
reorganizing files	109
restoring Dictionary files	106
setting up D204 DDGEN based on Dictionary	
files	112
updating files using delayed processing .	113
D204RPT file	
contents	77
reorganizing	108
sizing	91
D204SYS file	
contents	77
sizing for initial installation	93
data sets	
creating on initialized volumes	146
erasing	147
listing information about	149
security exit routine	33
DATALINK file	
contents	77
sizing	91
DATE system value	
formatted for date/time stamp feature	37
date/time stamp	
customizing field name	37
default format	36
implementing the feature	36
see also DTS	
DTS	
see also date/time stamp	
date/time stamp fields	

name definition	36
DCSS (discontiguous saved segments)	164
DDBKPROC EXEC and CCAIN	106
DDFIMCMD file	113
DDGEN	101, 110, 111
DDGENSET	112
DDPROCP	85
DDRSPROC EXEC	107
DDTINIT	114
decrypting	27, 140
defining date/time stamp field name	36
DEMDDL jog	
renaming a schema	124
demonstration database	
allocating files	52
file privileges for multiuser access	61
size	11
DICDIST file	
contents	77
sizing	84
Dictionary	
authorizing access	98
backing up files	96, 98, 104
CCAIN files	72
Cross-Reference facility	18, 92
customizing EXECs	73 to 75
distribution file	84
EXECs	72
file management	113
file sizes	89
FILEDEF statements	73
files allocated and used by EXECs	77
information provided by	71
installation EXECs	72
passwords	76
recovery	97
reorganizing files	107
required privileges for D204 functions and EX- ECs	75
required storage	73
restoring files	100, 106
standard facilities	72
upgrading standard facilities	100
when to modify EXECs	104
directories	
GCS	16
MAINT204	9
user	17
discontiguous saved segments (DCSS)	164
disk formats for system files	14
distributed processing	48
Distributed/204. see Parallel Query Option	
distribution archive	
identifier	3

DSORG settings	15
DTS field	
and FOPT parameter	36
customizing for your site	38
defining	36
DTSFN value	
where established	37
dumps	
automatic	68
manual	68
dynamic allocation	34

E

early warnings	57
ERASE function	147
EXECs	
required storage	73
EXECs	
customizing	73 to 75
D204	73
D204 INSTALL	94
D204FDEF	79, 105
DDBKPROC	106
DDRSPROC	107
Dictionary	72
files allocated and used by EXECs	77
M204DEF	54, 58
M204FDEF	106
MAINPROF	28
modifying	54
modifying on MAINT204	9
ONLINE	96
prefixes	54
printing	73
relationship to M204USER PARMS file	41
reorganizing	109
required privileges	75
REXX	54
SAMPDEF	54
SAMPREST	54
SERVPROF	29
VMFASM and VMFHASM	30, 38
ZBLDTAB	56
exits	
accounting (ACCTEXIT)	31
allocation (ALLOEXIT)	34
FLOD	38
security (SCTYEXIT)	33
sort (OI15CMS and OI35CMS)	30

F

file enqueueing	13
FILEDEF statements	
CMS-format	74
for D204FDEF	79
OS-format	75
syntax	74 to 75
FILELOAD	16
Files	
Autofix	3
files	
allocated and used by Dictionary EXECs .	77
CCAAUDIT	68
CCAGRP	56
CCAJRNL	68
CCAPRINT	67
CCASTAT	56
CMS-format	159, 161
concatenating	139
D204RPT	91, 108
D204SYS	93
DATALINK	91
delayed processing	113
DICTDIST	84
formats	14, 159
loading	16
M204DCTL	92
METADATA	90, 108
Model 204 system files	14 to 16
OUTDDG1	113
recovery through ONLINE EXEC	50
restoring	107
variable-format	159, 161
variable-format disks	143
work files	52
FLOD exits	
file formats	16
modifying loadlists	174
OI15CMS and OI35CMS	39
FOPT parameter	
and DTS field definition	36
FORTRAN	
mathematical routines	7
FORTRAN textlib	30
free threads for software	
Connect★	4
Horizon	4
PQO	4
full-screen device	64
FUNU module	35 to 38

G

GCS	
directory entry	16
server	47

H

Horizon	
free threads	4
Horizon interface installation	48
Horizon/LE interface installation	48

I

IBM	
zSeries and other System z, required hardware	
.....	2
IFAM1	65
IFAM2	66
IFDIAL	66
INITIAL function	148
initializing	
disks	20, 148
password table	137
installation	
checklist	24
testing	60 to 66
interactive single-user access to Model 204 Online	
62 to	64
interface, Model 204 and CMS	140
IPL statement, MODEL204 machine	14
IUCV statement	
GCS service machine	17
MODEL204 machine	14

J

journal	
AUDIT 204	136
files	68, 136
MERGEJ	136
merging	136
printing	136

K

KEY attributes	
in CCACAT file	119
saved segments	43

L

LABEL function	149
LABELDEF command	144
LENQTBL parameter	97
LIBUFF parameter	124
line-at-a-time device	64
LINK statements	
GCS service machine	17
MAINT204 machine	10
MODEL204 machine	14
user machines	18
LIST function	
listing data set information	149
LLGTBL parameter	97
loading files	16
loadlists	
%SLC statement	173
format	173
location	172
modifying for FLOD exits	174
ONLINE	38, 174
saved segment names	172
LOADONLN loadlist	39
LOGADD parameter	114
LOGIN CCAIN file	76
LOGINSM CCAIN file	76
LOGOUT records	32
LOUTPB parameter	97
LPDLST parameter	97, 124
LQTBL parameter	124
LRECL settings	15
LSXAONLN loadlist	39
LTTBL parameter	124

M

M204 Operations section	
M204USER PARMS file	43
M204APND command	139
M204CCA PARMS file	40
M204CMS command	140
M204CRYP command	
report	141
return codes and messages	142
M204DCTL file	
contents	77
sizing for initial installation	92
M204DEF EXEC	54
M204FDEF command	143
M204GEN command	
configurations	152, 162
defaults	45

description	151
input files	10
input to	151
options	151
return codes	153
syntax	151
M204INST file	78, 96
M204INST files	
pages required	83
M204LDEF command	144
M204ONLN	29
M204PROC file	
allocating and creating	84
backing up	105
contents	78
increasing for additional products	85
restoring	107
size	84
M204SAVE command	
description	154
MSS generation	155
resaving segments	46
return codes	156
storage required	154
M204SCTY command	139
M204SSEG.XA section	
M204USER PARMS file	43
M204TEMP file	
contents	78
reinitializing	114
sizing	92
M204USER command	64
M204USER PARMS file	
attributes of saved segment definitions ...	43
defaults for the M204 EXEC	43
guidelines	170
modification examples	44 to 45
sample file	41 to 42
TXTLIBs	45
M204UTIL command	
ALLOCATE function	51, 52, 146 to 147
ERASE function	147
functions	145
INITIAL function	148
LABEL function	149
LIST function	53, 149
shared DASD	146
syntax	145
M204XFER command	144
MACHINE mode statement	
MODEL204 machine	13
MAINPROF EXEC	28
MAINT204 machine	9, 156
profile	28

MAINTACC command	156
maintenance changes (early warnings)	57
maintenance virtual machine	
customizing the MAINT204 profile	28
defining	9
MAXCONN operand	13
MAXIMUM_SCALE field	122
MDISK statements	
MAINT204 machine	10
MODEL204 machine	12, 14
MERGEJ utility	136
messages, console	68
METADATA file	
contents	78
reorganizing	108
sizing	90
minidisks	
accessing with appropriate file modes	51
address and label in INITIAL function	148
addressing for MODEL204 machine	12
contents listing	24
linking 193 disk	76
MAINT204 machine	10
MODEL204 machine	14
MINIMUM_SCALE field	122
Model 204 and z/VSE	
supported z/VSE and ESA versions	2
MODEL204 machine	
accessing from MAINT204 machine	10
bringing up the Online	58
description	11
directory definition	12
system identification	13
MODEL204 profile	28
modules	
bringing up the online	59
generating	45
Model 204 object	127
ONLINE	38
running recovery	59
site-specific	35
when to use	162
MSGCTL parameter	68
MSGU module	35 to 38
MSS (member saved segments)	
defined	164
guidelines	171
using M204SAVE	155
multiple users	
access to Model 204 Online	60
and the MAXCONN operand	13
environment	50

N

NCMS204 CNTRL file	38
NDCBS parameter	97
NDIR parameter	97
NFILES parameter	97
NOARG parameter and user-written \$functions	36
non-reentrant code	171
nonrent (non-reentrant) attribute, saved segments	43
NORQS parameter	97
NOSAVE option, M204GEN command	46

O

object modules	127
OI15CMS (sort exit)	30, 39
OI35CMS (sort exit)	30, 39
Online	
accessing in BATCH2 mode	64 to 65
accessing in interactive single-user mode	62 to 64
accessing in multiuser mode	60
accessing in single-user IFDIAL mode	66
bringing down	59
bringing up	58
interactive single-user mode	62 to 64
loadlist	38
multiuser	60 to 62
ONLINE EXEC	
description	50
functions	96
operating modes, Model 204	50
OPTION statement	
GCS service machine	17
MODEL204 machine	13
optional features	27
Ordered Index updates	30
origin attribute, saved segments	43
OUTDDG1 file	113
OUTDDGN file contents	78
OUTFILE reports file	78

P

Parallel Query Option	48
Parallel Query Option (PQO)	
free thread	4
parameter files	39 to 45
passwords	
creating data set	56
Dictionary	76
Model 204	137

security	137
ZBLDTAB	137
ZCTLTAB	137
PREV section, M204USER PARMS file	43
printing EXECs and CCAIN files	73
privileges for Dictionary EXECs	75
problem determination tools	67
PROD saved segments	171
PROFILE EXECs for MAINT204 and MODEL204 machines	28 to 29

R

RECFM settings	15
recovery	59, 97
reentrant code	171
RENAME function	150
Renaming a schema	
DEMDDL job	124
rent (reentrant) attribute, saved segments	43
requirements for Model 204	
hardware	2
IBM zSeries and other System z	2
RESTART	15
restoring Dictionary files	106
rstd (restricted) attributes, saved segments	43

S

SAMPDEF EXEC	54
SAMPONLN CCAIN files	54
SAMPRECV CCAIN files	54
SAMPREST EXEC	54
saved segments	
attributes defined by M204USER PARMS file	
43	
BETA versions	44
bringing up the Online	59
changing	174
changing definitions	169
DCSS	164
default definitions	165
definition attributes	166
description	163
generating	45
guidelines	170
KEY attributes	43
loading	170
loadlist names	172
M204XFER command	144
MSS	171
non-reentrant code	171
reentrant code	171

reserving address space	29
running recovery	59
space	164
SSP	164, 171
TEST versions	171
valid configurations	169
schema name, and DDLUTIL	124
security exit routine	
SCTYEXIT	33
security exits (SCTYEXIT)	33
SEGMENT RESERVE statement	29
service virtual machine	11 to 14
accessing from maintenance machine	10
bringing up the Online	58
customizing the MODEL204 profile	28 to 29
SERVPROF EXEC	29
SERVSIZE parameter	97, 124
shared code	171
since-last records	32
single-user access	
IFDIAL mode	66
interactive mode	62 to 64
site-defined Dictionary/204 extensions	
including in installation	108
site-specific values	27
SMFLORN and SMFSLRN parameters	32
SNA Communications Server (formerly VTAM)	
unencrypted	3
SNAPCTL option	68
Software requirements	
SQL catalog subsystems	118
sort exits	30, 39
SORT utility	30
SORT5 system file	15
SPCORE parameter	97
spname (space name) attribute, saved segments	
43	
SQL catalog	119
SQL catalog subsystems	
software dependencies	118
SQL installation	118
SQL support	
installing SQL subsystems	121
installing the SQL catalog	119
SQLBUFSZ parameter	124
SQLIQBSZ parameter	124
SQLRINST job	122
SSP (saved segment spaces)	
defined	164
guidelines	171
standalone utilities	66
statistics records	31
storage requirements for Dictionary EXECs	73
supported releases	2

SVC 99 request block	34 to 35
SYSOPT parameter	97
system files	14 to 16
system identification	13
System Management Facility (SMF)	13

T

T204ONLN	29
TAPEn system files	15
TERMOPT parameter	48
TEST saved segments	171
testing the installation	60 to 66
TIME system value	
formatted for date/time stamp feature	37
troubleshooting	67
TXTLIBs	
changing defaults	45
specifying	29

U

user directories	17
User Language Math Pack \$functions	8
user machines	18
USER statement	
MAINT204 machine	9
MODEL204 machine	12
user-written \$functions	35 to 38
user-written exits	30 to 35
user-written FLOD exits	38
utilities	
AUDIT204	136
M204CMS	140
M204UTIL	150
MERGEJ	136
MODEL 204	66
standalone	66
ZBLDTAB	137
ZCTLTAB	137

V

Variable-format disks	
defining files	143
M204FDEF	143
variable-format disks	
erasing	146
labeling	149
M204UTIL INITIAL	21
M204UTIL LABEL	149
renaming	150
variable-format files	159, 161

virtual machines	
customizing profiles	28 to 29
GCS service machine	16
maintenance (MAINT204)	9
service (MODEL204)	11 to 14
user	18
VM/ESA operating system. See z/VM	7
VMDUMP command	68
VMFASM and VMFASM EXECs	
assembling CDTB, FUNU, and MSGU modules	
.....	38
VMFLOAD command	174
VOLSER identifier on distribution archive	3
Volume Tables of Contents	148
VT204 command	139
VTAM interface	
GCS directory entry	16
installing	16
network definition	47
VTAM Interface installation	47
VTAMGCS group	17
VTOC capacities	148

X

XREF service machine	18 to 20
----------------------------	----------

Z

z/OS-format	
disk and password data set allocation	56
FILEDEF statements	75
z/VM	
31-bit mode and user-written \$functions ..	35
utility commands	139
z/VM operating system	7
z/VSE	
supported releases for Model 204	2
ZAP command	57
ZAPTEXT command	57
ZBLDTAB EXEC	56, 137
ZCTLTAB	137