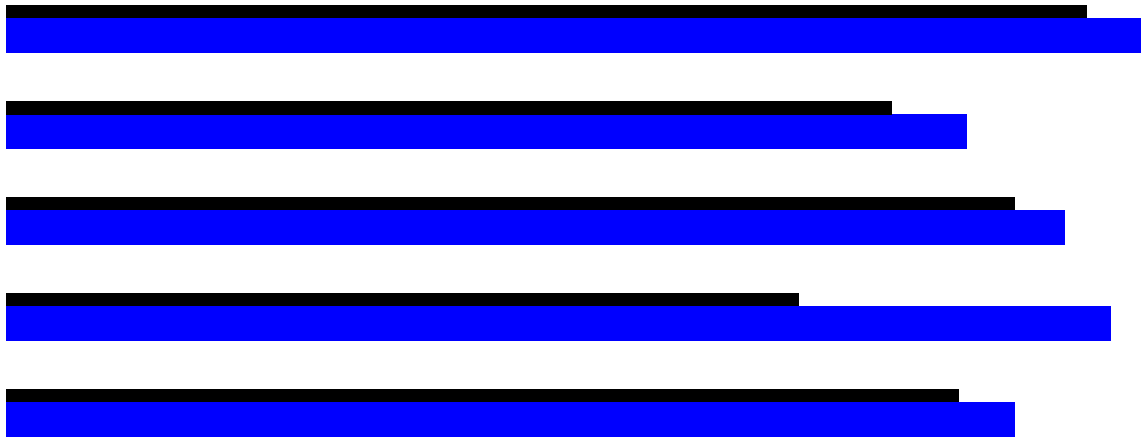




Model 204[®]



Release Notes

Version 5 Release 1

Computer Corporation of America

<i>Corporate Headquarters:</i>	<i>CCA International:</i>
500 Old Connecticut Path Framingham, MA 01701 USA	First Floor, Edinburgh House 43-51 Windsor Road Slough, Berkshire SL1 2EE England
Phone: (508) 270-6666	Phone: +44-1753-472800
Fax: (508) 270-6688	Fax: +44-1753-472888

Model 204 Release Notes Version 5 Release 1
Document Number RLN51A
March 27, 2002

Copyright © 2002 Computer Corporation of America.
All rights reserved. Printed in the United States of America.

Access/204™, Advantage/SQL™, CCA Analytics™, Connect★™, Dictionary/204™, MP/204™, MQ/204™, PQO/204™, WebGate™, and Workshop/204™ are trademarks, and ACCOLADE®, Imagine®, MarketPulse®, Model 204®, and 204® are registered trademarks of Computer Corporation of America.

IBM® is a registered trademark of International Business Machines, Inc.

Windows® is a registered trademark of Microsoft Corporation.

Other trademarks and trade names are used to identify entities claiming the marks and names of their products and are hereby acknowledged.

Any rights not expressly granted herein are reserved.

Permission to use this technical documentation (the "Document") retrieved from the Computer Corporation of America's ("CCA") server is granted, provided that (1) the copyright notice in the document appears on all copies, (2) use of this Document is for informational and non-commercial or personal and internal use only, and (3) no modifications are made to this Document. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

CCA makes no representations about the suitability of the information contained in the Document. THIS DOCUMENT IS BEING PROVIDED TO YOU "AS IS" ACCORDINGLY, CCA MAKES NO WARRANTY AS TO ITS ACCURACY OR ITS USE. CCA HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS DOCUMENT, INCLUDING ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL CCA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION CONTAINED IN THIS DOCUMENT.

Any use of the technical documentation or the information contained herein is at the risk of the user. This document may include technical or other inaccuracies or typographical errors. Changes are periodically added to this Document and CCA reserves the right to improve or make changes to the document at any time and without prior notice.

Contents

Model 204® Release Notes

Preface

Audience	ix
Introducing Model 204 electronic documentation.....	ix
Contacting CCA Customer Support	ix
Notation conventions.....	x

1 Model 204 Version 5.1 Features and Functionality

In this chapter.....	1-1
Release Notes overview	1-3
This shipment contains	1-4
Using the CCA Web site.....	1-4
Accessing Early Warnings.....	1-5
Version 5.1 features at a glance	1-6
Operational and performance improvements in recovery	1-8
Automated secondary recovery: reuse JCL	1-8
Sizing the CHKPOINT dataset correctly.....	1-9
Rerunning RESTART recovery after a successful recovery	1-9
Operational changes to ROLL BACK processing.....	1-9
Operational changes to ROLL FORWARD processing.....	1-10
RESTART recovery performance improvements	1-11
Thread signaling and synchronization facilities.....	1-13
Implementing thread signaling and synchronization in User Language	1-13
Programming thread synchronization in User Language	1-14
NonStop/204	1-18
Managing third-party backups	1-18
Programming a third-party backup	1-19
Programming a wait for extended quiesce	1-20
Submitting a backup.....	1-20
Managing the extended quiesce	1-21
Delayed ending for extended quiesce or status message.....	1-22
Freeway/204	1-24
Installation considerations for Freeway/204	1-24
CCATEMP and CCASERVR in Storage feature	1-25
Activating the CCASERVR in Storage feature for a job	1-25
Activating the CCATEMP in Storage feature for a job.....	1-25
Sample output from a MONITOR DATASPACE command	1-26

System programmers consideration.....	1-27
Activating the APSY Precompiled Procedures In Storage feature.....	1-28
Storage requirements.....	1-28
Setting the DSPOPT parameter.....	1-29
APSY load statistics.....	1-29
Extended Task Input/Output Table support.....	1-31
Using TIOT and XTIOIOT with datasets.....	1-31
Using TIOT and XTIOIOT with dynamically allocated Model 204 files.....	1-31
Special considerations for obsolete form of ALLOCATE.....	1-32
User Language enhancements.....	1-33
Defining common variables.....	1-33
VALUE IN clause processing.....	1-33
Using a FIND ALL RECORDS statement.....	1-34
Using a FIND ALL VALUES statement.....	1-35
Using a FOR EACH RECORD WHERE statement.....	1-36
IN GROUP MEMBER clause enhancement.....	1-37
Functions introduced or updated in Version 5.1.....	1-39
\$CHKPINF function.....	1-39
\$ECBDGET function.....	1-41
\$ECBDSET function.....	1-42
\$ECBTEST function.....	1-43
\$ERRCLR function.....	1-44
\$POST function.....	1-45
\$STATUS and \$STATUSD.....	1-46
\$STATUSR function.....	1-47
\$UNPOST function.....	1-47
\$VALIDATE_NUMERIC_DATA function.....	1-48
\$WAIT function.....	1-49
Storing invalid numeric data: FLOAT or ORDERED NUMERIC.....	1-53
Identifying pre-Version 5.1 floating point values.....	1-53
SQL processing floating point numbers.....	1-55
XREF support for Block Comments.....	1-56
Performance enhancements.....	1-57
General improvements.....	1-57
NTBL and QTBL made resident and shareable in virtual storage.....	1-58
FOR EACH OCCURRENCE loops.....	1-58
Managing the Record Enqueuing Table.....	1-58
Spinning on an MP lock.....	1-59
Performance improvement for global string variables.....	1-59
CHKPOINT records enhanced.....	1-60
ROLL BACK processing speed increased.....	1-60
Fast logical page reads.....	1-61
Subscripted field extraction.....	1-61
SQL processing improvements and changes.....	1-63
SELECT DISTINCT processing.....	1-63
Handling SQL statements greater than 32K bytes.....	1-64
Handling foreign keys.....	1-64
Handling primary keys.....	1-65
SQL statement enhancement.....	1-66
Options in the SELECT LIST statement.....	1-66

Additional SQL INNER JOIN features	1-68
SQL OUTER JOIN features	1-69
Restrictions using the OUTER JOIN feature	1-70
New SQL reserved words	1-71
Horizon for TCP/IP and Horizon/LE for TCP/IP	1-72
Required IBM APARs.....	1-72
Expanded syntax for the DEFINE PROCESSGROUP command.....	1-72
Parameters introduced or changed in Version 5.1	1-76
APSYPAGE: Number of virtual pages for precompiled procedures.....	1-76
AUDITOPT: Audits MQ/204 messages.....	1-76
CPQZACTN: Action to take when the CPQZSECS time limit expires.....	1-76
CPQZSECS: Maximum duration, in seconds, of an extended quiesce.....	1-77
CUSTOM: Using customized parameters	1-78
DSPOPT: Data- and hiperspace options.....	1-79
ECISUBS: Subtasks initially attached for running external modules.....	1-80
ECMODS: External modules to load.....	1-80
ECMSUBS: Number of ECF subtasks for running modules.....	1-80
ECNAMES: Number of external call names.....	1-81
ECPRIV: ECF privileges.....	1-81
ECPSIZE: Size of initial parameter area allocated by ECF.....	1-81
ECWAIT: ECF wait time	1-82
GTBLHASH: Number of buckets for global string variables.....	1-82
GTBLPCT: Initial percentage of GTBL to allocate for global string variables.....	1-83
LENQTBL: Resource enqueueing table entries	1-83
MAXBUF: File page buffer maximum.....	1-84
MAXINCL: Number of procedure INCLUDE nesting levels.....	1-85
MAXOBUF: Maximum number of open disk buffers per server	1-85
MINBUF: File page buffer minimum	1-86
MPSYS: MP/204 processing options	1-87
MSRES: Maximum number of Single Record Entries.....	1-87
NECBS: Number of Event Control Blocks (ECB).....	1-88
NJBUFF: Number of journal buffers	1-88
OPSYS: Operating system.....	1-89
RESLTHR: APSY resident load threshold.....	1-90
SCHDOPT: MP/204 scheduler operation and accounting	1-90
SYSOPT2: System options	1-92
TEMPCUR: Current CCATEMP pages in use.....	1-92
TEMPCURE: CCATEMP pages currently in use in the expansion area	1-92
TEMPCURS: CCATEMP pages currently in use in the small page area	1-93
TEMPHIE: Highest page number used in CCATEMP expansion area	1-93
TEMPHIS: Highest page number used in CCATEMP small page area	1-93
TEMPMAX: CCATEMP pages high-water mark	1-93
TEMPPAGE: Number of CCATEMP pages in storage	1-94
TEMPSIZE: Highest CCATEMP page count.....	1-94
UFEHASH: Optimization of FOR EACH OCCURRENCE loops	1-95
UFEOMISS: Number of unsuccessful optimizations for FEO	1-96
Enhanced commands	1-97
BUMP MODULE: Logging out a user from an external module	1-97
CHECKPOINT: Handling a checkpoint	1-97
CHKMSG: Returning the checkpoint status	1-98

MONITOR CFR: Monitoring critical file resources.....	1-98
MONITOR: An Online system	1-100
MONITOR: Checkpoints.....	1-101
MONITOR DATASPACE: Display dataspace and hiperspace.....	1-101
MONITOR SERV: Display server number, SERVSZ and LSERVER	1-103
MONITOR: The subsystem	1-104
MSGCTL: Setting message output.....	1-104
RENAME FILE: Renaming a file	1-104
VIEW SERVSZ: Display the size of the server	1-105
VIEW LSERVER: Display number of bytes in use	1-106
New and changed statistics in Version 5.1	1-107
Since-last statistics.....	1-109

2 Compatibility Issues

In this chapter.....	2-1
Recovery incompatibilities.....	2-2
RESTART recovery incompatibility	2-2
RESTART ROLL FORWARD recovery is version specific.....	2-2
Installation and reinstallation compatibility issues	2-3
Softspy	2-3
Additional information in CCAPRINT.....	2-3
Subsystems available at initialization.....	2-3
CCACAT field attribute change	2-3
Do not use the MERGCUST job.....	2-4
Sizing considerations	2-5
Increasing LHEAP	2-5
Changes in QTBL and VTBL requirements.....	2-5
Increase in STBL.....	2-5
Increase in server size	2-5
PDL usage increases	2-5
CCAJRNL size increase.....	2-6
CCATEMP usage.....	2-6
Increase in file page buffer minimum	2-6
Commands and parameters incompatibilities	2-7
REDEFINE command and INVISIBLE fields behavior change	2-7
Backpaging count now accurate	2-7
Changes to CHKABORT and CHKMSG commands.....	2-7
MSGCTL command behavior change	2-7
User Language incompatibilities	2-9
Changes to pattern matching	2-9
\$FSTERR and \$ERRMSG changes.....	2-9
IF statements generating compile errors.....	2-10
FIND statement performance enhancement	2-10
Using Language Environment mathematics \$functions	2-10
Changes required to user-written \$functions.....	2-10
Altered state information	2-11
Changes in information messages affecting location offset	2-11
MONITOR and MONITOR SL commands output reformatted	2-11
PRIORITY command output reformatted	2-12
Messages paired for more information.....	2-12

Displaying the running users	2-13
Returning the checkpoint status	2-13
Message for the Operator	2-13
Additional wait types	2-13
All operating systems	2-15
Assembling source	2-15
Installing Connect ★	2-15
AUDIT204 and processing journals.....	2-15
Obsolete parameter and statistic.....	2-15
Global table high water marks.....	2-15
IFAM application programs	2-15
For OS/390 and z/OS operating systems	2-17
Introducing Model 204 support for z/OS.....	2-17
Reinstalling CRAM	2-17
Installing CRAM and/or M204XSVC.....	2-17
Changes to INS204 in Version 5.1	2-17
Limited support for EXCPVR.....	2-19
CICS Transaction Server support	2-19
MQ/204 enhancements	2-19
For VM operating system	2-20
31-bit BATCH2 / IFAM2 processing	2-20
Dynamic allocation in CMS	2-20
Changes to M204GEN command in Version 5.1	2-20
For VSE operating system	2-22
CICS Transaction Server 1.1 support	2-22
Identifying AUDIT204 utility users	2-22
Horizon for TCP/IP and Horizon/LE for TCP/IP limitations.....	2-22
Limitations with sequential data files	2-22

3 Model 204 External Call Facility

In this chapter.....	3-1
Overview	3-2
Model 204 commands relating to ECF	3-4
BUMP MODULE: Logging out a user from an external module	3-4
ECF User Language statements	3-5
EXTERNAL CALL statement.....	3-5
EXTERNAL DELETE statement.....	3-5
EXTERNAL LOAD statement.....	3-6
EXTERNAL MODULE statement	3-6
EXTERNAL NAME statement	3-7
EXTERNAL START statement.....	3-8
EXTERNAL STOP statement.....	3-8
Tracking ECF	3-10
Wait types for ECF	3-10
New ECF statistics in Version 5.1	3-10
Return codes.....	3-11
\$ECFSTAT function	3-12
Introducing ECF User 0 parameters	3-14
Subtask and load module management.....	3-15
Subtasks assignment	3-15

Subtask affinity	3-15
Load modules.....	3-16
Fulfillment order	3-16
Restrictions and cautions	3-17
ECF examples.....	3-18
COBOL sample Number 1	3-18
COBOL sample Number 2	3-20
SAS/C sample.....	3-21
Assembler sample.....	3-22

4 Model 204 V5R1 Revisions to Existing Documents

In this chapter.....	4-1
Revisions to Command Reference Manual.....	4-2
SECTRLOG: Security trusted login environments	4-2
Updating options for the MSGCTRL command.....	4-2
Revisions to Horizon: Intersystem Processing Guide	4-3
Changes documented for the DEFINE LINK command.....	4-3
Revisions to System Manager's Guide	4-4
In Chapter 3: Defining the User Environment (CCAIN).....	4-4
In Chapter 12: Establishing and Maintaining Security.....	4-4
In Chapter 15: Using a System Journal and Statistics	4-5
Revisions to the User Language Manual	4-6
In Chapter 3: Basic User Language Statements and Commands.....	4-6
In Chapter 4: Record Retrievals	4-6
In Chapter 11: Flow Control in User Language	4-6
In Chapter 18: Program Communications Facilities	4-7
In Chapter 20: Global Features	4-7
Clearing the GTBL work area	4-9
In Chapter 27: User Language Functions	4-11

5 New and Revised Messages

In this chapter.....	5-1
New and revised Model 204 numbered messages	5-2
New Model 204 unnumbered messages.....	5-47
New SQL messages	5-48

Index

Preface

This document contains information about the contents of Version 5 Release 1 of Model 204. The information in this document applies to the contents of the Model 204 Version 5.1 product tape.

Audience

The audience for this document includes Model 204 end-users, application developers, installers, system managers, and file managers.

Introducing Model 204 electronic documentation

Model 204 documentation includes several other manuals to which you might want to refer. The CD-ROM, titled *Model 204 Documentation*, contains the most recently released documentation for Model 204.

The document files are in Portable Document Format; each has a PDF file extension. You can view, navigate, and print the individual manuals, and you can search the entire document set using Adobe™ Acrobat Reader™ with Search software, which is also provided on the CD-ROM. Either view the manuals directly from the CD-ROM, or download the files to a network server.

In the PDF directory, open the README.TXT file on the Windows Notepad. This file includes instructions to download a copy of the Acrobat Reader with Search and to open the *Model 204 Documentation Library Catalog*.

Note: You may access the documentation online or print out copies, as needed. However, consistent with the terms of your license agreement, you may not copy or distribute the CD-ROM, or distribute hard-copies to third parties.

Contacting CCA Customer Support

If you need assistance with this product beyond the provided online help and documentation, and you have licensed this product directly from CCA, either call CCA Customer Support at 1-800-755-4222, or access the Customer Support section of the CCA Web site. The Web address is:

`http://www.cca-int.com`

If you have not licensed this product directly from CCA, please consult your vendor.

Notation conventions

This manual 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.
UNI QUE PRI MARY 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 LI KE}	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 pulldown menu. In this example, select the Save As command from the File menu.

1

Model 204 Version 5.1 Features and Functionality

In this chapter

- Release Notes overview
- This shipment contains
- Version 5.1 features at a glance
- Operational and performance improvements in recovery
- Thread signaling and synchronization facilities
- NonStop/204
- Freeway/204
- CCATEMP and CCASERVR in Storage feature
- Extended Task Input/Output Table support
- User Language enhancements
- Functions introduced or updated in Version 5.1
- Storing invalid numeric data: FLOAT or ORDERED NUMERIC
- XREF support for Block Comments
- Performance enhancements
- SQL processing improvements and changes

- SQL statement enhancement
- Horizon for TCP/IP and Horizon/LE for TCP/IP
- Parameters introduced or changed in Version 5.1
- Enhanced commands
- New and changed statistics in Version 5.1

Release Notes overview

This document contains installation and migration information pertinent to Model 204 Version 5.1. Please read through the material for the latest information on contents and installation before you start the actual installation.

For SoftSpy documentation, refer to the *ITS Products for Model 204 Installation Guide*, *SoftSpy for Model 204 Reference Manual*, and *SoftSpy for Model 204 Message Manual*.

This shipment contains

- CPU ID zap
- Decryption keys
- *Release Notes*
- An installation guide for your operating system
- Model 204 Documentation Library CD
- Tape map
- Mainframe cartridge for your operating system:

Operating System	Contents	Label
OS/390 or z/OS	Nucleus and install	MV5100
VM	Nucleus and install	CM5100
VSE	Nucleus and install	VS5100

Using the CCA Web site

Early Warnings, object replacement modules, and *Late Breaking News* are posted on the CCA Web site. Information in these *Release Notes* is accurate for the date on the copyright page. For the most current information check the CCA Web site: Customer Support > Access Secure Maintenance Information.

To access this Secure Maintenance page, you need a user name and password from CCA. If you do not yet have yours, please send an e-mail request to:

`mai nt i nfo@cca- i nt . com`

Accessing the Web site

The following steps access the Access Secure Maintenance Information maintained on the CCA Web site:

`ht t p: //www. cca- i nt . com`

To visit the maintenance area of the CCA Web site:

1. Select the [Customer Support](#) button on the menu bar.
2. Select the [Access Secure Maintenance Info](#) link.
3. Select the [Model 204](#) link on the Customer Support page.
4. Enter your user name and password in the Username and Password Required dialog box and click OK.

On the Model 204 Support Services page, select the topic of your choice and follow the navigation provided.

5. In the [FTP Access to Early Warnings and Late Breaking News](#) link select the [Early Warnings](#) link. Then select the [Model 204 Version 5 Release 1.0](#) link.

Downloading from the Web site

To download information, enter your user name and password in the Username and Password Required dialog box and click OK.

In the Model 204 Support Services dialog box, select the topic of your choice and following the navigation provided.

Accessing Early Warnings

As the final step of installation you must apply all current 5.1.0 Early Warnings. They are available in the Access Secure Maintenance Info area on the CCA Web site. Follow the instructions in "Using the CCA Web site" on page 1-4 to access and download this maintenance.

Version 5.1 features at a glance

The following table lists the new features and functionality in Model 204 Version 5.1 and Connect*.

Table 1-1. Version 5.1 features briefly described

New feature	Briefly	Operating system	For details see
Streamlining RESTART recovery	Significant performance and operational improvements in RESTART recovery	All operating systems	“Operational and performance improvements in recovery” on page 1-8
Thread signaling and synchronization	Threads can signal that an event is complete, wait for a signal, and examine a signal.	All operating systems	“Thread signaling and synchronization facilities” on page 1-13
NonStop/204	You can use third-party software products, which run independently of Model 204, to back up Model 204 files. The Online remains running and responsive to non-update requests.	OS/390 and z/OS	“NonStop/204” on page 1-18
Freeway/204	Model 204 comes Connect ★, Horizon, and PQO/204 enabled.	All operating systems	“Freeway/204” on page 1-24 and “CICS Transaction Server support” on page 2-19
CCATEMP and CCASERVR and data- and hiperspaces	Performance improvements in 64-bit real storage environments. Eliminating server size restrictions.	OS/390 and z/OS	“CCATEMP and CCASERVR in Storage feature” on page 1-25
APSY Precompiled Procedures In Storage	Frees buffer pool pages used by Model 204 files or non-APSY CCATEMP pages	OS/390 and z/OS	“Activating the APSY Precompiled Procedures In Storage feature” on page 1-28
Extended TIOT support	A single Model 204 region can dynamically allocate and access an almost unlimited number of database and sequential files.	OS/390 and z/OS	“Extended Task Input/Output Table support” on page 1-31
Performance pack	You may experience CPU savings in Model 204 and machine resources.	All operating systems	“Performance enhancements” on page 1-57 and “Parameters introduced or changed in Version 5.1” on page 1-76

Table 1-1. Version 5.1 features briefly described (continued)

JOIN engine	You can invoke most forms of relational join from Connect ★ or Advantage/SQL, and User Language.	All operating systems	“SQL statement enhancement” on page 1-66 and “VALUE IN clause processing” on page 1-33
TCP/IP support for Horizon	Horizon was updated to TCP/IP support, including Horizon as a TCP client.	All operating systems	“Horizon for TCP/IP and Horizon/LE for TCP/IP” on page 1-72
RENAME without DUMP or RESTORE	You can rename a Model 204 database file without issuing DUMP and RESTORE commands.	All operating systems	“RENAME FILE: Renaming a file” on page 1-104
DISPLAY EW ALL at initialization	A listing of all currently applied maintenance is displayed at initialization.	All operating systems	“Additional information in CCAPRINT” on page 2-3
Support for Language Environment mathematics routines	Model 204 can use Language Environment routines for mathematics \$functions, as an alternative to the FORTRAN routines.	All operating systems	“Using Language Environment mathematics \$functions” on page 2-10
Support for CICS Transaction Server 1.3 and 2.1	You can execute server Java applications from a subsystem.	OS/390 and z/OS	“CICS Transaction Server 1.1 support” on page 2-22
VM/CMS batch above the line	Thirty-one bit storage relief for VM/CMS batch	VM only	“31-bit BATCH2 / IFAM2 processing” on page 2-20
External Call Facility (ECF)	You can invoke external processes written in any host-based language that can, in turn, invoke other programs and processes.	OS/390 and z/OS	Chapter 3
Enhanced support for IBM's MQSeries and CCA's MQ/204	You can use ?%variables in the User Language \$ construct that controls the MQSeries interface.	OS/390 and z/OS	Please consult <i>Model 204 MQ/204 Reference V5.1</i>

Operational and performance improvements in recovery

Version 5.1 introduces a number of significant improvements and enhancements to RESTART recovery. The result is a more efficient and less complex process that shortens the outage related to the recovery processes. The following describes the changes and highlights compatibility issues that you must address as you migrate to Version 5.1.

Automated secondary recovery: reuse JCL

You do not need to alter your JCL when you run RESTART recovery multiple times. Occasionally, when RESTART processing is underway, it might be interrupted:

- During ROLL BACK processing
- During ROLL FORWARD processing

Prior to Version 5.1, you would first correct the problem. If the interruption occurred:

- During ROLL BACK processing, you resubmitted the job.
- During ROLL FORWARD processing, you changed the RESTART dataset to point to the CHKPOINT dataset from the failed recovery run and defined a new CHKPOINT dataset.

As of Version 5.1, you correct the problem. If the interruption occurs:

- During ROLL BACK processing, you resubmit the job.
- During ROLL FORWARD processing, you resubmit the job. Make *no* changes to the RESTART or CHKPOINT datasets.

Model 204 determines whether this is a primary or secondary recovery run and uses the appropriate file: RESTART or CHKPOINT for ROLL BACK recovery.

Note special requirement for VSE: To take advantage of the automation of secondary recovery, you must add a new file, CHPNTD, to your primary recovery job. CHPNTD uses the same dataset name as CHPNT, but specifies DA rather than SD. Specify CCAJRNL, CCARF and RESTART as always. For example:

```
// DLBL CHPNTD, MODEL204. CHKPOINT. FILE. 2, 0, DA
// EXTENT SYS023, SYSWK2, , 2000, 13000
// DLBL CHPNT, MODEL204. CHKPOINT. FILE. 2, 0, SD
// EXTENT SYS023, SYSWK2, , 2000, 13000
// ASSGN SYS023, DISK, VOL=SYSWK3, SHR
```

If you do not include CHPNTD, then you must run secondary recovery as you did prior to Version 5.1.

Sizing the CHKPOINT dataset correctly

For OS/390 and z/OS sites, Model 204 verifies that the new CHKPOINT dataset is large enough to contain all the information that needs to be written to the CHKPOINT file during ROLL FORWARD processing. If the new CHKPOINT dataset is too small, Model 204 issues the following error message and terminates before any actual recovery processing begins:

```
M204. 2605: CHKPOINT TOO SMALL FOR ROLL FORWARD -  
number 1 BLOCKS REQUIRED; number 2 FOUND
```

The CHKPOINT dataset must also be a single-volume dataset due to BSAM limitations regarding read-backwards. This requirement is *not* new in Version 5.1; it has always been the case.

Rerunning RESTART recovery after a successful recovery

If, for any reason, you want to force a rerun of your successful recovery job, you must define a new CHKPOINT dataset. If you rerun your recovery job without a new CHKPOINT dataset, recovery will be bypassed and the following message is displayed:

```
M204. 0143: NO FILES CHANGED AFTER LAST CP, RESTART  
BYPASSED
```

Operational changes to ROLL BACK processing

Beginning in Version 5.1, the ROLL BACK process ends with a checkpoint.

ROLL BACK processing prior to Version 5.1

Prior to Version 5.1, when the ROLL BACK processing completed successfully, the following message was issued:

```
M204. 0158: END OF ROLLBACK
```

ROLL BACK processing in Version 5.1

Under the same circumstances in Version 5.1, when ROLL BACK processing is completed, the following messages are displayed:

```
M204. 0158: END OF ROLLBACK  
M204. 0843: CHECKPOINT COMPLETED ON yy. ddd hh: mmss. th
```

CHKPOINT dataset required

Changing the point at which the checkpoint is taken has the following impact. Beginning in Version 5.1, the CHKPOINT dataset is required for any RESTART recovery processing, because the ROLL BACK facility needs the CHKPOINT dataset to write the end-of-processing ROLL BACK checkpoint.

If you do not define a CHKPOINT dataset, the following message is issued:

```
M204. 1300: RESTART COMMAND REQUIRES CHECKPOINT
LOGGING - RUN TERMINATED
```

This is unlike prior versions of Model 204, in which the CHKPOINT dataset was not required for ROLL BACK-only processing.

Operational changes to ROLL FORWARD processing

Tracking the application of updates

As the CCARF dataset is processed by ROLL FORWARD, the following messages are printed each time a new journal block is read in which the hour in the date-time stamp at the beginning of the journal block is one hour (or more) greater than the hour in the last M204.1992 message printed. For example,

```
M204. 1992: RECOVERY: PROCESSING ROLL FORWARD BLOCK#
0000001B 01. 235 16: 31: 32. 43
M204. 1992: RECOVERY: PROCESSING ROLL FORWARD BLOCK#
00001C57 01. 235 17: 00: 23. 34
M204. 1992: RECOVERY: PROCESSING ROLL FORWARD BLOCK#
00002F05 01. 235 18: 00: 12. 38
```

These messages are intended to provide an indication that ROLL FORWARD processing is progressing and help you estimate when recovery will complete. The ROLL FORWARD BLOCK# is the sequential number (from the beginning of CCARF) of the journal block read at that point during ROLL FORWARD processing. See also “Message for the Operator” on page 2-13.

ROLL FORWARD processing can now be run separately

In Version 5.1, subsequent to a successful ROLL BACK process, you can invoke ROLL FORWARD processing in a separate job or job step by issuing the following command:

```
RESTART ROLL FORWARD
```

The following message does not indicate that the full ROLL BACK processing is occurring; this message indicates that the correct starting point for ROLL FORWARD processing is being located.

```
M204. 2512: ROLL BACK WILL USE THE FOLLOWING DATASET:
RESTART | CHECKPOINT
```

If Model 204 detects that a successful ROLL BACK process did not occur, it forces a full ROLL BACK process and all the standard ROLL BACK messages are displayed.

Setting RCVOPT when running ROLL BACK and ROLL FORWARD separately

When running ROLL BACK-only CCA recommends setting RCVOPT parameter to 0. If you then run a subsequent ROLL FORWARD-only step, the ROLL BACK processing will be suppressed. If you do run ROLL BACK-only with RCVOPT set including 1, do not set CPMAX TO 0 OR 1.

- If you run ROLL BACK-only with an RCVOPT setting that does include X'01' and the CPMAX parameter is set to 0 or 1, a subsequent ROLL FORWARD-only request must process ROLL BACK again to find the correct checkpoint for ROLL FORWARD processing to locate and apply the ROLL FORWARD updates.
- If in your ROLL BACK-only step RCVOPT included X'01' and the CPMAX parameter was set greater than one, you can suppress the ROLL BACK processing on a ROLL FORWARD-only step by supplying the ROLL BACK TO checkpoint ID information that was previously rolled back to. Issue a RESTART command, as shown in the following syntax:

```
RESTART ROLL BACK TO yy. ddd hh: mm: ss. t h ROLL  
FORWARD
```

ROLL FORWARD-only processing uses this information to locate the correct ROLL FORWARD starting point in the CCARF dataset.

If you do not provide the checkpoint ID information, then the ROLL BACK processing is automatically triggered to locate the correct starting point for ROLL FORWARD processing.

- Also, if running a ROLL BACK-only process, the CCARF file definition is no longer required.

ROLL FORWARD processing, in any context, requires RCVOPT=1. CPMAX should be set to a value that meets your site's need. Your current RESTART configuration should reflect whether you take the CPMAX default value or set some other value.

RESTART recovery performance improvements

In addition to the performance improvements realized by the preceding operational changes to RESTART recovery, Version 5.1 has the following performance improvements related to RESTART recovery.

Finding checkpoint in journal improved

Prior to Version 5.1, the CCARF dataset was searched from the beginning for the checkpoint record rolled back to in ROLL BACK processing. This checkpoint contains a date-time stamp, which was the starting point for ROLL FORWARD processing.

In Version 5.1, locating this checkpoint record is done using the NOTE/POINT facility provided for BSAM datasets. The NOTE/POINT facility allows direct

access to a single record in a file of any size. This eliminates the potentially long delay currently encountered between the end of ROLL BACK processing and the beginning of ROLL FORWARD processing. The NOTE/POINT facility handles all forms of CCARF datasets, including ring streams, parallel streams, and single sequential datasets.

Note special requirement for VSE: Finding the checkpoint in a VSE journal is accomplished as it was prior to Version 5.1. IBM restricts the NOTE/POINT facility for sequential data files under VSE.

Dynamic file allocation bypassed

If RESTART recovery processing determines that no files need to be recovered, dynamic allocation is bypassed, which saves CPU and wall clock time.

Note: In Version 5.1, if your Online is in the same step as RESTART, it can no longer depend on dynamic allocation by RESTART processing. Instead, you must explicitly allocate all required files using DD, DLBL, or FILEDEF statements or use the Model 204 ALLOCATE command. See “ROLL BACK processing speed increased” on page 1-60.

Thread signaling and synchronization facilities

In some applications, User Language programmers need to coordinate processing of two or more threads. For example, one thread may perform process-to-process communication, and then signal the completion of the process to a user represented by a SNA Communications Server (formerly VTAM) thread. At that moment, the user may be working or waiting for the signal.

Prior to Version 5.1, you could implement a synchronization facility by having an application or users constantly or periodically check data in a shared file. When that data is changed by another application or user, that change signals the completion of an event, which indicates other applications or users can now proceed with their processing. However, this method may waste considerable resources.

In Version 5.1, User Language has new signaling and synchronization capabilities. A thread can:

- Send a signal that an event associated with the signal is finished
- Wait for the signal to arrive and examine the contents of a signal

Implementing thread signaling and synchronization in User Language

The signal is implemented using an Event Control Block (ECB), a permanent data structure in memory that indicates the event state and user supplied data associated with the event. The ECB is shared by all threads participating in the synchronization process.

- To send a signal to a thread, a User Language application uses the \$POST function, which marks the event as posted (or completed) and causes the waiting thread to wake up and resume processing. Furthermore, you can supply a numeric post code that is stored in the ECB, which may be examined by the waiting thread or threads.
- To wait for a signal, a User Language application uses the \$WAIT function that puts a thread in a wait state until the event is posted by another thread and the signal arrives.

Introducing ECB-related functions

Using the \$POST and \$WAIT functions, you can synchronize the execution of two threads. A User Language application may associate a string of text with an ECB using the \$ECBDSET function; the text may be read by other threads using \$ECBDGET function.

The following \$functions are available in Version 5.1 for implementing signaling and synchronization between threads:

Table 1-2. \$Functions for signaling and synchronization

\$Function	Purpose	Details found in ...
\$ECBDGET	Retrieve string data associated with an ECB.	"\$ECBDGET function" on page 1-41
\$ECBDSET	Associate a string of data with an ECB.	"\$ECBDSET function" on page 1-42
\$ECBTEST	Test an ECB to see whether it is posted or not.	"\$ECBTEST function" on page 1-43
\$POST	Indicate that an event has occurred. The thread waiting on an ECB can resume processing.	"\$POST function" on page 1-45
\$UNPOST	Indicate that an event did not occur or recur.	"\$UNPOST function" on page 1-47
\$WAIT	Suspend a user until an event happens and the ECB is posted.	"\$WAIT function" on page 1-49

You can also use these functions to implement a third-party software backup plan for Model 204; see "NonStop/204" on page 1-18.

Assigning and using an ECB number

The ECBs used to synchronize threads are referenced by number in the ECB-related functions. User Language programmers should carefully choose the ECB numbers for use by threads and applications.

Programming thread synchronization in User Language

Programming thread synchronization in User Language using the ECB-related functions requires a clear understanding of how threads interact with each other. You must avoid situations where deadlocks are possible; for example, when Thread A waits on the signal from Thread B, which is already waiting on a signal from Thread A. In this situation, both threads are unable to run and will wait indefinitely until canceled with a BUMP command.

You must take into consideration that once posted, the ECB remains posted until explicitly unposted using the \$UNPOST function. The \$WAIT function does not change the ECB contents after completion, so the next invocation of the \$WAIT function with the unchanged ECB has exactly the same effect as the first one. Use the \$UNPOST function after all the threads waiting on the event have had a chance to run and check the ECB.

Posting and unposting ECBs

Posting an ECB does not mean that all threads waiting on that ECB can run immediately. Those waiting threads are now eligible to run and will be dispatched by the Model 204 scheduler based on priority and available resources.

If an ECB is posted and then immediately unposted, any threads that have not yet run and are still waiting on that ECB will never see the post and will continue to wait.

Only an unposted ECB may be posted. Attempting to post an already posted ECB produces a nonzero return code; the content of the ECB is not changed.

A string may be associated with an ECB using the \$ECBDSET or \$POST functions. However, in most cases you should use the \$POST function to associate a string with an ECB. The \$POST function stores a string *only* if the referenced ECB is unposted; the \$ECBDSET function stores a string regardless of the state of the ECB.

Thread synchronization programming details

The number of ECBs available to an Online is determined by the NECBS parameter set in CCAIN. The parameter cannot be reset. See “NECBS: Number of Event Control Blocks (ECB)” on page 1-88.

ECBs are referenced by a number from 1 to NECBS and are available to any User Language application running on any thread. Therefore, care must be taken when assigning an ECB number to a process or an application to avoid conflict. A thread or an application cannot reserve an ECB, because it is universal in scope.

You may use a %variable to program an ECB number in all the ECB-related \$functions.

If the \$ECBDSET function associates a string with an ECB more than once, the \$ECBDGET function can retrieve only the most recent string.

Pay special attention to initialization of the synchronization process, since some applications may be sensitive to the order in which threads are initialized. When Model 204 is initialized, all ECBs are reset to zero, indicating that no events have yet happened. A thread restart has no effect on any ECBs. Therefore, Thread A waiting on a particular ECB will remain waiting even if Thread B, which intended to post that ECB, is restarted prior to the evaluation of the \$POST call.

Examples of programming thread synchronization

To better understand how User Language synchronization is implemented in Version 5.1, let's look at the following example. Thread A (server) performs a service for many client threads. To request a service, a client thread B posts ECB 5, sets a post code indicating the required service, and passes the ECB

number. It waits for the service completion, which will be a string associated with ECB 5.

The client thread B also checks whether the post operation was successful. Another client thread C may have already posted ECB 5. In that case, the client thread B waits on ECB 3, which is posted when server thread A finishes with the current request. No provisions for restart are made. If thread A is bumped or fails before it posts ECB 12, client thread B is left in a permanent wait state. A BUMP command or a procedure, which posts thread B with a post code indicating a restart, may be required.

```

Server BEGIN
thread A REPEAT FOREVER
    %X = $POST(3, 1)
    %X = $WAIT(5)
    %CLIENT.ECB = $ECBDGET(5)
    %POSTCODE = $STATUSD
    %X = $UNPOST(5)
    %X = $UNPOST(3)
    IF %POSTCODE = 1 THEN
        PERFORM TYPE 1 SERVICE
    ELSEIF %POSTCODE = 2 THEN
        PERFORM TYPE 2 SERVICE
    ELSEIF %POSTCODE = 10 THEN
        /* REQUEST TO BRING DOWN THE SERVER THREAD */
        %X = $POST(%CLIENT.ECB, 0)
        /* INDICATE TO CLIENT THE REQUEST COMPLETED */
        JUMP TO EXIT
    END IF
    IF SERVICE SUCCESSFUL THEN
        %X = $POST(%CLIENT.ECB, 0)
        /* INDICATE TO CLIENT THE REQUEST */
        /* COMPLETED SUCCESSFULLY */
    ELSE
        %X = $POST(%CLIENT.ECB, 100)
        /* INDICATE AN ERROR RETURN */
    END IF
    END REPEAT
EXIT: END

```

```

Client BEGIN
thread B %MYECB = 12
    %ACTION = 1
    START:
    %X = $POST(5, %ACTION, %MYECB)
    /* POST SERVER THREAD AND PASS REQUEST TYPE */
    /* AND ECB NUMBER */
    IF %X = 14 THEN
        /* IF ECB ALREADY POSTED WAIT TILL SERVER */
        /* IS FREE */
        %X = $WAIT(3)

```

```
        JUMP TO START
    END I F
    %X = $WAIT(%MYECB)
    %POSTCODE=$STATUSD
    %X = $UNPOST(%MYECB)
* CHECK %POSTCODE AND CONTINUE PROCESSING
END
```

Each client thread should use a different ECB number for %MYECB.

Note: Server thread A code first copies the post code and data into %variables, then it unposts ECB 5. As soon as ECB 5 is unposted, a client thread may post it again and change the post code and associated string.

NonStop/204

Version 5.1 includes the capability to use third-party software to backup Model 204 files. Third-party backups run independently of Model 204, while the Online remains running and responsive to non-updating activity. This provides a greater ability to run Model 204 in a nonstop 24*7 mode.

The following describes the functionality and facilities provided in Version 5.1 that allow you to develop, integrate, and manage your third-party software backup methodology.

Managing third-party backups

As third-party software is unaware of Model 204 file management, it must be assured that when a file is being backed up that the file is logically and physically consistent. An external backup of a file taken when Online updates are occurring is unreliable and must not be used in any subsequent attempt to restore or recover the file. To assure file integrity, all updating must cease and remain stopped until the backup processing is complete.

Understanding an extended quiesce

Beginning in Version 5.1, you can create a period of time during which files cannot be updated. By using a set of new commands, two new CCAIN parameters, and a set of \$functions, you can create an interval, after a checkpoint is taken, during which updating is stopped. This interval is called the extended quiesce interval or, simply, an **extended quiesce**.

All file updating users remain in a swappable wait for the duration of the extended quiesce. They receive the following message after the extended quiesce ends.

```
M204. 2614: EXTENDED QUIESCE IN PROGRESS - UPDATING
SUSPENDED, PLEASE WAIT
```

Signing on IFAM threads during an extended quiesce

When single or multicursor IFAM threads signs on to Model 204, they issue the following message: M204. 0962 SIGN ON, JOB NAME = XXXXXXXX'. During an extended quiesce, IFAM2 threads (IODEV=23) cannot sign on to Model 204. If these threads try to sign on during extended quiesce, they are placed in a swappable wait of type 20. The sign-on messages are not displayed. The wait ends when the extended quiesce terminates. Then, the waiting threads can sign on. When signing on is complete, the MONITOR and IFAMSTAT commands can display their status

Coordinating an external backup

External backups must be taken only during an extended quiesce. The duration of the extended quiesce depends on when a CHECKPOINT END EXTENDED

QUIESCE command is issued or, on the values of the parameters, CPQZSECS and CPQZACTN. For more discussion of the parameters, see “CPQZACTN: Action to take when the CPQZSECS time limit expires” on page 1-76 and “CPQZSECS: Maximum duration, in seconds, of an extended quiesce” on page 1-77.

You can create and manage an extended quiesce with the following System Manager commands:

- **CHECKPOINT SET Extended Quiesce**
When issued, this command places the Online into an extended quiesce immediately after the next successful checkpoint.
- **CHECKPOINT UNSET Extended Quiesce**
The UNSET option reverses the effect of the SET option.
- **CHECKPOINT END Extended Quiesce**
This command terminates an extended quiesce.

For more discussion of these CHECKPOINT commands, see “CHECKPOINT: Handling a checkpoint” on page 1-97.

Ring stream journals and extended quiesce

If a ring stream journal is in use, an offload is automatically attempted at the beginning of the extended quiesce. If the offload is successful, the next ring stream member is made active. At the end of the extended quiesce, a checkpoint is taken and a checkpoint record is written on the active journal member. If a subsequent RESTART recovery is required, the active journal member will contain a checkpoint record.

Programming a third-party backup

The automated submission of third-party backup jobs requires a User Language interface to the internal extended quiesce data structures. This is provided primarily by using the ECB-related \$functions, \$ECBDGET, \$ECBDSET, \$POST, \$UNPOST, and \$WAIT as well as the named ECBs used for extended quiesce: CPQZ and QZSIG. Both ECBs are defined internally to the checkpointing subsystem.

- The CPQZ ECB is automatically posted at the start of an extended quiesce and automatically unposted at the end of an extended quiesce.
By specifying CPQZ as a keyword, you can use the \$ECBDGET and \$ECBDSET functions to write and read up to 255 bytes of data to and from a checkpoint-maintained internal area. This data area is cleared automatically at the end of extended quiesce.
- A User Language thread can post the QZSIG ECB by issuing a \$POST function only during an extended quiesce.

Another User Language thread can wait on the posting of the QZSIG ECB, but only during an extended quiesce interval. Outside an extended quiesce, the QZSIG ECB is unposted.

See “Functions introduced or updated in Version 5.1” on page 1-39 to examine the syntax and use of the \$ECBDGET, \$ECBDSET, \$ECBTEST, \$POST, \$UNPOST and \$WAIT functions.

Programming a wait for extended quiesce

The following User Language procedure demonstrates the use of the extended quiesce and the named ECBs to submit and control a third-party back-up job.

```
PROCEDURE WAIT.FOR.EXTENDED.QUIESCE
BEGIN
  IF $SETG('HDRCTL', $VIEW('HDRCTL')) THEN
    PRINT 'GTBL NOT SET - BACKUP WILL NOT BE SUBMITTED'
    STOP
  ELSE
    PRINT '$WAIT CPQZ' with $WAIT('CPQZ')
  END IF
END
R HDRCTL 3
USE $JOB
DISPLAY BACKUP.MODEL204.DATABASE.FILES.JOB
R HDRCTL ?&HDRCTL
BEGIN
  PRINT '$WAIT QZSIG - WAIT UNTIL BACKUP COMPLETES' -
    WITH $WAIT('QZSIG')
  IF $ECBDGET('CPQZ') = 'GOOD BACKUP' THEN
    PRINT 'BACKUP SUCCESSFUL'
  ELSE
    PRINT 'BACKUP UNSUCCESSFUL'
  END IF
END
CHECKPOINT END EXTENDED QUIESCE
END PROCEDURE
```

Once the extended quiesce starts, the previous User Language request continues to the \$WAIT('QZSIG') function call and waits for third-party back-up job to end.

Submitting a backup

In the following procedure, BACKUP.MODEL204.DATABASE.FILES.JOB, the BACKUP step executes an IEFBR14—essentially a no-operation. However, this job could be the submission of any third-party backup software that meets your backup requirements. The subsequent BATCH2 steps are executed depending upon the condition code from the step named BACKUP.

```

PROCEDURE BACKUP. MODEL204. DATABASE. FILES. JOB
//PLACE A SITE-VALID JOBCARD HERE
//BACKUP EXEC PGM=IEFBR14, COND=EVEN
//DD1 DD DISP=SHR, DSN=YOURDATASET. TOBACKUP
//* ITFAILED STEP IF BACKUP DID NOT END WITH CONDITION CODE 0
//* EOJ THE ONLINE
//ITFAILED EXEC PGM=BATCH2, PARM='CRIOCHNL', COND=(0, EQ, BACKUP)
//STEPLIB DD DSN=MODEL204. LOADLIB, DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CCAOUT DD SYSOUT=*
//CCAIN DD *
LOGIN SUPERKLUGE
PIGFLOUR
BEGIN
    PRINT 'PASS INFO TO $WAIT QZSIG WAITER ' -
        WITH $ECBDSET('CPQZ', 'BAD BACKUP')
    PRINT '$POST QZSIG - WAKE UP $WAIT QZSIG WAITER ' -
        WITH $POST('QZSIG')
    END
EOD
EOJ
LOGOUT
//* ITWORKED STEP IF BACKUP DID COMPLETE WITH CONDITION CODE 0
//* END THE EXTENDED QUIESCE IN THE ONLINE
//ITWORKED EXEC PGM=BATCH2, PARM='CRIOCHNL', COND=(0, NE, BACKUP)
//STEPLIB DD DSN=MODEL204. LOADLIB, DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CCAOUT DD SYSOUT=*
//CCAIN DD *
LOGIN SUPERKLUGE
PIGFLOUR
BEGIN
    PRINT 'PASS INFO TO $WAIT QZSIG WAITER ' -
        WITH $ECBDSET('CPQZ', 'GOOD BACKUP')
    PRINT '$POST QZSIG - WAKE UP $WAIT QZSIG WAITER ' -
        WITH $POST('QZSIG')
    END
LOGOUT
/*
//
END PROCEDURE

```

Managing the extended quiesce

In the previous example, should a CPQZSECS time-out cause a CPQZACTN argument to issue an internal CHECKPOINT END EXTENDED QUIESCE command, the ending of the extended quiesce is delayed until the thread, which issued the \$WAIT('QZSIG'), drops out of wait (wait type 48). The \$WAIT('QZSIG') can be satisfied by one of the following:

- Have another thread issue a \$POST('QZSIG') function call

- Bump the thread in wait 48
- Issue an EOJ command

Use of the \$WAIT('QZSIG') function is an aid to prevent the inadvertent ending of an extended quiesce while a third-party backup job is running.

Note: A \$POST('QZSIG') function call can be issued only once per extended quiesce. A \$ECBDSET function call may not be issued after a \$POST('QZSIG') function has been issued.

The previous code examples perform no backups and are not intended as full representations of what you can or should use as your backup strategy.

BATCH2 backup requirements

To use a BATCH2 connection to notify or signal the Online as to how to proceed when your backup processing is complete, the Online needs:

- BATCH2 IODEV definition
- CRAM installed and activated
- CRIOCHNL defined in the CCAIN stream, if it is not using the default CRIOCHNL name.

Delayed ending for extended quiesce or status message

During an extended quiesce, the following message is displayed on the operator's console every minute, or the CPQZSECS time limit if it is less than 60.

```
M204.2610. SYSTEM ENTERED EXTENDED QUIESCE AT:
mm/dd/yy hh:mm:ss, ALL FILE UPDATING REMAINS
SUSPENDED { -- END EXTENDED QUIESCE PROCESSING IS
DELAYED WITH nnnn USER(S) IN WAIT TYPE 47, mmmm
USER(S) IN WAIT TYPE 48}
```

When the extended quiesce ends, one of the versions of M204.2613 message is displayed on the operators console.

```
M204.2613. SYSTEM ENTERED EXTENDED QUIESCE AT:
mm/dd/yy hh:mm:ss, SYSTEM EXITED EXTENDED QUIESCE
AT: mm/dd/yy hh:mm:ss, REASON = reason-text
```

where *reason-text* is one of the following:

- END COMMAND ISSUED BY USER: nnnnnn
- EOJ COMMAND
- CPQZACTN TIME OUT ACTION
- CHKPPST RESTART

- Issuing a CHKMSG or MONITOR CHECKPOINT command during extended quiesce displays the M204.2610 message, in addition to other CHKMSG or MONITOR CHECKPOINT command information.
- Issuing a CHKMSG or MONITOR CHECKPOINT command after an extended quiesce ends displays a M204.2613 message with other checkpoint information.

If an extended quiesce is to end, because either a CHECKPOINT END EXTENDED QUIESCE command was issued, or X'40' is specified for ENDEQ as the CPQZACTN argument at CPQZSECS timeout, and there is a User Language thread in \$WAIT('QZSIG') wait, the M204.2610 message is redisplayed on the operators console every minute as follows:

```
M204. 2610.  SYSTEM ENTERED EXTENDED QUI ESCE AT:  
yy/ddd hh: mm: ss,  ALL FILE UPDATING REMAINS SUSPENDED  
- - END EXTENDED QUI ESCE PROCESSING I S DELAYED WITH  
nnnn USER(S)  I N WAIT TYPE 48.
```

This form of the M204.2610 message can also be displayed with the CHKMSG or MONITOR CHECKPOINT commands.

Freeway/204

Beginning in Version 5.1, 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 the feature, a zap will be provided which will open the door to expanded usage. The software features offered are:

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

Furthermore, SNA Communications Server 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/204 communication capabilities, is the cornerstone for other Computer Corporation of America products: CCA Analytics and WebGate.

Installation considerations for Freeway/204

Beginning in Version 5.1, the installation process has been simplified by incorporating SNA Communications Server, Horizon, and PQO code directly into the core Model 204 nucleus code by default.

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:

- For OS/390 or z/OS sites— found in the INSPARM member of the INS204.JCLLIB dataset.
- For VM sites—found in the M204GEN command.

Customers who are already enjoying the advantages of these products will see no growth in code size, but should review the abbreviated install process for changes that might affect their final configuration. For OS/390 or z/OS sites, see “CICS Transaction Server support” on page 2-19

CCATEMP and CCASERVR in Storage feature

This feature is for sites which:

- Have z/OS in 64-bit mode and real storage larger than 2-gigabytes, or
- Run OS/390 or z/OS in 31-bit mode and have enough expanded storage to support dataspace with no excessive paging. A **dataspace** is an address space in OS/390 with none of the control structures for tasks; it is simply data.

Other operating system configurations will begin excessive paging, and serious performance degradation of the entire system is likely to occur.

If you have enough real storage on your system, you can place either or both CCATEMP file and CCASERVR file in storage. The benefit to you depends on how much latent wait time is spent by your jobs waiting for the CCATEMP file and the CCASERVR file I/O to complete.

- Users may server swap to CCASERVR file in-memory instead of to disk dataset CCASERVR.
- There is no CYLINDER size restriction on SERVSZ, which is limited only by available memory.
- The CCATEMP file initialization is not required with CCATEMP file in Storage feature. This significantly reduces the time required for Model 204 initialization when a large CCATEMP file is required.
- Less CPU is consumed per logical CCATEMP I/O. A CCATEMP I/O, with this feature, is logical since data is moved in-memory and not to or from an external device.
- An additional benefit is that relief is provided on I/O subsystems. Because there is less contention, channels and disks are free to perform other services and jobs run faster.
- The APSYPAGE parameter may complement the CCATEMP and CCASERVR in Storage feature; see “Activating the APSY Precompiled Procedures In Storage feature” on page 1-28.

Activating the CCASERVR in Storage feature for a job

To activate the CCASERVR in Storage feature, comment out the CCASERV DD statements in a job. Model 204 will allocate the correct amount of storage based on the SERVSZ and NSERVS parameters.

Activating the CCATEMP in Storage feature for a job

To activate the CCATEMP in Storage feature in a job, you must set a new parameter in the USER0 CCAIN stream. This parameter is `TEMPPAGE=number-of-CCATEMP-pages-to-allocate`. You may comment out

or remove the CCATEMP DD record(s), but this is not a requirement. If the TEMPPAGE parameter is set, the DDs for the CCATEMP file are ignored by Model 204, although not the operating system.

Sample output from a MONITOR DATASPACE command

The DATASPACE option of the MONITOR command displays dataspace and hiperspace information as shown in the following example. *Hiperspace* is an area of virtual storage requested from the hardware's expanded storage. Access to hiperspace saves you from paging out to disk.

NAME	DATASPACE	TYPE	4K PAGES	PAGE HWM	EXTRA DATASPACES
CCASERVR	HI PERSPACE	- SCROLL	912	392	
CCATEMP	DATASPACE		3474	2396	
CCAAPSY	HI PERSPACE	- CACHE	52	36	

NAME	READS	WRITES	PAGES READ	PAGES WRITN	SLOWRD	SLOWR
CCASERVR	12	21	357	687	0	364
CCATEMP	1561	1575	2357	2378	0	0
CCAAPSY	2	3	13	33	0	33

Since IBM support for access to real storage above two gigabytes is limited to 2-gigabyte chunks of storage, multiple 2-gigabyte dataspace are chained together and logically mapped as either CCATEMP or CCASERVR.

Under the 4K PAGES heading, each X'00080000' number of 4K blocks is a 2-gigabyte chunk. There are two logical dataspace, CCATEMP and CCASERVR. However, since IBM requires that every dataspace be named, the additional (if required) physical, 2-gigabyte chunks have generated sequential names beginning with CCA00001.

Sample output from the operating system

There are five physical dataspace in the following example. When a TSO user issues the following command, where DVUSER2 is the job name of the Online using the dataspace, the following output is displayed:

/D J, DVUSER2

```

IEE115I  11. 27. 40 2001. 155 ACTIVITY 900
  JOBS   M/S    TS USERS   SYSAS   INITS   ACTIVE/MAX VTAM   OAS
00024  00034   00015   00026   00190   00015/00065   00024
  DVUSER2 ONLI NE           NSW  J  A=00AE   PER=NO   SMC=000
                               PGN=088   DMN=020   AFF=NONE
                               CT=122. 307S  ET=590. 945S
                               WUI D=JOB07991  USERI D=DVUSER2
                               ADDR SPACE ASTE=13950B80
                               DSPNAME=CCA00003  ASTE=055A5300
  
```

DSPNAME=CCATEMP ASTE=033CCC80
DSPNAME=CCA00002 ASTE=055A5400
DSPNAME=CCA00001 ASTE=09A21100
DSPNAME=CCASERVER ASTE=033CC880

System programmers consideration

To use the CCASERVER and CCATEMP in Storage feature, be aware that the maximum number of data- or hiperspaces and the total cumulative size of data- or hiperspaces for a given job may be controlled by the SMF IEFUSI exit.

In this exit, the parameter description shows that subwords 2 and 3 of the word 7 parameter are the keys, where:

- Word 7 subword 2 controls the maximum cumulative data- or hiperspace size
- Word 7 subword 3 controls the maximum number of data- or hiperspaces that a job may allocate

Check the IBM supplied settings and change them, if necessary.

Information regarding this exit routine is found in the *OS/390 MVS Installation Exits Document Number SC28-1753*.

Activating the APSY Precompiled Procedures In Storage feature

Keeping precompiled procedures in storage removes them from the disk buffer pool, which makes buffer pool pages available for Model 204 files or non-APSY CCATEMP pages. This can reduce disk I/O for both CCATEMP and other Model 204 files.

To activate the APSY Precompiled Procedures in Storage feature in a job, users must set the APSYPAGE parameter in the USER0 CCAIN stream. The APSYPAGE parameter specifies the number of virtual storage pages to allocate.

Note: While APSYPAGE looks very much like TEMPPAGE, the units are somewhat different:

- APSYPAGE refers to 4096 byte virtual storage pages
- TEMPPAGE refers to 6184 byte Model 204 file pages

There are several reasons to use the APSYPAGE parameter instead of, or in addition to, the TEMPPAGE parameter:

- APSYPAGE virtual storage pages do not have to be big enough to hold all of CCATEMP or even all saved compilations. The APSY Precompiled Procedures in Storage feature uses an LRU algorithm to ensure that frequently APSY loaded procedures remain in storage, while infrequently loaded procedures tend to be loaded from CCATEMP. So, if there is not sufficient real storage to back all of CCATEMP, you might set APSYPAGE to save frequently loaded APSY compilations in real storage.
- The path length for retrieving a page out of storage is significantly lower than for retrieving one out of the disk buffer pool, especially in an MP/204 environment.
- When APSYPAGE is set, the large Model 204 server tables are (hardware) page-aligned as are the saved compilations. This makes it possible to use the hardware MVPG (MoVe PaGe) facility to move pages into a server during an APSY load. On some processors, specialized hardware makes the facility significantly faster than byte-oriented data movement.

Storage requirements

APSY precompiled procedures can be saved in expanded storage under OS/390, although this is not an issue on a z-Series machine.

The CCASERVER, CCATEMP and APSY Precompiled Procedures in Storage features cause operating system paging. Performance degradation is likely, if real storage frames are insufficient, or expanded storage frames in hiperspaces are used to hold the data saved in this storage. The MONITOR

DATASPACE command can provide some information about the storage usage of these features.

Setting the DSPOPT parameter

If the APSY Precompiled Procedures In Storage feature is used, or CCASERVR In Storage is used with page-oriented data movement (DSPOPT X'01' set), then certain large server tables and servers themselves are page-aligned. This might require up to five extra hardware pages of real and virtual storage per server, although more typically it requires about three extra pages or 12K bytes per server. If NSERVS is set to 40 and APSYPAGE is set to a nonzero value, or CCASERVR is kept in storage with DSPOPT X'01' set, enough real storage frames should be available to hold up to an extra 200 pages or 800K of server data.

The setting of the DSPOPT parameter depends on your installation. If your site has enough expanded storage, you might use it for CCASERVR In Storage and APSY Precompiled Procedures In Storage in a cache hiperspace (DSPOPT=X'43'). If this decision causes a large number of CCATEMP reads because OS/390 steals a lot of pages from the cache hiperspace, then use DSPOPT=X'23' (no cache hiperspace). If there is an excessive paging, eliminate one or both In Storage features. If your installation is z/OS, 64-bit real-storage, you should not use any hiperspaces options, but use dataspace instead.

Users with limited resources should try and test what feature is the most effective.

In addition, enough real storage should be available to hold the APSY precompiled procedures and/or CCASERVR data in real or expanded storage. See "DSPOPT: Data- and hiperspace options" on page 1-79.

APSY load statistics

There are three system statistics associated with APSY loads. These statistics are intended to assist in the proper setting of the APSYPAGE and RESLTHR parameters and to determine the ratio of requests using pre-compiled and non-precompiled requests, the percentage of requests which are precompiled being the ratio of APSYLD to REQ. The APSY load statistics are:

Statistic	Tracks the number of APSY loads...
APSYLD	Including internal CCASYS procedures that are run as part of APSY processing.
APSYLDD	From a dataspace. This statistic is zero unless the APSYPAGE parameter is set. This parameter can be compared with APSYLD and APSYLDT to determine the percentage of eligible APSY loads that are being performed from a dataspace.

Statistic	Tracks the number of APSY loads...
APSYLDT	<p data-bbox="613 254 764 281">That are tiny.</p> <p data-bbox="613 296 1414 478">All APSY loads, even those done from dataspace, read a certain amount of control information from CCATEMP. Immediately following this control information is data that is loaded into NTBL, QTBL, STBL, and VTBL. If this table data does not extend to an extra CCATEMP page beyond the control information, the APSY load is counted as a <i>tiny load</i>.</p> <p data-bbox="613 493 1414 617">This statistic is maintained because tiny APSY loads are not worth saving in dataspace, so it must be considered when comparing APSYLDD with APSYLD in determining which percentage of pre-compiled procedures are being loaded from dataspace.</p>

Extended Task Input/Output Table support

The OS/390 and z/OS operating systems Task Input/Output Table (TIOT) places a limit of 3,273 datasets (allocated dynamically or with DD statements) in any, not just Model 204, job step. To bypass this limit, you can use the Extended Task Input/Output Table (XTIOT) instead of the TIOT for certain dataset types. There is no effective limit to the number of datasets that you can allocate using the XTIOT.

Using TIOT and XTIOT with datasets

- Every dataset uses either TIOT or the XTIOT, never both.
- You decide whether to use the TIOT or the XTIOT on a per-dataset basis.
- The TIOT is always used for datasets:
 - Specified in the JCL through DD statements.
 - That are dynamically allocated as sequential or VSAM: that is, are not allocated as Model 204 files.
- In a RESTART recovery run, each dataset that was dynamically allocated in the original run is reallocated, unless the NODYNAM option was specified. Reallocation always uses the table, either TIOT or XTIOT, that was used for that dataset in the original run.

Using TIOT and XTIOT with dynamically allocated Model 204 files

- Either the TIOT or the XTIOT option can be used for dynamically allocated Model 204 files.
- Options XTIOT and TIOT were added to the DEFINE and ALLOCATE commands.
- You can specify the TIOT option or the XTIOT option for only Model 204 files with the OLD and DIRECT options on an ALLOCATE or DEFINE DATASET command, for example:

```
ALLOCATE DATASET PAYROLL WITH SCOPE=SYSTEM  
DSN=A. B. C. OLD DIRECT XTIOT
```

- If the TIOT option is specified on the ALLOCATE command, the TIOT is used.
- If the XTIOT option is specified on the ALLOCATE command, the XTIOT is used.
- The new SYSOPT2 parameter setting, X'80', was added to govern default behavior of dynamic allocation, when you do not specify the TIOT or XTIOT option in the ALLOCATE or DEFINE DATASET command. See "SYSOPT2: System options" on page 1-92 for settings and behavior.

If neither the TIOT nor the XTIO option is specified on the ALLOCATE command, SYSOPT2 determines whether the TIOT or XTIO is used. If SYSOPT2=X'80' is set, then the XTIO option is used for only Model 204 file allocations. Otherwise, the TIOT option is used.

- Use of the XTIO requires use of IOS Branch Entry in the run, where XMEMOPT=X'02' is set in CCAIN. If IOS Branch Entry is not active and XTIO is specified, the following error is generated when the allocation is attempted and fails.

```
M204. 2581 XMEMOPT=2 (IOS BRANCH) REQUIRED FOR
XTI OT OPTI ON
```

Special considerations for obsolete form of ALLOCATE

Special considerations apply when the following obsolete form of the ALLOCATE command is used:

Syntax ALLOCATE DDNAME DSNAME

Where:

- The TIOT and XTIO options cannot be specified.
- If the DDNAME starts with TAPE or OUT, sequential organization is assumed; the TIOT option is used.

Otherwise, a Model 204 file is assumed; either the TIOT or the XTIO is used, depending on the setting of SYSOPT2.

User Language enhancements

Defining common variables

You can now define common variables at the beginning of all programs without later redefinitions. The new syntax lets you use the '%VAR IS COMMON' definition without duplicating the previous attributes.

If, however, any attributes, such as STRING or LEN, are included in the redefinition which differ from those previously defined, a compilation error occurs. In addition, if the variable is not previously defined, it is allocated based on the current default variable definition.

For example,

```
BEGIN
%X IS STRING LEN 3 INITIAL ('AAA') STATIC COMMON
SUBROUTINE A
* FULL DEFINITION ON NEXT LINE NO LONGER REQUIRED
** %X IS STRING LEN 3 INITIAL ('AAA') STATIC COMMON
* NEW SYNTAX ON NEXT LINE REPLACE PREVIOUS LINE
%X IS COMMON           /? DEFAULTS TO PREVIOUS %X DEFINITION ?/
CALL B
END SUBROUTINE

SUBROUTINE B
* BUT NEXT LINE WILL FAIL COMPILATION SINCE %X ALREADY EXISTS
%X IS STRING LEN 4 COMMON           /? compiler error ?/
* AND THE NEXT LINE WILL RESULT IN A DEFAULT DEFINITION
* SINCE %Y IS NOT PREVIOUSLY DEFINED.
%Y IS COMMON
END SUBROUTINE
PRINT %X
CALL A
END
```

VALUE IN clause processing

Beginning in Version 5.1, a VALUE IN clause was added to the following User Language statements:

- FIND ALL RECORDS
- FIND ALL VALUES
- FOR EACH RECORD WHERE

The VALUE IN clause is valid for fields that include the ORDERED attribute. The interprocessing between fields works between fields of the same ORDERED type: ORDERED CHARACTER or ORDERED NUMERIC. If processing between fields of differing types, you will likely not find records.

Rewriting applications to take advantage of VALUE IN clause processing

If your User Language applications now use nested loops to perform join processing, you could rewrite your applications to take advantage of the VALUE IN clause.

You are likely to see performance improvements, which result in a reduction of overall wall clock time due to:

- A decrease in disk I/O
- Decreases in the number of FINDs and BXFINDs, because the processing has moved. You will see an increase in BXNEXTs

Using a FIND ALL RECORDS statement

The FIND ALL RECORDS statement accepts a VALUE IN clause as an input condition. The complete syntax for a FIND ALL RECORDS statement in User Language is as follows:

Syntax

```
[ IN { FILE filename | GROUP groupname } ] -  
FIND [ AND RESERVE ] [ ALL ] RECORDS -  
[ IN label | ON [ LIST ] listname ] -  
FOR WHICH | WITH retrieval - conditions  
. . . . .  
END FIND
```

Where

- The *retrieval-conditions* can be:

```
[ fieldname { EQ | NE } VALUE IN value_set ]
```
- The *value_set* label represents an existing value set. Values of the value set in the EQ VALUE IN *value_set* clause are treated as boolean OR, for example:

```
field = value1 OR field = value2 OR. . . .
```
- The NE VALUE IN *value_set* clause is likewise treated as boolean AND, for example:

```
field <> value1 AND field <> value2 AND. . . .
```

IFAM syntax

The format in IFAM is as follows:

```
IFFIND  
  { IN { FILE filename; | GROUP groupname; } }  
  FD;  
  [ fieldname { EQ | NE } VALUE IN value_set ];  
  . . . . .  
END;
```

Coding example for an FD statement

```
OPEN CLAIMS90
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
OPEN CLIENTS
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
BEGIN
/? FIND ALL CLIENTS WHO MADE CLAIMS ?/
CLAIM.POLICYS: IN CLAIMS90 FDV POLICY NO
CLIENT.FD: IN CLIENTS FD POLICY NO EQ VALUE IN CLAIM.POLICYS
    END FIND
FR CLIENT.FD
    PRINT POLICY NO AND FULLNAME
    END FOR
END
```

Using a FIND ALL VALUES statement

The FIND ALL VALUES statement accepts a VALUE IN clause as an input condition. The complete syntax for a FIND ALL VALUES statement format in User Language is as follows:

Syntax [IN {FILE filename | GROUP groupname}] -
FIND [ALL] VALUES OF fieldname -
[FROM value1] [TO value2] -
[(NOT) LIKE pattern] -
[(NOT) VALUE IN value_set]

Where

- The *value_set* label represents an existing value set. Values of the value set in the EQ VALUE IN *value_set* clause are treated as boolean OR, for example:

field = value1 OR field = value2 OR. . .

- The NE VALUE IN *value_set* clause is likewise treated as boolean AND, for example:

field <> value1 AND field <> value2 AND. . .

IFAM syntax An example FIND ALL VALUES statement format in IFAM is as follows:

```
IFFDV
  {IN {FILE filename: | GROUP groupname; }}
  FDV fieldname; [FROM value1] [TO value2];
  [(NOT) LIKE 'pattern'];
  [(NOT) VALUE IN value_set];
END;
```

Coding example for FDV statement

```
OPEN CLAIMS89
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
OPEN CLAIMS90
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
BEGIN
/? FIND ALL POLICYS THAT HAD CLAIMS IN 1989 AND 1990 ?/
CLAIMS89: IN CLAIMS89 FDV POLICY NO
CLAIMS90: IN CLAIMS90 FDV POLICY NO VALUE IN CLAIMS89
FDCLAIMS90: IN CLAIMS90 FD FOR WHICH POLICY NO EQ -
          VALUE IN CLAIMS90
END FIND
COUNT90: COUNT RECORDS IN FDCLAIMS90
IF COUNT IN COUNT90 THEN
FR FDCLAIMS90
PRINT POLICY NO AND VIN INVOLVED AND DRIVER INVOLVED
END FOR
ELSE
PRINT 'NO CLAIMS WERE MADE FOR THE SAME POLICY IN 1989 AND 1990'
END IF
END
```

Using a FOR EACH RECORD WHERE statement

The FOR EACH RECORD WHERE statement accepts a VALUE IN clause as an input condition. The complete syntax for a FOR EACH RECORD WHERE statement in User Language is as follows:

Syntax

```
FOR {EACH | n} {RECORD | RECORDS} -
[IN label | ON [LIST] listname] -
[IN [ASCENDING | DESCENDING] [SORTKEY] ORDER -
[BY [EACH] fieldname] -
[FROM value1] [TO value2]
[BY {%variable | literal}] -
[OPTIMIZING FNV]
[ {WHERE | WITH} retrieval - conditions
```

Where

- The *retrieval-conditions* can be:

```
[fieldname {EQ | NE} VALUE IN value_set]
```
- The *value_set* label represents an existing value set. Values of the value set in the EQ VALUE IN *value_set* clause are treated as boolean OR, for example:

```
field = value1 OR field = value2 OR...
```

- The NE VALUE IN *value_set* clause is likewise treated as boolean AND, for example:

```
field <> value1 AND field <> value2 AND. . .
```

Coding example for an FR statement

```
OPEN CLIENTS
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
OPEN CLAIMS90
FILEMGR
REDEFINE FIELD POLICY NO (ORD CHAR)
BEGIN
/? FIND ALL CLIENTS THAT HAD CLAIMS IN 1990 ?/
CLAIMS. POLICYS: IN CLAIMS90 FDV POLICY NO
IN CLIENTS FR WHERE POLICY NO EQ VALUE IN CLAIMS. POLICYS
PRINT POLICY NO AND FULLNAME
END FOR
END
CLOSE ALL
```

IN GROUP MEMBER clause enhancement

Starting in Version 5.1, the IN GROUP MEMBER clause includes a FOR RECORD NUMBER (FRN) option.

Syntax IN [TEMPORARY | PERMANENT] GROUP *groupname*
 MEMBER *filename statement*

Where:

- *groupname* is the name of the file group.
- *filename* is a valid %variable, image, or screen item that indicates the group member for the CLEAR LIST, FIND, FOR RECORD NUMBER (FRN), or STORE RECORD statement.

Note: The FOR RECORD NUMBER option does not support remote files, unlike the other statements that can be preceded by IN GROUP.

- *statement* is one of the following:
 - CLEAR LIST
 - FIND ALL RECORDS
 - FIND ALL VALUES
 - FOR RECORD NUMBER (FRN)
 - STORE RECORD

Example BEGIN

```
%GRPMEMBER=' AUGUST'  
FD: IN GROUP YR2001 MEMBER %GRPMEMBER FRN 164  
PRINT FULLNAME WITH DOB AT 35 WITH SSN AT 50  
END FOR  
END
```


Functions introduced or updated in Version 5.1

\$CHKPINF function

Function

Returns information about checkpoints. It takes one argument, which specifies what information to return.

Syntax

`$CHKPINF(request - code)`

Where:

request-code	Asking to return...	\$CHKPINF returns...	Meaning that checkpointing is...
0	Checkpoint status	0	Not active
		1	Currently being taken (by CPTIME or by CHECKPOINT command)
		2	Active, but no auto checkpoints; CPTIME=0
		3	Active and CPTIME does not equal 0
1	Date-time for the next checkpoint attempt, scheduled by the CPTIME argument	9999/99/99 99:99:99.99	Not active
		9999/99/99 99:99:99.99	CPTIME=0
		Current date-time as <i>yyyy/mm/dd hh:mm:ss.hh</i>	In progress
		<i>yyyy/mm/dd hh:mm:ss.hh</i>	Scheduled date-time
2	Seconds till the next checkpoint attempt, scheduled by the CPTIME parameter	999999999	Not active
		999999999	CPTIME=0
		0	In progress
		<i>sss.hh</i>	Scheduled in this many seconds
3	Time of last successful checkpoint	Date-time of last checkpoint as <i>yyyy/mm/dd hh:mm:ss.hh</i>	Due to CPTIME parameter or CHECKPOINT command
		0000/00/00 00:00:00.00	Not active

request-code	Asking to return...	\$CHKPINF returns...	Meaning that checkpointing is...
4	Seconds since the last successful checkpoint	Time in seconds of checkpoint as <i>sss.hh</i>	Due to CPTIME parameter or the CHECKPOINT command
		999999999	Not active
5	Total number of records currently in CHKPOINT stream, which includes checkpoints and preimages	Number of records	
6	Number of checkpoints currently in CHKPOINT stream	Number of checkpoints	
7	Extended quiesce status	0	Extended quiesce inactive for this run
		1	Extended quiesce unset: will not be entered
		2	Extended quiesce set; at end of next successful checkpoint extended quiesce state will be reentered
		3	Currently in extended quiesce
		4	Extended quiesce facility non-functional: in EOJ

Example

The following procedure illustrates using the \$CHKPINF function.

```

* * * TOP OF PROCEDURE * * *
BEGIN
PRINT $CHKPINF(0) WITH ' CURRENT CHECKPOINTING STATUS' AT 25
PRINT $CHKPINF(1) WITH ' NEXT SCHEDULED CHECKPOINT' AT 25
PRINT $CHKPINF(2) WITH ' SECONDS UNTIL NEXT SCHEDULED CHECKPOINT'
AT 25
PRINT $CHKPINF(3) WITH ' LAST SUCCESSFUL CHECKPOINT TAKEN' AT 25
PRINT $CHKPINF(4) WITH ' SECONDS SINCE LAST SUCCESSFUL CHECK-
POINT' AT 25
PRINT $CHKPINF(5) WITH ' # RECORDS CURRENTLY IN CHKPOINT STREAM'
AT 25
PRINT $CHKPINF(6) WITH ' # CHECKPOINTS IN CHKPOINT STREAM' AT 25
PRINT $CHKPINF(7) WITH ' EXTENDED QUIESCE STATUS' AT 25
END
* * * BOTTOM OF PROCEDURE * * *

```

Model 204 displays the following output:

```
3          CURRENT CHECKPOINTING STATUS
2001/11/19 12: 40: 00. 75  NEXT SCHEDULED CHECKPOINT
249. 68          SECONDS UNTIL NEXT SCHEDULED CHECKPOINT
2001/11/19 12: 20: 00. 66  LAST SUCCESSFUL CHECKPOINT TAKEN
950. 42          SECONDS SINCE LAST SUCCESSFUL CHECKPOINT
7083          # RECORDS CURRENTLY IN CHKPOINT STREAM
1             # CHECKPOINTS IN CHKPOINT STREAM
1             EXTENDED QUIESCE STATUS
```

\$ECBDGET function

Function

Get string data associated with an Event Control Block (ECB)

Syntax

```
$ECBDGET( ECB- number | ' CPQZ' )
```

Where:

- *ECB-number* is a string with a numeric value from one to the NECBS parameter that identifies the ECB from which to retrieve the text. The *ECB-number* can be expressed as a numeric literal, a %variable, or a field name.
- CPQZ is a named ECB used by the NonStop/204 facility to automatically post an extended quiesce. See “NonStop/204” on page 1-18 for an explanation of the facility.

To use CPQZ, you need not set the NECBS parameter. CPQZ can be expressed as a literal, a %variable, or a field name. When CPQZ is specified, the value of the \$ECBDGET function can be non-null (except for error values) during only extended quiesce. CPQZ is internally cleared to null when the system exits extended quiesce.

Usage

When the \$ECBDGET function is successful, it returns your data as a text string up to 255 bytes long. If you make a coding mistake, you may receive one of the following return codes as a string.

Return code	Meaning
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The input argument is less than one or greater than the NECBS parameter

Return code	Meaning
5	NUSERS = 1
8	No argument specified
9	Checkpointing is inactive, if using extended quiesce ECB named CPQZ
12	Invalid argument QZSIG specified.

The \$ECBDGET function retrieves data set by either the \$ECBDSET or \$POST functions for the specified ECB.

Example

```
%X = $ECBDGET( 17)
```

\$ECBDSET function

Function

Set string data associated with an Event Control Block (ECB)

Syntax

```
$ECBDSET( { ECB- number | ' CPQZ' }, { ' string' } )
```

Where:

- *ECB-number* is a string with a numeric value from one to the NECBS parameter that identifies the ECB in which to store the string of data. The *ECB-number* can be expressed as a numeric literal, a %variable, or a field name.
- CPQZ is a named ECB used by the NonStop/204 facility to automatically post an extended quiesce. See “NonStop/204” on page 1-18 for an explanation of the facility.

To use CPQZ, you need not set the NECBS parameter. CPQZ can be expressed as a literal, a %variable, or a field name. When CPQZ is specified, the value of the \$ECBDSET function can be non-null (except for error values) during only extended quiesce. CPQZ is internally cleared to null when the system exits extended quiesce.

- *string* can be up to 255 bytes long. It can be a numeric, a literal, a %variable, or a field name.

Unless explicitly reset to null, data strings persist whether or not the ECB is posted. Depending on the sequence, data strings can be changed by either the \$POST or \$ECBDSET functions.

Usage

The \$ECBDSET function returns the following return codes:

Return code	Meaning
0	Success
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The first argument is less than one or greater than the NECBS parameter
5	NUSERS = 1
7	Missing data argument
8	No argument specified
9	Checkpointing inactive, if using extended quiesce ECBs, CPQZ or QZSIG
11	Cannot be issued after QZSIG has been posted
12	Invalid argument QZSIG specified
13	Cannot be issued outside of extended quiesce

Note: The \$ECBDSET function associates a string with an ECB, regardless of whether the ECB is posted or not. String data set by \$ECBDSET is accessible using the \$ECBDGET function.

Example

```
%X=$ECBDSET(1, ' This is about managing user threads')
```

\$ECBTEST function

Function

Check an Event Control Block (ECB) to see if it is posted

Syntax

```
$ECBTEST( ECB- number | ' CPQZ' | ' QZSI G' )
```

Where:

- *ECB-number* is a string with a numeric value from one to the NECBS parameter that identifies which ECB to test for its post status. The *ECB-number* can be expressed as a numeric literal, a %variable, or a field name.

- CPQZ is a named ECB used by the NonStop/204 facility to automatically post an extended quiesce. See “NonStop/204” on page 1-18 for an explanation of the facility. To use CPQZ, you need not set the NECBS parameter. CPQZ can be expressed as a literal, a %variable, or a field name.
- QZSIG is a named ECB used by the NonStop/204 facility to signal when an external backup is completed. See “NonStop/204” on page 1-18 for an explanation of the facility. To use QZSIG, you need not set the NECBS parameter. QZSIG can be expressed as a literal, a %variable, or a field name.

Usage

Use the \$ECBTEST function to obtain ECB status, posted or not, through the return code. The post code, if set by \$POST, is accessible using the \$STATUSD function. The following return codes apply to the \$ECBTEST function:

Return code	Meaning
0	Not posted
1	Posted
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The first argument is less than one or greater than the NECBS parameter
5	NUSERS = 1
8	No argument specified
9	Checkpointing inactive, if using extended quiesce named ECBs, CPQZ or QZSIG

Example

```
%RC=$ECBTEST( 1)
```

\$ERRCLR function

Function Clears the error message text returned by the \$ERRMSG and \$FSTERR functions. The \$ERRCLR function takes no argument.

Example %DUMMY=\$ERRCLR

\$POST function

Function

Indicates that some event has occurred. The thread(s) waiting on the Event Control Block (ECB) can resume processing.

Syntax

```
$POST( { ECB- number [ , post - code ] [ , string ] | ' QZSI G' } )
```

Where:

- *ECB-number* is a string with a numeric value from one to the NECBS parameter that identifies the ECB to be posted. The *ECB-number* can be expressed as a numeric literal, a %variable, or a field name.
- *post-code* is a numeric value between zero and 16,777,215. You can use this optional argument for whatever you wish. If omitted, a default post code of zero is used.

The post code is accessible using the \$ECBTEST function followed by the \$STATUSD function for the specified posted ECB.

Once set, unless explicitly reset to zero, post codes persist whether or not the ECB is posted.

- *string* can be up to 255 bytes long. It can be a numeric, a literal enclosed in quotation marks, a %variable, or a field name.

String data is accessible using the \$ECBDGET function for the specified ECB.

Once set, unless explicitly reset to null, data strings persist whether or not the ECB is posted. Depending on the sequence, data strings can be changed by either the \$POST or \$ECBDSET functions.

- QZSIG is an extended quiesce named ECB. To use it, you need not set the NECBS parameter. QZSIG can be expressed as a literal, a %variable, or a field name. Although QZSIG can be posted, it cannot be unposted. Unposting is handled internally at the end of the extended quiesce. You cannot specify a post code or a string for QZSIG.

Usage

The \$POST function posts a specified ECB with an optional post code and an optional associated string. Only an unposted ECB may be posted. If an ECB is already posted, the posted state is indicated by return code 14. The ECB contents and associated string do not change. Any other thread waiting (\$WAIT) on the specified ECB becomes eligible to resume processing.

Use the \$POST, \$UNPOST, and \$WAIT functions to coordinate processing between threads.

The following return codes apply to the \$POST function

Return code	Meaning
0	Success
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The first argument is less than one or greater than the NECBS parameter
5	NUSERS = 1
8	No argument specified
9	Checkpointing inactive
10	Post code out of range, 0 to 16,777,215
11	The QZSIG ECB is already posted
12	Invalid argument CPQZ specified or invalid argument following QZSIG
13	Cannot be issued outside of extended quiesce
14	An attempt to post an already posted ECB

Example

On Thread A

```
BEGIN  
PRINT $UNPOST( 1)  
PRINT $WAIT( 1)  
PRINT $STATUSD  
END
```

On Thread B

```
BEGIN  
PRINT $POST( 1, 32)  
END
```

In the previous code, the \$UNPOST function on Thread A marks the ECB specified as unposted. Then, Thread A waits at the \$WAIT(1) until another thread issues a \$POST(1,nnn). Thread B posts the ECB number one with \$POST(1,32), at which point Thread A resumes evaluation with PRINT \$STATUSD and prints the post code of 32.

\$STATUS and \$STATUSD

The \$STATUS and \$STATUSD functions can be used after an External Call Facility statement to return information about the success or otherwise of an

ECF operation. For more information on ECF return codes see "Return codes" on page 3-11.

\$STATUSR function

Function

Resets the value of \$STATUS to zero and returns a value of zero.

Syntax

\$STATUSR

Usage

The \$STATUSR function accepts no arguments.

Programs that use a DO WHILE \$STATUS=0 loop can use \$STATUSR within the loop to process a nonzero \$STATUS and continue looping.

\$UNPOST function

Function

Resets a specified Event Control Block (ECB) to an unposted state, meaning that the event has not yet occurred or has not recurred. Resets the post code to zero.

Syntax

\$UNPOST(ECB- number)

Where:

ECB-number is a numeric value from one to the NECBS parameter that identifies the ECB to be unposted. The *ECB-number* can be expressed as a numeric literal, a literal enclosed in quotation marks, a %variable, or a field name.

Usage

Use the \$UNPOST function to unpost a specified, numbered ECB. An ECB must be unposted before another post or wait on this ECB takes place. The \$POST and \$WAIT functions do not unpost an ECB, so that a subsequent wait on the same ECB cannot take place, because the ECB is still posted from the previous posting.

Use the \$POST, \$UNPOST, and \$WAIT functions to coordinate processing between threads for numbered ECBs. For the extended quiesce ECB, QZSIG, only the \$WAIT and \$POST functions are valid.

Note: Even in a non-multiprocessing environment, the \$UNPOST function should be used with extreme care. Unposting an ECB while users are waiting on it may keep users in a wait state forever or until the next \$POST function is issued.

\$UNPOST does not reset the contents of the post code or string data set by the \$POST or \$ECBDSET functions for the specified ECB.

The following return codes apply to the \$UNPOST function:

Return code	Meaning
0	Success
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The first argument is less than one or greater than the NECBS parameter
5	NUSERS = 1
8	No argument specified
12	Invalid argument CPQZ or QZSIG specified

\$VALIDATE_NUMERIC_DATA function

Function

Detects invalid numeric data when data is read into an image from an external source.

Syntax

```
$VALIDATE_NUMERIC_DATA(' %variable'
| ' %imagename:itemname' , s1, s2, s3, s4, s5)
```

Where:

- *%variable* is a string expression enclosed in single quotation marks that resolves to the name of an item in an image. The value of this string is the standard format for referencing an image item: *%imagename:itemname*, which is also enclosed in single quotation marks.
- If the argument identifies an array item, up to five subscripts can be used, *s1-s5*. If an array is not specified, these arguments are ignored. On an array, as many of these parameters are used as the number of dimensions

on the array. If any needed subscript parameter is not specified, then a value of 1 is used.

Example

If *%imagenam:itemname* identifies an array, then you must identify the particular array element to validate. To do so, note the number of subscripts of the array, and specify the same number of optional subscript arguments (the s1–s5 arguments). For example, to validate %IMAGE:ARRAY(1,2), code

```
%RC=$VALI DATE_NUMERI C_DATA( ' %I MAGE: ARRAY' , 1, 2)
```

Usage

The image item whose name is passed is looked up in the image definition.

- If the item is a numeric type, the contents of the data area of the image are checked to see if the data that corresponds to the item is in the correct format.
- If the item is an array, the element identified by the subscripts is tested.

The numeric return code values have the following meanings:

Return code	Means...
0	Return code zero indicates good data. Either item is numeric and the data is of the correct format, or the item is not numeric (no validation performed).
1	Item is numeric and the data is invalid.
2	On an array element, at least one subscript is out-of-bounds.
3	Item named does not exist.

\$WAIT function

Function

Suspend a user until an Event Control Block (ECB) is posted

Syntax

```
$WAI T( ECB- number [ , ' SWAP' | ' NOSWAP' ]  

       [ , time- interval ] | ' CPQZ' | ' QZSI G' )
```

Where:

- *ECB-number* is a string with a numeric value from one to the NECBS parameter, which identifies the ECB upon which to wait. The *ECB-number*

can be expressed as a numeric literal, a %variable, or a field name. This is a user defined event. When these users are placed in a wait state, the wait type is 30.

- SWAP keyword specifies a swappable wait; this is the default. The NOSWAP keyword specifies a nonswappable wait. If you enter a keyword, enclose it in single quotation marks.
- *time-interval* specifies the maximum number of seconds to wait. A wait that is finished, because the time expired, is indicated by a return code 16.

A *time-interval* may be from zero to 86400 and indicated as a number, %variable, or a string. A *time-interval* less than one is treated as no time-interval supplied with no time-out to happen.

If a *time-interval* is specified, the second argument, SWAP or NOSWAP, is activated. You can specify SWAP or NOSWAP, or accept the default, SWAP, using the following syntax:

```
$WAIT( ECB- number , , t i m e- i n t e r v a l )
```

The *time-interval* can be used as an argument to \$WAIT to determine when an extended quiesce event starts. At that time, the backup can be submitted. When these users are placed in a wait state, the wait type is 47.

- The CPQZ and QZSIG keywords are the named ECBs for extended quiesce used by the NonStop/204 facility. See “NonStop/204” on page 1-18 for an explanation of the facility.
 - CPQZ is posted internally at the beginning of each extended quiesce and unposted internally at the interval end.
 - QZSIG can be posted during the extended quiesce. It is unposted internally at the end of extended quiesce.

The \$WAIT function applied to CPQZ and QZSIG are bumpable, swappable waits of type 47 for CPQZ and 48 for QZSIG. You cannot specify the SWAP or NOSWAP keywords or *time-interval* with the named ECBs.

Usage

You can use the \$WAIT function to suspend a user, meaning: put that user into a wait state until the ECB is posted by another user with the \$POST function. Users who have issued a \$WAIT function call are bumpable and may be swappable depending on whether SWAP or NOSWAP was used in the \$WAIT call.

Caution: Limit the use of the \$WAIT function with the NOSWAP option to situations where only a small number of threads may use it. This will avoid having all servers occupied by users in a NOSWAP state and having no available server for a posting user to swap into.

Using the \$WAIT function, you can put a User Language thread into a wait state. To perform third-party backups, the thread must wait for the extended quiesce of a checkpoint to start and then submit a backup job.

- For numbered ECBs, you can use the \$POST, \$UNPOST, and \$WAIT functions to coordinate processing between threads.
- For the QZSIG ECB, you can use the \$POST and \$WAIT functions to signal and recognize the end of an extended quiesce for third-party backups.
- For the CPQZ ECB, you can use the \$WAIT function to wait on the start of an extended quiesce.

The following return codes apply to the \$WAIT function:

Return code	Meaning
0	Success
2	Bad argument specified
3	NECBS parameter is not specified or is zero
4	The first argument is less than one or greater than the NECBS parameter
5	NUSERS = 1
6	NOSWAP and NSERVS EQ 1
8	No argument specified
9	Checkpointing inactive, if using extended quiesce ECBs, CPQZ or QZSIG
11	The CPQZ or QZSIG ECB is already posted
12	Invalid argument following CPQZ or QZSIG specified
13	For QZSIG, the system is not in extended quiesce or already leaving extended quiesce
15	Time interval is not numeric or greater than 86,400
16	\$WAIT finished due to expired time interval
17	Second argument entered is not SWAP or NOSWAP

When the ECB specified in the \$WAIT call is posted, the waiting user will resume evaluation and may capture the post code with \$STATUSD. See the following \$WAIT example.

Example

The following code illustrates an interaction between User 1 and User 2. User 1 issues the following:

```
BEGIN
  %X=$ECBDSET(3, ' THIS IS ECB 3' )
  PAUSE 60
  %X=$POST(3, 5678)
END
```

User 2 starting after the previous \$ECBDSET, but before 60 seconds have elapsed:

```
BEGIN
PRINT ' WAITING ON ECB 3'
NP
%X=$WAIT(3)
PRINT ' ECB 3 HAS BEEN POSTED WITH POST CODE= ' WITH -
  $STATUSD
END
```

User 2 will be suspended (WT=30) when the \$WAIT call is evaluated and will resume processing when ECB number 3 is posted by User 1. \$STATUSD will return the post code value = 5678.

If User 2 starts after 60 seconds have elapsed and the ECB number 3 has been posted, then User 2 will not wait, but will print the last post code for the ECB number 3.

Storing invalid numeric data: FLOAT or ORDERED NUMERIC

Prior to Version 5.1, Model 204 did not enforce storage rules for floating point numbers outside the valid range and could incorrectly store invalid floating point data as small, but nonzero floating point numbers.

The valid range of floating point numbers is:

1E-74 through 7.2370E+75

- 7.2371E75 is outside the valid range for floating point numbers. Prior to Version 5.1, it was stored as a true floating point number of the order of E-78 following the error message:

M204. 0563: ARI THMETI C OVERFLOW

- 1E-75 is also out of the valid range; it was stored as a true floating point number 1E-75.

As of Version 5.1:

- Storing the value 7.2371E75 in a FLOAT field does not result in error M204. 0563: ARI THMETI C OVERFLOW. The value is treated as an invalid floating point number; it is stored as a data string '7.2371E75'.
- 1E-75 is treated as an invalid floating point number and stored as a data string '1E-75'.

If an invalid value is stored for a field with the ORDERED NUMERIC attributes, the value stored in the Ordered Index is the data string. That data string is stored on the Invalid Numeric Data portion of the Ordered Index. This changes the output order of these values when the ordering is done against the Ordered Index, for example, when using the User Language IN ORDER BY or FOR EACH VALUE clauses.

Identifying pre-Version 5.1 floating point values

The changed storage of invalid floating point values affects only new values you store in Version 5.1. Values already stored in the file are not corrected. You must find and correct these values, if necessary. The following procedure may help you, if the field is also defined as ORDERED NUMERIC or FRV.

Floating point ordered numeric values outside the valid range may be explicitly identified, removed, and replaced with valid values. This process can be done either before or after migrating to Version 5.1. Migrating itself, without rebuilding files, does not remove the bad values; nor does issuing a REORGANIZE ORDERED INDEX command against the file, regardless of the release version of the REORGANIZE command.

It is safer to rebuild under Version 5.1, because invalid values are inserted into the Invalid Numeric section of the Ordered Index. If you rebuild under a prior version, the only safeguard against improper reinsertion of invalid values is their explicit identification and removal prior to the rebuild process. You can

confirm the presence of invalid values by running a procedure such as the following. Note the following fields.

- problem_field

For each occurrence of problem_field substitute a Model 204 field name, which you suspect stores invalid floating point values. problem_field must be an ORDERED NUMERIC or FRV field, which is already defined to the file.

- TEMP_FIELD

A Model 204 field name of an ORDERED CHARACTER field defined for use in only the following procedure.

```
DEFINE TEMP_FIELD (ORD CHAR)
BEGIN
STORE RECORD
* MAKE SURE THAT THERE IS A TRUE FLOATING PT 0
* FOR THE SUSPECTED FIELD
problem_field = 0
TEMP_FIELD = X
END STORE
END
```

```
BEGIN
FDV: FIND ALL VALUES OF problem_field
FR: FOR EACH VALUE IN FDV
%X = VALUE IN FR
  IF %X = 0 THEN
    %COUNT_OF_ZEROS = %COUNT_OF_ZEROS + 1
  END IF
END FOR
  IF %COUNT_OF_ZEROS GT 1 THEN
    PRINT ' FIELD CONTAINS VALUES OUT OF RANGE'
  END IF
END
BEGIN
FD: FD
  TEMP_FIELD = X
END FIND
FR IN FD
* REMOVE THE DUMMY RECORD YOU ADDED TO THE FILE
DELETE RECORD
END FOR
END
DELETE FIELD TEMP_FIELD
```

Note: The previous procedure can find floating point values out of range only if the field is FRV or ORDERED. CCA recommends that you add the ORDERED attribute to use the previous procedure.

SQL processing floating point numbers

SQL processing of floating point values greater than the largest valid value has changed:

- An INSERT statement results in:

```
SQL error - 103. ' 7. 237E75 is an invalid numeric
literal. '
```

- A SELECT statement returns an SQL warning with:

```
' Invalid data was skipped by the SQL Engine. '
```

- A SELECT DISTINCT statement retrieves no data for such a value.

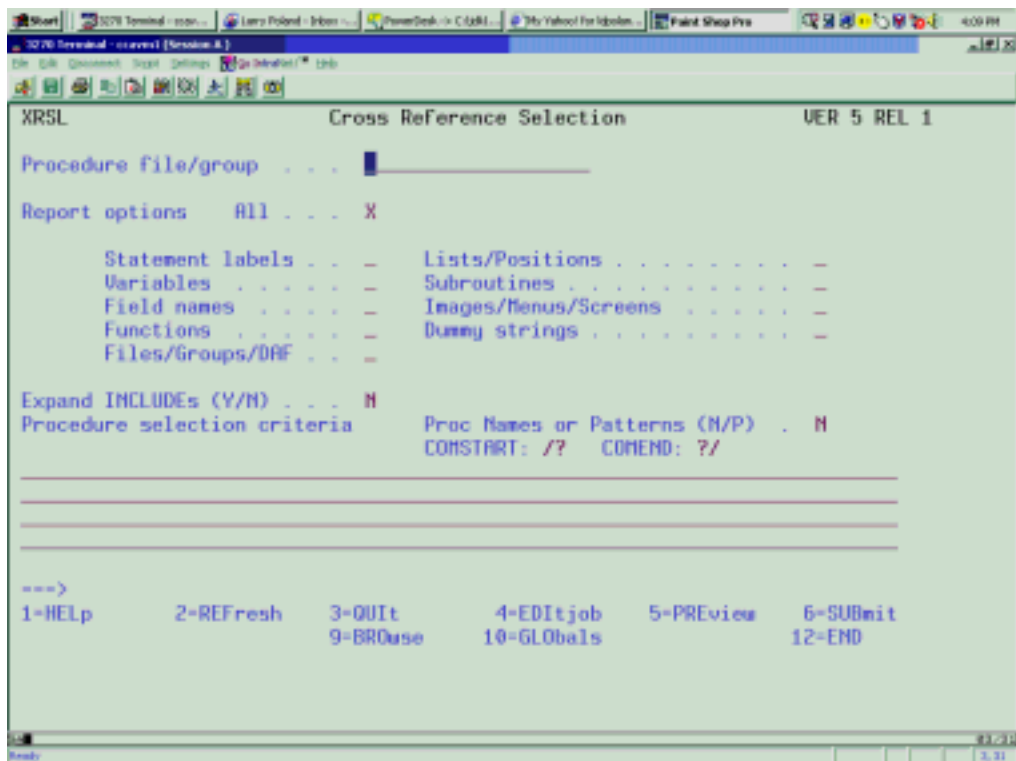
XREF support for Block Comments

The XREF facility of Dictionary was enhanced to add support for User Language Block Comments in your procedures.

In Version 5.1, when each user enters the XREF screen for the first time, the user must enter the COMSTART and COMEND characters that designate a Block Comment in procedures that a user is going to process.

A profile record is stored for each user that contains the last COMSTART and COMEND characters the user entered on the screen. On a subsequent entry into XREF, those COMSTART and COMEND characters are displayed on the screen. If you type over the displayed characters, the new characters are stored in your profile record. Figure 1-1 displays the Cross Reference Selection screen with the COMSTART and COMEND fields.

Figure 1-1. Cross Reference Selection screen



Performance enhancements

General improvements

The following list of improvements and enhancements may bring CPU and/or I/O reduction to your site.

- Event Definition Blocks have been moved above the 16M-line.
- The PAI INTO statement can now run on an MP subtask. This allows more maintask work to proceed and increases the amount of off-loaded processing to subtasks.

Normal PAI processing remains maintask oriented, as there is likely an I/O operation for each field that requires maintask processing. Switching back and forth from maintask to MP subtask for each field is counter productive.

- Blocked-in users are on an invisible long-wait queue, so the scheduler does not have to scan these users each time it has to run.
- Prior to Version 5.1, locating an APSY subsystem procedure in the Auxiliary Procedure Dictionary was done via a sequential search. As of Version 5.1, the procedure is done with a binary search.
- The code path for retrieving values either from STBL, which handles string variables, string array variables, results of string expressions, and string constants, or from absolute address KOMM fields has been reduced about 25 percent. This has been partially offset by the expense of other assignment code paths increasing by four instructions.
- The CPU path length for \$function calls that assign their result to a %variable, and the setting of missing numeric \$function arguments has been reduced. This was done by directly moving the value, instead of having the User Language QUAD processor do the work. The VTBL and QTBL space that was required for the QUAD processor is eliminated.
- The NTBL scans for nonglobal lists, foundsets, and those related to an IDENTIFY statement have been eliminated, which reduces CPU time.
- NTBL scans and CPU time for screen variable look ups are reduced.
- APSY file resources are not obtained for the user, if the subsystem holds them. For nonprecompiled procedures and nonautocommit APSYs this prevents precompile processing from obtaining a file resource on behalf of a user, if the subsystem already holds the file resource on the file.

If a file resource is grabbed in a nonautocommit subsystem because of open nonsubsystem files, these resources are released at the place where an implied commit is done in autocommit subsystems, as long as the file does not have pending updates against it.

- There is a tremendous overhead in doing logical page reads, especially in an MP environment. A large percentage of logical page reads are multiple

repeats for the same pages. By not going through a full close-open path for these pages, DKBM traffic can be avoided.

The concept of pending releases is exploited. This feature scans the open page array. If the requested page is pending release it is reused, by-passing a full open-close path. This technique significantly reduces DKPRs and at the same time reduces CPU and MP lock traffic.

- The number of entries required in the Resource Enqueue Table was reduced. You can monitor the free space in this table with the \$CENQCT function. The number of entries freed by this enhancement is equal to the number of procedures that match the precompile prefix, plus the number of procedures that match the non-precompile prefixes for all started APSY subsystems.

NTBL and QTBL made resident and shareable in virtual storage

You can now cause NTBL and QTBL for APSY precompiled procedures to become resident and shareable in virtual storage using the new RESLTHR parameter. Once made resident in virtual storage these tables no longer need to be loaded into a user's server for evaluation. This feature reduces both CCATEMP and server I/O.

See "RESLTHR: APSY resident load threshold" on page 1-90.

FOR EACH OCCURRENCE loops

Prior to Version 5.1, a FOR EACH OCCURRENCE (FEO) scan of a record in an unlocked record set, one that was established using FDWOL, started the search for each subsequent occurrence at the beginning of the record.

As of Version 5.1, the performance of a FOR EACH OCCURRENCE loop for records in an unlocked foundset was improved using the new parameter, UFEOHASH. When the UFEOHASH parameter is active, each scan of a record for the next occurrence of a field can start from the location of the previous occurrence, if the record has not been changed by another user. If the record has been changed, the record is scanned from the beginning. See "UFEOHASH: Optimization of FOR EACH OCCURRENCE loops" on page 1-95

You can check the value of the UFEOMISS parameter to determine how many times a non-optimized FEO scan occurred. See "UFEOMISS: Number of unsuccessful optimizations for FEO" on page 1-96.

Managing the Record Enqueuing Table

You may activate a new feature that controls how entries for Transaction Back Out pending updates are stored in the Record Enqueuing Table. This feature is activated by setting the MSRES parameter to a nonzero value and can reduce the CPU cost of processing. However, you may need to increase

LRETBL to provide more space in the record enqueueing table for the additional entries stored by this feature.

See “MSRES: Maximum number of Single Record Entries” on page 1-87.

Spinning on an MP lock

In Version 5.1, the MPSYS parameter was introduced to keep a thread spinning on an MP lock even if it appears that the task holding the lock cannot be dispatched. MPSYS=X'01' causes Model 204 to spin, which means loop on an attempt to get an MP lock such as the lock for the:

- Buffer pool
- Record locking table
- LRU queue, and so on

This loop repeatedly tries to acquire the lock by running for MAXSPINS times before issuing an operating system wait. Without MPSYS=1 the pre-Version 5.1 behavior prevails, which immediately issues an operating system wait if the lock is unavailable.

The overhead of issuing an operating system wait is significant. It is usually much higher than the overhead of spinning even 200 times. In most cases, Model 204 does not spin or loop 200 times before the lock becomes available. You will have saved significant overhead compared to issuing an operating system wait.

See “MPSYS: MP/204 processing options” on page 1-87 for more details.

Performance improvement for global string variables

As of Version 5.1, you can increase the speed and reduce the CPU time to find and update a global string variable by setting the new parameter GTBLHASH to a nonzero value. The GTBLHASH parameter specifies the number of buckets allocated in the global string variable section of GTBL. When GTBLHASH is a nonzero value, and you set or get a global string variable, the global string variable name is hashed to determine the bucket in which the name is located.

This reduces the overall amount of data that must be scanned to find a global string variable or must be moved when a value is deleted or changes in size. If GTBLHASH=0, global string variables are processed as in pre-5.1 versions of Model 204. See “GTBLHASH: Number of buckets for global string variables” on page 1-82.

The GTBLPCT parameter determines the initial percentage of GTBL to allocate for global string variables. The default value of GTBLPCT is 50, meaning 50 percent of GTBL is initially allocated for global variable strings. The remainder, in this case 50 percent, is the initial allocation for global objects. However, if GTBLHASH=0, a nonzero setting for GTBLPCT has no effect.

When in effect, if either area of GTBL fills and there are still free pages in GTBL, then GTBL can be rearranged if more space is required in the full area of GTBL. Because these rearrangements can be CPU intensive, CCA recommends that you determine an accurate setting for GTBLPCT to avoid frequent rearrangements. You can monitor the performance of the hash GTBL feature using the GTBLRU user statistic and the GTBLRS since-last statistic. See “GTBLPCT: Initial percentage of GTBL to allocate for global string variables” on page 1-83.

Rearranging GTBL and tracking the rearrangements

The following statistics are available as system statistics, user statistics and since-last statistics to keep track of GTBL rearrangements required for the hashed GTBL feature:

Statistic	Tracking
GTBLRU	Number of GTBL rearrangements required to add a string variable global.
GTBLRS	Number of GTBL rearrangements required to add a global object.

After reviewing the GTBLRU and GTBLRS statistics, you can consider taking the following actions:

- If both of these values are high, increase the size of GTBL by increasing LGTBL.
- If GTBLRU is high but GTBLRS is not, increase GTBLPCT or decrease GTBLHASH.
- If GTBLRS is high but GTBLRU is low, decrease GTBLPCT.

CHKPOINT records enhanced

When a checkpoint is taken, a checkpoint record is written to both the CHKPOINT and CCAJRNL datasets. In Version 5.1, the checkpoint record written to CHKPOINT contains the physical location of the same record just written to CCAJRNL. During ROLL FORWARD processing, this physical location information is used by the NOTE/POINT facility of BSAM to open and position CCARF immediately to the correct checkpoint record. This dramatically reduces the time required for ROLL FORWARD processing.

Note: This is not supported for VSE as IBM does not support the NOTE/POINT facility for VSE.

ROLL BACK processing speed increased

Prior to Version 5.1, writing disk page preimages to database files during the ROLL BACK processing was single threaded; that is, only one page was written at a time. Starting in Version 5.1, writing disk page preimages to database files

during the ROLL BACK processing are overlapped. In most cases, ROLL BACK processing time might decrease by a factor of four or five. For maximum efficacy of this speedup, CCATEMP should have at least 33 pages.

Fast logical page reads

A logical page read (DKPR) is a moderately expensive operation, especially in an MP/204 environment. Often a page is logically read repeatedly; each logical read incurs the same overhead as the previous logical read.

In Version 5.1, logical page reads are optimized; a popular page is kept pending or in deferred release after use. If a page is open when a logical page read is done for it, the logical page read can be done very quickly. This type of logical page read is called a **fast read**, because of the shorter path length.

A fast read is tracked with the DKPRF statistic. A fast page read does not increment the DKPR statistic, which means that after upgrading to Version 5.1, there is a significant drop in DKPRs. The decrease is reflected in the DKPRF statistic.

The relative ratio of fast reads to standard page reads might be improved by increasing the value of the new MAXOBUF parameter and by setting the SCHDOPT X'04' bit as described in "SCHDOPT: MP/204 scheduler operation and accounting" on page 1-90. See also "MAXOBUF: Maximum number of open disk buffers per server" on page 1-85.

Subscripted field extraction

In Version 5.1, subscripted field references attempt to maintain their position inside a record much as an FEO loop attempts to maintain its position inside a record. This means that subscripted field references now tend to be as efficient as FEO loops. It is no longer more efficient to extract multiple fields for a repeating field group via multiple FEO loops.

The following code uses two FEO loops to extract corresponding occurrences for a repeating field group:

```
%INC IS STRING ARRAY (12) NO FS
%I DATE IS STRING ARRAY (12) NO FS
FEOINC: FEO INCIDENT
    %INC(OCCURRENCE IN FEOINC) = VALUE IN FEOINC
    END FOR

FEOI DATE: FEO INCIDENT DATE
    %I DATE(OCCURRENCE IN FEOI DATE) = VALUE IN FEOI DATE
    END FOR
```

The following somewhat more natural coding may actually result in decreased CPU time:

```
%INC IS STRING ARRAY (12) NO FS
%I DATE IS STRING ARRAY (12) NO FS
FEOI DATA: FEO INCIDENT
  %INC(OCCURRENCE IN FEOI DATA) = VALUE IN FEOI DATA
  %I DATE(OCCURRENCE IN FEOI DATA) = INCIDENT
  DATE(OCCURRENCE IN FEOI DATA)
  END FOR
```

Although this enhancement introduces an additional eight bytes for each record variable in VTBL, most procedures should show at least some reduction in VTBL and QTBL size requirements. Still, it is possible that procedures with an extremely large number of sorts or subscripted field references could show an increase in VTBL size. To assess the impact, run any possibly problematic procedures in a test environment and compare the VTBL size requirements under Version 5.1 with the earlier release of Model 204.

SQL processing improvements and changes

SELECT DISTINCT processing

When using a SELECT DISTINCT statement on an SQL column, significant performance improvements result if the underlying Model 204 field is defined as ORDERED or FRV.

The Model 204 SQL engine is specifically designed to handle Model 204 field types. CCA recommends that you define all fields that are used in SQL DISTINCT or WHERE processing with the ORDERED attribute. Also, NULL values will be excluded from the result set.

Figure 1-2 compares the paths taken when fields are *not* ORDERED or *not* FRV with the path taken when fields are defined ORDERED or FRV.

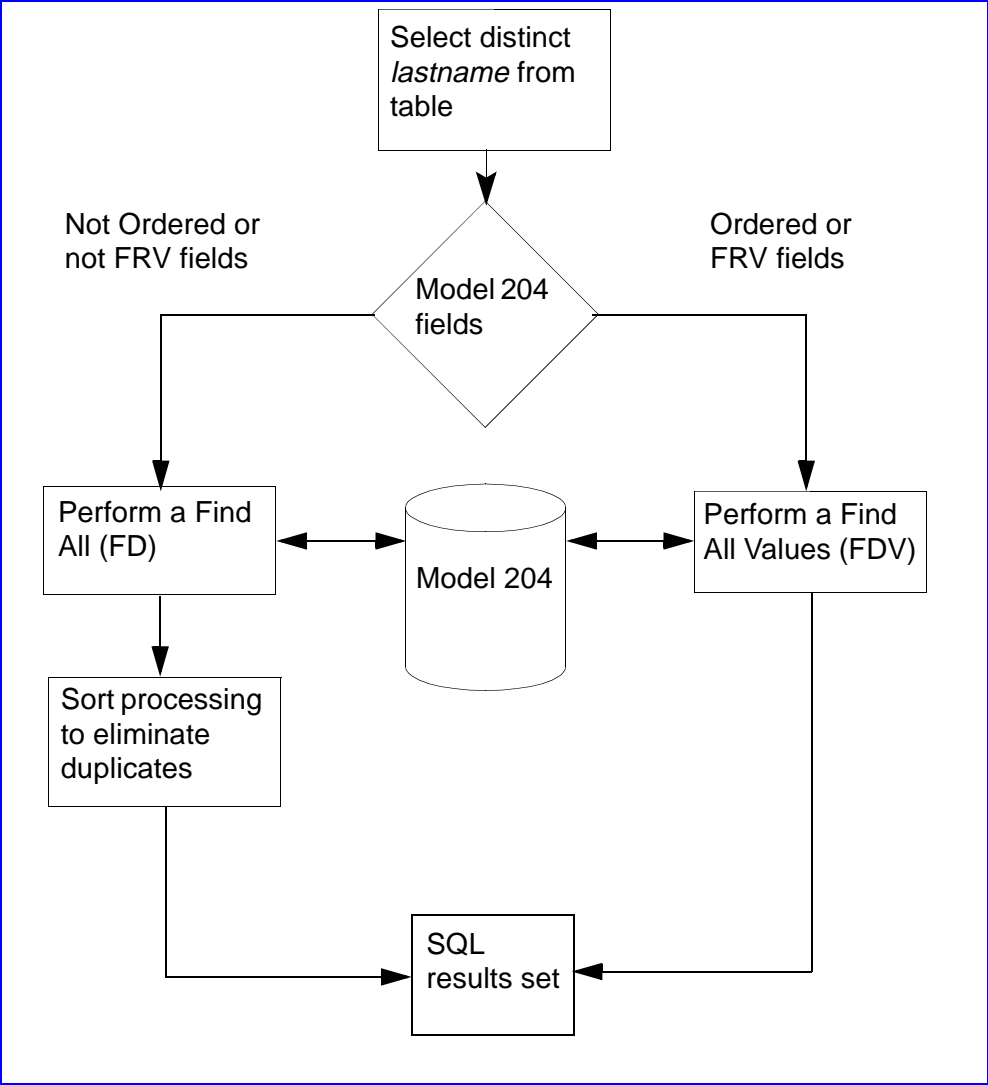


Figure 1-2. SQL SELECT DISTINCT processing paths

Handling SQL statements greater than 32K bytes

When handling an SQL statement that is greater than 32K, you must consider the following parameters:

- The SQLBUFSZ parameter is the buffer to collect the entire SQL statement. It must be slightly greater than the maximum size of the largest SQL statement, which is the value of SQLI in the audit trail. For example, set SQLBUFSZ=60000 to run a DDL statement that is 52K.
- The SQLIQBSZ parameter is the buffer for the interface with IFAM. For example, set SQLIQBSZ=32000 to run an IFGET result that requires a buffer greater than 31K.
- Other pertinent parameters to review are VTBL, LHEAP, and SERVSZ. If their values are not large enough, the error message invoked includes the name of the relevant parameter.

To allow LHEAP and VTBL to be as large as possible, conserve address space by setting the buffer size parameters as low as possible.

Handling foreign keys

If a foreign key is defined twice, first with a REFERENCES clause and then with a FOREIGN KEY clause, the statement is accepted only if the two clauses are identical and reference the same column.

SQL error message -4703 is generated if a FOREIGN key is defined twice for a nested table and the two keys do not reference the same column. For example:

Acceptable:

```
CREATE TABLE NESTEDINVENTOR2 NESTED USING PART_NO
(PART_NO
  DECIMAL(8) NOT NULL
  REFERENCES INVENTORY,
ON_HAND
  SYSNAME 'ON HAND'
  FLOAT(4) NOT NULL,
LOCATI ON
  CHAR(255) NOT NULL,
FOREI GN KEY (PART_NO) REFERENCES INVENTORY)
```

Returns SQL error -4703:

```
CREATE TABLE NESTEDINVENTOR2 NESTED USING PART_NO
(PART_NO
  DECIMAL(8) NOT NULL
  REFERENCES INVENTORY,
ON_HAND
  SYSNAME 'ON HAND'
  FLOAT(4) NOT NULL,
LOCATI ON
  CHAR(255) NOT NULL,
FOREI GN KEY (ON_HAND) REFERENCES INVENTORY)
```

Handling primary keys

SQL error message -6315 is now displayed if a column in a table is defined with PRIMARY KEY SYSTEM and the SYSNAME clause. The following example returns SAL error message -6315:

```
CREATE TABLE XYZ
(COL1 INT,
COL2 CHAR (25),
F_KEY INT
PRIMARY KEY SYSTEM
SYSNAME 'INDEX')
```

SQL statement enhancement

In this release of Model 204, the RSQL engine was significantly modified with the following features:

- You can use correlation names or aliases for column names and aggregate functions.
- There are new keywords for current date, time, timestamp, user, and so on.
- New syntax was added for inner join processing.
- Right and left outer joins are supported.

Options in the **SELECT LIST** statement

Correlation name feature

The name in the SELECT LIST statement can be assigned to a new name, which functions as an alias, by either:

- The AS keyword
- An equal sign (=) sign
- A white space.

If the name represents a column, the newly assigned name may be referred to in other clauses. The following examples are supported:

```
SELECT COUNT(*) AS COUNT FROM TABLEX

SELECT T1.TITLE TITLE, CHAPTER = T2.TITLE
      FROM BOOK T1, CHAPTERS T2
      WHERE TITLE = 'WWII' AND CHAPTER = 'US'
```

The following example is not supported:

```
SELECT DOUBLE_ADVANCE = (ADVANCE * 2)
      FROM PUBLISHERS
      WHERE DOUBLE_ADVANCE > 1000
```

Note: Correlation names may not carry over to a query at a different level of an SQL statement.

Wildcard asterisk (*) for an individual table feature

```
SELECT T.TITLE, T.*, P.*
      FROM TITLES T, PUBLISHERS P
      WHERE T.ID = P.ID
```

CURRENT_TIME keyword

The CURRENT_TIME keyword is assigned as CHAR(11) in the form of *hh:mm:ss.tt*, representing the local time at the time of the query execution.

```
SELECT CURRENT_TIME, COUNT(*) FROM TABLEX
```

```
10: 05: 24. 38
```

Note: In this example query and those to follow, the COUNT(*) is added merely for the purpose of generating a single row of output.

CURRENT_DATE keyword

The CURRENT_DATE keyword is assigned as CHAR(10) in the form of *yyyy-mm-dd* to represent the date at the time of the query execution.

```
SELECT CURRENT_DATE, COUNT(*) FROM TABLEX
```

```
2001- 03- 08
```

CURRENT_TIMESTAMP keyword

The CURRENT_TIMESTAMP keyword is assigned as CHAR(22) in the form of *yyyy-mm-dd hh:mm:ss.tt*, the time and the date at the time of the query execution.

```
SELECT CURRENT_TIMESTAMP, COUNT(*) FROM TABLEX
```

```
2001- 03- 08  10: 05: 24. 38
```

USER keyword

The USER keyword is assigned as CHAR(10). It represents the name of the user who logged on to the thread.

```
SELECT USER, COUNT(*) FROM TABLEX
```

```
USER101
```

SQLVERSION keyword

The SQLVERSION keyword is assigned as CHAR(16). It represents the version code of the RSQL module and the released date in the form: *n.n.na mm/dd/yy*

```
SELECT SQLVERSION, COUNT(*) FROM TABLEX
```

```
5. 1. 0C  05/30/01
```

This keyword is intended for diagnostic purposes.

SQLERROR keyword

The SQLERROR keyword is assigned as CHAR(5). It represents the last error code detected and recorded in the RSQL module. Six locations are reserved to store the last six error codes. For example:

```
SELECT SQLERROR, SQLERROR COUNT(*) FROM TABLEX
```

```
+0000, - 0206
```

The SQLERROR keyword is intended for diagnostic purposes.

See "2500" "SQL ERROR error-code error-message" on page 5-21.

SQLSTATE keyword

The SQLSTATE keyword is assigned as CHAR(32). It represents the significant part of the last error message or the object name which caused the error, as recorded in the RSQL module. For example, SQLSTATE will contain 'UNITCOST', if the previous error was:

```
SQL Error - 408. A value is not compatible with its  
object column UNITCOST, for EXECIMM, in EXECUTE  
completion routine
```

The SQLSTATE keyword is intended for diagnostic purposes.

Additional SQL INNER JOIN features

CROSS JOIN feature

The CROSS JOIN feature produces a cross-product of a table join in which no join condition is specified. For example:

```
SELECT * FROM TITLES CROSS JOIN PUBLISHERS
```

The previous query is equivalent to the following example:

```
SELECT * FROM TITLES, PUBLISHERS
```

NATURAL JOIN feature

In Model 204 Version 5.1, INNER is the only join type that is supported for NATURAL JOIN.

The NATURAL JOIN feature combines two tables on all columns that share the same names. For example, if PUB_ID and AUTHOR are the only two columns which are common to both tables, then the following query with the NATURAL JOIN clause produces the same results as the query with the WHERE clause:

```
SELECT * FROM TITLES NATURAL JOIN PUBLISHERS
```

```
SELECT * FROM TITLES, PUBLISHERS
```

```
WHERE TITLES.PUB_ID = PUBLISHERS.PUB_ID
AND TITLES.AUTHOR = PUBLISHERS.AUTHOR
```

ON clause feature

The conditional join uses an ON clause to impose the join condition. It is interchangeable with the WHERE clause. For example, the following query uses an ON clause:

```
SELECT * FROM TITLES JOIN PUBLISHERS
ON TITLES.PUB_ID = PUBLISHERS.PUB_ID
```

The previous query is equivalent to the following query which employs a WHERE clause:

```
SELECT * FROM TITLES, PUBLISHERS
WHERE TITLES.PUB_ID = PUBLISHERS.PUB_ID
```

USING clause feature

The NATURAL JOIN feature matches all column names in one table with the same column names in a second table. If only some column names in one table match those column names in a second table, you can explicitly specify those columns with the USING clause, as follows:

```
SELECT * FROM TITLES JOIN PUBLISHERS
USING (PUB_ID, AUTHOR)
```

The previous query is equivalent to a query in which the USING keyword is replaced by the ON clause, such as

```
SELECT * FROM TITLES, PUBLISHERS
WHERE TITLES.PUB_ID = PUBLISHERS.PUB_ID
AND TITLES.AUTHOR = PUBLISHERS.AUTHOR
```

Note: The ON clause is *not* allowed to contain a subquery.

SQL OUTER JOIN features

Unlike the INNER JOIN feature, the OUTER JOIN feature preserves the unmatched rows from one of the two tables, depending on the keywords LEFT and RIGHT.

The following two equivalent queries are examples of a LEFT OUTER JOIN clause.

```
SELECT * FROM TITLES LEFT OUTER JOIN PUBLISHERS
USING (PUB_ID, AUTHOR)
```

```
SELECT * FROM TITLES LEFT OUTER JOIN PUBLISHERS
ON (TITLES.PUB_ID = PUBLISHERS.PUB_ID
AND TITLES.AUTHOR = PUBLISHERS.AUTHOR)
```

Note: The keyword OUTER in these queries is optional.

In these examples, each row in the first table, TITLES, in a LEFT OUTER JOIN clause must be included in the result. If no matching value is found in the second table, PUBLISHERS, the corresponding columns of PUBLISHERS are filled with NULLs.

The RIGHT OUTER JOIN clause operates similarly to a LEFT OUTER JOIN clause, except that the RIGHT or the second table is the parent table and its rows are preserved. For example, the previous two LEFT OUTER JOIN queries can be expressed as RIGHT OUTER JOIN queries.

```
SELECT * FROM PUBLISHERS RIGHT OUTER JOIN TITLES
      USING (PUB_ID, AUTHOR)
```

```
SELECT * FROM PUBLISHERS RIGHT OUTER JOIN TITLES
      ON (TITLES.PUB_ID = PUBLISHERS.PUB_ID
          AND TITLES.AUTHOR = PUBLISHERS.AUTHOR)
```

Note: The ON clause is *not* allowed to contain a subquery.

Restrictions using the OUTER JOIN feature

You can retrieve all rows from one of the joined tables using the OUTER JOIN feature and one of the following keywords: LEFT or RIGHT.

Currently, there are two restrictions on the implementation of the OUTER JOIN feature:

1. A single child table in an outer join must have one and only one parent table. For example:

```
SELECT T.PUB_ID TPID, TA.TITLE_ID, P.PUB_ID PPIID,
       T.TITLE
FROM TITLES1 T RIGHT OUTER JOIN PUBLISHERS AS P
      ON T.PUB_ID = P.PUB_ID
      RIGHT OUTER JOIN TITLEAUTHORS TA
      ON T.TITLE_ID = TA.TITLE_ID
ORDER BY T.PUB_ID, TA.TITLE_ID
```

In the first outer join of this query, the PUBLISHERS table is the parent table of the TITLES1 table. In the second outer join, the TITLEAUTHORS table also becomes the parent table of TITLES1. So TITLES1 has two parent tables.

Another example:

```
SELECT T.TITLE_ID, A.AU_ID, R.ROYALTY
FROM TITLEAUTHORS A LEFT JOIN TITLES1 T
      ON T.TITLE_ID = A.TITLE_ID
      RIGHT JOIN ROYALTY R
      ON T.TITLE_ID = R.TITLE_ID
```

The first join is a LEFT join where the TITLEAUTHORS table is the parent table to the child table, TITLES1. The second join is a RIGHT join of the

ROYAL14 table to the TITLES1 table. In this case also, the TITLES1 table is the child table to two parent tables, TITLEAUTHORS and ROYAL14.

The error message returned for these queries is:

SQL Error - 2088. Multiple parent tables in join qualifier(s).

2. Outer joins which reference nested tables are not supported. For example:

```
SELECT * FROM SALES LEFT OUTER JOIN ORDERS  
USING (PONUM)
```

ORDERS in this query is a nested table, so the query will not be executed but will result in the following error message:

SQL Error - 2089. A nested table ORDERS in an outer join is not supported.

New SQL reserved words

The following words are now reserved words in SQL:

CROSS

CURRENT_DATE

CURRENT_TIME

CURRENT_TIMESTAMP

FULL

INNER

JOIN

LEFT

NATURAL

OUTER

RIGHT

SQLSTATE

SQLVERSION

Horizon for TCP/IP and Horizon/LE for TCP/IP

Horizon for TCP/IP and Horizon/LE for TCP/IP both handle process-to-process communications.

Required IBM APARs

You must apply the following APARs from IBM to eliminate IBM TCP/IP problems:

- OW43994
- PQ39019
- PQ49319: takes care of byte being duplicated with fast socket support in memory.
- PQ46754: takes care of entire message thrown away because TCP message sequence number wrapped around zero.

Expanded syntax for the DEFINE PROCESSGROUP command

To use Horizon for TCP/IP to support TCP client applications, you must now specify the port number of the remote client process, as shown in the following syntax. If you are using Horizon/LE for TCP/IP to support Connect ★ threads, then you do not need to make this change.

Syntax

```
DEFI NE PROCESSGROUP name [ LI KE previousness ]  
  
WI TH SCOPE=SYSTEM  
  
LI NK=link-name REMOTEI D=address I NLI MI T={ nnnn | 0 }  
  
[ PORT=nnnnn ]  
  
[ GUESTUSER={ ACCEPT | REJECT } ]  
  
[ LOGI N=TRUST | NOTRUST ]  
  
[ MASK={ dot t ed-deci mal | 255. 255. 255. 255 ]
```

Where

PORT specifies the port number of the remote TCP/IP server process. The value of *PORT* is a number between 1024 and 65535, inclusive, and must be unique within the host machine.

The *PORT* parameter is required only for a client connection to a remote server. Therefore, in defining a client as shown in “For the client” on page 1-73, you must define the *PORT* parameter in the DEFINE PROCESSGROUP command, but in defining the remote server as shown in “For the server” on page 1-73, you do not define the *PORT* parameter in that DEFINE PROCESSGROUP command.

Example

To set up a Horizon for TCP/IP connection, define a link, process group, a process for a client, and open the link. Then, define the same items on the server side and open the link, as shown in the following steps.

For the client

1. Define a link for a client named TCPC.

```
DEFINE LINK TCPC WITH SCOPE=SYSTEM PROTOCOL=IP -  
LOCALID=192.168.0.2 INBUFSIZE=4096 -  
TRANSPORT=TCPSE -  
CONNECTIONS=60 SERVPOR=65535
```

2. Define a process group named TCPG.

```
DEFINE PROCESSGROUP TCPG WITH SCOPE=SYSTEM -  
LINK=TCPC -  
INLIMIT=30 OUTLIMIT=30 REMOTEID=192.168.0.2 -  
MASK=255.255.255.255 PORT=65534
```

The value of PORT, 65534, is used as the SERVPOR value in the DEFINE LINK command for the server.

3. Define a process named CPR.

```
DEFINE PROCESS CPR WITH SCOPE=SYSTEM -  
DATALEN=9999 PARTNER=SPR -  
DESTINATION=TCPG UI DSOURCE=OPEN
```

The value of PARTNER, SPR, is the process name in the DEFINE PROCESS command for the server.

4. Open the client link named TCPC.

```
OPEN LINK TCPC
```

For the server

1. Define a link for the server named TCPS that corresponds to the link for the client. In this example, the SERVPOR value, 65534, matches the PORT value defined in the client DEFINE PROCESSGROUP command.

```
DEFINE LINK TCPS WITH SCOPE=SYSTEM PROTOCOL=IP -  
LOCALID=192.168.0.2 INBUFSIZE=4096 -  
TRANSPORT=TCPSE -  
CONNECTIONS=60 SERVPOR=65534
```

2. Define a process group TSPG for the server.

```
DEFINE PROCESSGROUP TSPG WITH SCOPE=SYSTEM -  
LINK=TCPS -  
INLIMIT=30 OUTLIMIT=30 REMOTEID=192.168.0.2 -  
MASK=255.255.255.255
```

3. Define a process named SPR for the server.

```
DEFINE PROCESS SPR WITH SCOPE=SYSTEM FROM TSPG -  
DATALEN=9999 TIMEOUT=0 -  
SUBSYSTEM=CONS SUBSYSPARM=' CONSPROC RE-TPROC -  
NOPEX'
```

The process name defined for the server must match the value of the PARTNER parameter in the client DEFINE PROCESS command.

4. Open the server link named TCPS.

```
OPEN LINK TCPS
```

Usage

CNOS limitations

Horizon for TCP/IP and Horizon/LE for TCP/IP do not support CNOS. CNOS is exclusively an SNA LU62 protocol as is documented in the *Horizon Intersystem Processing Guide*.

VSE operating system limitations

The VSE operating system does not support Horizon for TCP/IP. The VSE operating system continues to support the Horizon functionality of earlier releases.

Conforming to IBM SNA LU62 Peer Protocol Reference

To conform with the specification set down in the IBM SNA LU62 Peer Protocol Reference, the DEFINE PROCESSGROUP command LOGIN parameter must be consistent for all sessions between the same Logical Unit pairs. This means that all PROCESSGROUPs sharing the same LINK and REMOTEID must also share the same LOGIN characteristic. See *SNA LU62 Peer Protocol Reference*, Chapter 5 "Session Manager, Session Control RUs."

If you have not met with this requirement, you must modify the LOGIN parameter on your DEFINE PROCESSGROUP commands.

To implement Horizon for TCP/IP, refer to the *Model 204 Command Reference Manual* for a complete description of the appropriate DEFINE commands.

Converting from Horizon for VTAM to Horizon for TCP/IP

If you use Horizon for VTAM and want to convert to Horizon for TCP/IP, you may encounter the error: BIND ERRNO 48. This can occur when IBM's TCP—either as a socket or port number—is in a TIME-WAIT state and is not immediately available after a TCP connection is closed.

IBM implemented this TIME-WAIT state to counter the following scenario. If a TCP connection is closed and opened immediately, it is possible that a packet from the closed connection will reach the peer TCP. If the sequence numbers are in the correct range, the peer TCP might assume these packets are from the live connection. To avoid such a confusion, the system waits for twice the

Maximum Segment Life (MSL) before reusing the port, which gives enough time to flush old connection packets from the network.

You may perceive this TIME-WAIT as a delay in the OPEN LINK processing; in fact, it functions as IBM designed it to avoid the BIND ERRNO 48. If you encounter this error, it may be alleviated or eliminated by keeping the link open. Then, open and close processes as needed, rather than opening and closing the link.

The following table suggests a possible workflow. In this workflow, the socket is opened and closed as each process is opened and closed, but the bind for the link remains open until you have completed a work session.

Model 204 commands	What is TCP/IP Doing?
OPEN LINK xyz	socket(), bind(), listen(), accept() - Get socket for the server and listen for a client
OPEN PROCESS	connect() - Open a client and connect to the server
...do work	read(), write() - Read and Write to and from the server read(), write() - Read and Write to and from the server
...more work	read(), write() - Read and Write to and from the server
CLOSE PROCESS	closesocket() - Close the client connection to the server
OPEN PROCESS	connect() - Open a client and connect to the server
...do work	read(), write() - Read and Write to and from the server
...more work	read(), write() - Read and Write to and from the server
CLOSE PROCESS	closesocket() - Close the client connection to the server
....	
CLOSE LINK xyz	closesocket(), shutdown() - Close the server and shutdown the TCP/IP socket

Parameters introduced or changed in Version 5.1

APSYPAGE: Number of virtual pages for precompiled procedures

Class	SYSTEM
Default	0
Setting	On User 0's parameter line
Meaning	Activates the APSY Precompiled Procedures In Storage feature by specifying a number of 4K operating system pages. For z/OS and OS/390 only.

AUDITOPT: Audits MQ/204 messages

Class	SYSTEM
Default	X'00000000'
Setting	On User 0's parameter line or reset by system manager
Meaning	The AUDITOPT parameter indicates special handling for the auditing of various messages. The parameter is bit settings with the following possible values:

Setting	Meaning
X'00000000'	Default value
X'40000000'	Indicates that MQ/204 messages M204.2504, 2505, and 2506 are not to be written to the audit trail if an MQ/204 operation (other than MQGET, MQPUT, or MQPUT1 statement) is successful.
X'80000000'	Indicates that MQ/204 messages M204.2504, 2505, and 2506 are not to be written to the audit trail if an MQGET, MQPUT or MQPUT1 statement is successful.

CPQZACTN: Action to take when the CPQZSECS time limit expires

Class	SYSTEM
Default	X'80'
Setting	On User 0's parameter line or reset by system manager

Meaning

If an extended quiesce is in progress, one of the following actions will take place when the CPQZSECS time limit expires. The action depends on the setting of CPQZACTN. The settings are mutually exclusive.

Setting	Meaning
X'10'	EOJ - terminate the run
X'20'	OPERATOR - prompts the Operator to provide what action to take: EOJ, ENDEQ, or CONTINUE
X'40'	ENDEQ - end extended quiesce state, release all updating users
X'80'	CONTINUE - the extended quiesce should continue for the length of CPQZSECS seconds

If the extended quiesce is ended before the CPQZSECS time limit expires, the CPQZACTN setting is ignored.

Usage

Unlike most other parameters, the value of CPQZACTN is partially validated at Online initialization and partially validated when a CHKPOINT dataset is first opened.

- If NUSERS is one, or RCVOPT bit X'01' is off, which means checkpointing is not active, the extended quiesce CCAIN parameters are not validated. They cannot be reset and they are viewed as zero.
- If neither extended quiesce parameter, CPQZACTN or CPQZSECS, is specified on CCAIN, and NUSERS is greater than one, and checkpointing is active, the CPQZACTN is set to X'80' at the end of system initialization.
- If NUSERS is greater than one and checkpointing is active, CPQZACTN can be reset at any time during the run.
 - When the system is not in extended quiesce, the reset takes effect immediately.
 - During extended quiesce, a new value of CPQZACTN takes effect only *after* a M204.2610 message is issued. The M204.2610 message is issued every minute.
 - If the extended quiesce ends before the next M204.2610 message is issued, the new value of CPQZACTN is in effect at the next checkpoint.

CPQZSECS: Maximum duration, in seconds, of an extended quiesce

Class	SYSTEM
Default	99999
Setting	On User 0's parameter line or reset by system manager

Meaning

The CPQZSECS parameter establishes an extended quiesce timer. An extended quiesce begins after a successful checkpoint taken by a system following a CHECKPOINT SET EXTENDED QUIESCE command; it typically ends with a CHECKPOINT END EXTENDED QUIESCE command.

The CPQZSECS timer starts at the beginning of the extended quiesce. When the timer expires, the extended quiesce system takes the action specified in CPQZACTN.

The CPQZSECS parameter can be set to values 5 to 99999.

Usage

Unlike most other parameters, the value of CPQZSECS is partially validated at Online initialization and partially validated when a CHKPOINT dataset is first opened.

- If NUSERS is set to one, or RCVOPT bit X'01' is off, which means checkpointing is not active, the extended quiesce CCAIN parameters are not validated. They cannot be reset and they are viewed as zero.
- If neither extended quiesce parameter, CPQZSECS or CPQZACTN, is specified on CCAIN, and NUSERS is greater than one, and checkpointing is active, the CPQZSECS parameter is set to 99999 at the end of system initialization.
- If NUSERS is greater than one and checkpointing is active, the CPQZSECS parameter can be reset at any time during the run.
 - When the system is not in extended quiesce, the reset takes effect immediately.
 - During extended quiesce, the new value of CPQZSECS takes effect only after the M204.2610 message is issued. (The M204.2610 message is issued every minute.)
 - If the extended quiesce ends before the next M204.2610 message is issued, the new value of CPQZSECS is in effect at the next checkpoint.

CUSTOM: Using customized parameters

Starting in Version 5.1, CUSTOM =6, which changes the LOGOUT command message, and CUSTOM=7, which allows you to retain the pre-Version 5.1 MSGCTL command processing, are added to the options of the CUSTOM parameter.

CUSTOM=6

The LOGOUT command message number is changed by setting the CUSTOM parameter to 6. If CUSTOM is:

- Set to 6, the LOGOUT message number is:

```
M204. 2628: uuuuuuuuuu aaaaaaaaaa LOGOUT yy mmm  
dd hh. mm
```


- Not set to 6, the LOGOUT message number is as in releases prior to Version 5.1:

```
M204. 0353: uuuuuuuuuu aaaaaaaaaa llllll yy mmm
dd hh. mm
```

Using this option you can control the message numbers of the LOGIN and LOGOUT commands independently.

Previously, BATCH2 and IFAM2 applications would not receive the return code from Model 204, if the command MSGCTL M204.0353 NOTERM was issued in the Online. Model 204 sends the return code to BATCH2 and IFAM2 User Language applications as a hidden field in the LOGOUT message, and the application must receive the LOGOUT message to obtain the return code. You may now suppress M204.0353 and leave M204.2628 active and available for a BATCH2 or IFAM2 User Language application to receive the return code from the Online.

CUSTOM=7

See “MSGCTL command behavior change” on page 2-7 for a complete description of the changes to the MSGCTL command and how the message options are handled.

DSPOPT: Data- and hiperspace options

Class	SYSTEM
Default	X'00'
Setting	On User 0's parameter line and can be reset.
Meaning	The bits have the following meanings:

Settings	Meaning
X'01'	Use page-oriented movement for servers, which means use z/OS 4K pages instead of byte-oriented movement. Page-align servers and use the MVPG hardware instruction to do CCASERVR I/O. Using page-aligned CCASERVRs lets Model 204 use a page-oriented storage allocation algorithm that probably requires a significantly smaller working set size than using non-page-aligned CCASERVRs.
X'02'	Put servers in hiperspace, which means to put the servers in a hiperspace instead of a dataspace.

Settings	Meaning
X'20'	<p>Put APSY precompiled procedures in hiperspace instead of a dataspace. Hiperspace pages can be in either real or expanded storage.</p> <p>This option is only valid on OS/390 and requires that Model 204 be APF-authorized.</p>
X'40'	<p>Put APSY precompiled procedures in cache hiperspace instead of a scrolling hiperspace.</p> <p>Cache hiperspace pages can reside in only expanded storage, so they are sometimes called ESO (for Expanded Storage Only) hiperspaces.</p> <p>Also, cache hiperspace pages never result in operating system paging. Pages stolen by the operating system are simply discarded and Model 204 is responsible for replacing the pages from the copy of the precompiled procedures in CCATEMP.</p> <p>This option is only valid on OS/390 and requires that Model 204 be APF authorized.</p>

ECISUBS: Subtasks initially attached for running external modules

Class	SYSTEM
Default	0
Setting	On User 0's parameter line
Meaning	Number of subtasks initially attached for running external modules. The maximum number of subtasks permitted is 500; the minimum is 0.

ECMODS: External modules to load

Class	SYSTEM
Default	50
Setting	On User 0's parameter line
Meaning	Maximum number of external modules to load in this run. The maximum number of modules allowed is 32767; the minimum is 0.

ECMSUBS: Number of ECF subtasks for running modules

Class	SYSTEM
Default	0

Setting On User 0's parameter line

Meaning Specifies the maximum number of ECF subtasks for running external modules. The maximum number of ECF subtasks allowed is 500; the minimum is 0.

ECNAMES: Number of external call names

Class SYSTEM

Default 80

Setting On User 0's parameter line

Meaning Specifies the maximum number of ECF call names to be defined in this run. The maximum number of ECF call names possible is 32767; the minimum is 0.

ECPRIV: ECF privileges

Class SYSTEM

Default 0

Setting On User 0's parameter line

Meaning Specifies who can issue privileged ECF statements. All EXTERNAL statements, except EXTERNAL CALL, are privileged.

Value	Means
0	End-users can issue only EXTERNAL CALL statements; no other ECF statements are allowed.
1	Any APSY user can issue privileged ECF statements.
2	Any non-APSY user can issue privileged ECF statements.

You can select multiple options by summing the values pertaining to each desired option.

ECPSIZE: Size of initial parameter area allocated by ECF

Class SYSTEM

Default 1024

Setting On User 0's parameter line

Meaning Specifies the initial size of the buffer used to hold the copy of the parameter area passed to an external module. Buffers are dynamically expanded when required; therefore, ECPSIZE indicates the initial and minimum size of the buffer. The maximum initial size is 229,369 bytes; the minimum is 0.

There is no limit on the size of any image or images that are passed to any external module, except for modules with the PARMSIZE option.

ECWAIT: ECF wait time

Class SYSTEM

Default 0

Setting On User 0's parameter line or reset by system manager

Meaning Specifies the maximum number of milliseconds a user waits for an ECF module or an ECF subtask to become free. The maximum possible number of milliseconds to wait is 2,147,483,647; the minimum is 0.

GTBLHASH: Number of buckets for global string variables

Class USER

Default 0

Setting By any user

Meaning The number of hash buckets allocated in GTBL to hold global string variable names. The benefit of using GTBLHASH is that the scan time and CPU time to find and update global string variables is reduced, typically, by a factor of GTBLHASH.

When a global string variable is set or retrieved, the name of the global is hashed into one of the GTBLHASH buckets. Once the bucket is determined, the scanning and updating works as it did prior to Version 5.1, except there is less data to move when a value is deleted or its size changes.

By default, when you place a global string variable in GTBL, it is stored in order-of-entry. If you set GTBLHASH to a nonzero value the data is placed in GTBL in the order determined by the hash value of the global string variable name.

CCA recommends that you set the number of buckets to a prime number, which is not greater than the expected maximum number of global variables divided by four. You can find your optimal value through experimentation; CCA recommends that you start with a relatively small number such as five or seven.

Resetting the GTBLHASH parameter clears the global table.

Calculating GTBL space requirements

When hashed GTBL is being used, GTBL requires an extra 8 bytes for every hash bucket, which means that you should probably increase LGTBL by at least eight times the value of GTBLHASH. Since hashing does not distribute hash values exactly evenly across buckets, you should probably increase the size of GTBL by somewhat more than this for a little extra space in each bucket.

CCA recommends that you increase LGTBL by $GTBLHASH * (8 + av_len * 3)$ where *av_len* is the average expected length of the global string variable names, plus their values. So, if GTBLHASH is being set to seven and the average expected length of globals and their values is 20, LGTBL would be increased as follows:

$$7 * (8 + 20 * 3) = 476.$$

GTBLPCT: Initial percentage of GTBL to allocate for global string variables

Class	USER
Default	50
Setting	By any user
Meaning	<p>The GTBLPCT parameter indicates the initial percentage of GTBL to allocate for storage of global string variables; the remaining storage area is for global objects such as global screens, images, lists, foundsets, and positions.</p> <p>The GTBLPCT parameter has no effect if GTBLHASH=0.</p> <p>Resetting the GTBLPCT parameter clears the global table.</p> <p>CCA recommends that you set GTBLPCT high, if you have mostly global strings and a few global objects; low, if the reverse applies.</p>

LENQTBL: Resource enqueueing table entries

Class	SYSTEM
Default	6
Setting	On User 0's parameter line
Meaning	<p>The number of entries in the resource enqueueing table to be allocated for each user</p> <p>The resource enqueueing table is used for many purposes. Opening many files concurrently and using nested INCLUDEs contribute to filling up this table.</p>

The value of LENQTBL defaults to 6, although the table is always allocated with a minimum of 12 entries.

The resource enqueueing table is a volatile structure: its entries can be allocated and deallocated very quickly. As a result, any prediction of its optimum size can only be roughly approximate. Typically, the following formula for sizing LENQTBL suffices, although there might be cases where you need a larger or a smaller LENQTBL value.

Set LENQTBL to the sum of the following items divided by the number of users (NUSERS)

Algorithm for determining the value of LENQTBL

$$\text{LENQTBL} = ((4 * \text{number-of-files-per-request}) + 7 + \text{number-of-APSY-subsystems}) / 100$$

where:

- *number-of-files-per-request* are job determined
- *number-of-APSY-subsystems* are job determined

For example when:

- *number-of-files-per-request* = 3
- *number-of-APSY-subsystems* = 20

then the calculation is as follows:

$$((4 * 3) + 7 + 20) / 100 = .39, \text{ which rounds up to } 1.$$

Algorithm for corresponding storage requirements:

The total amount of storage required for the resource enqueueing table is:

$$(\text{NUSERS} * \text{LENQTBL} * 28) + 16 + (\text{NFILES} * 32)$$

For example, when:

$$\text{NUSERS} = 100$$

$$\text{LENQTBL} = 12, \text{ which is always minimum allocation}$$

$$\text{NFILES} = 200$$

then the calculation is as follows:

$$(100 * 12 * 28) + 16 + (200 * 32) = 40,016 \text{ bytes}$$

MAXBUF: File page buffer maximum

Class SYSTEM

Default	256
Setting	On User 0's parameter line
Meaning	<p>This is the maximum number of in-memory file page buffers that can be allocated during Model 204 initialization. The actual number of pages allocated to the disk buffer pool can be identified by viewing the parameter NUMBUF.</p> <p>Starting in Version 5.1, if MAXBUF is set to a value less than MINBUF, MAXBUF is automatically reset to the value of MINBUF and the following message is issued:</p> <p>M204. 1190: MAXBUF SET TO SAME VALUE AS MINBUF - %C</p> <p>Prior to Version 5.1, an error message prevented the Online from coming up.</p>

MAXINCL: Number of procedure INCLUDE nesting levels

Class	SYSTEM
Default	5
Setting	On User 0's parameter line
Meaning	The MAXINCL parameter indicates the maximum procedure INCLUDE nesting level. The default is 5, which is the setting that was used in Model 204 prior to Version 5.1. You can set MAXINCL to a value between 5 and 40, inclusive.
Usage	<p>Each extra INCLUDE level requires (65 + LAUDPROC) bytes of fixed server area, where LAUDPROC is the User 0 parameter value. If LAUDPROC is set to 55, each extra INCLUDE level would require 120 extra bytes of fixed server area.</p> <p>You can issue a VIEW SYSTEM command to display the MAXINCL readings.</p> <p>If you set the MAXINCL parameter to a value greater than five, and you run a request with \$RDPROC calls, the request will use more STBL space.</p>

MAXOBUF: Maximum number of open disk buffers per server

Class	SYSTEM
Default	4
Setting	On User 0's parameter line
Meaning	User's can have open buffers only while in a server. The maximum value of MAXOBUF is 255. Model 204 uses the extra open buffer limit above four to try

to do fast page reads on frequently accessed pages. CCA recommends that you adjust the MAXOBUF parameter setting only if you work in an MP/204 environment. Adjusting it does not have significant impact in other environments.

You can regard the MAXOBUF parameter as a user-level working set size for disk buffer pool pages. However, consider the following disadvantages of setting the value of the MAXOBUF parameter too high:

- First, a too-high value can reduce the accuracy of the Model 204 LRU algorithm for disk buffer pool pages.
- Second, and more importantly, a too-high value can actually increase CPU overhead, because of relatively long chains of open pages that must be scanned for a fast read.

CCA recommends that you test the values of MAXOBUF, gradually increasing it until the number of DKPRs compared to work accomplished stops decreasing significantly.

MINBUF: File page buffer minimum

Class	SYSTEM
Default	18
Setting	On User 0's parameter line
Meaning	The minimum number of file page buffers that must be allocated for initialization to succeed. If this number cannot be allocated due to insufficient virtual storage, initialization is terminated.

Starting in Version 5.1, the MINBUF parameter is automatically set to its smallest allowable value. The minimum valid value for MINBUF is the number of servers (the value of the NSERVS parameter) plus the maximum number of pseudo-subtasks (the value of NSUBTKS parameter), times the maximum number of open disk buffers per server (the value of the MAXOBUF parameter), plus fifteen, represented as follows:

$$(\text{NSERVS} + \text{NSUBTKS}) * \text{MAXOBUF} + 15$$

This is a change from previous releases where the minimum allowable value of MINBUF was (NSERVS*4)+14 for MP/204 and 18 otherwise.

Also starting in Version 5.1, if MINBUF is set to a value greater than MAXOBUF, MAXOBUF is automatically reset to the value of MINBUF. Prior to Version 5.1, an error message prevented the Online from coming up.

Refer to the *Model 204 System Manager's Guide* for more information on page buffers.

MPSYS: MP/204 processing options

Class	SYSTEM
Default	0
Setting	On User 0's parameter line or reset by system manager
Meaning	This parameter indicates a system-wide MP/204 processing option and applies only to OS/390 and z/OS.

Setting	Meaning	Purpose
X'01'	A thread should continue spinning on an MP lock even if it appears that the task holding the lock cannot be dispatched.	In many situations, continuing to spin provides better performance than doing an operating system wait for the lock to free up.
X'02'	A thread should stay on the maintask after getting a critical file resource.	To improve performance at an MP/204 site that is encountering a lot of task switching (a high STDEQ system statistic) and a lot of critical file resource conflicts.

Usage In Version 5.1, the default value is MPSYS=0 to preserve the previous behavior. However, setting MPSYS=X'01' improves performance in almost all cases.

This parameter is ignored when Model 204 is not running with MP/204 enabled.

MSRES: Maximum number of Single Record Entries

Class	SYSTEM
Default	0
Setting	On User 0's parameter line or reset by system manager
Meaning	<p>The MSRES parameter controls how entries are stored in the Record Enqueuing Table for Transaction Back Out (TBO) pending updates. When MSRES is nonzero, additional space may be required to store these entries, but the CPU cost of processing them may be reduced. (The amount of additional space and level of performance improvement are application dependent.)</p> <p>In Version 5.1, the default value of MSRES is 0 to preserve the previous behavior. However, setting MSRES to a small, nonzero value can often improve performance.</p>

Technical note: The MSRES parameter specifies the maximum number of Single Record Entries (SREs) per file to store in the Record Enqueuing Table for Transaction Back Out (TBO) pending updates. When a file requires more than MSRES SREs, all SREs are converted to record set entries.

Prior to Version 5.1, pending update entries in the RETBL were managed as though MSRES was set to 0, which means record set entries were always used.

Usage In a small number of cases, setting the MSRES parameter may require a larger record enqueuing table than with the default MSRES=0. If you get message

M204. 1219 RECORD LOCKING TABLE FULL

then either increase the size of the record enqueuing table by increasing LRETBL, or specify MSRES=0. This only needs to be done for the Model 204 job in which you are experiencing the record locking table full condition, not in all jobs.

- For single-user jobs, leave MSRES at the default value of 0.
- For multi-user jobs, CCA recommends setting MSRES to a low, nonzero value, perhaps 32, adjusting LRETBL, if required.

NECBS: Number of Event Control Blocks (ECB)

Class	SYSTEM
Default	0
Setting	On User 0's parameter line
Meaning	<p>The NECBS allocates 256, plus four, bytes of storage for each ECB that you specify. Four bytes for the ECB and 256 bytes for data that may be passed to and from users of the ECBs. Settings range from zero to 2000.</p> <p>The NECBS parameter is referenced by the ECB-related function calls: \$ECBDGET, \$ECBDSET, \$ECBTEST, \$POST, \$UNPOST, and \$WAIT.</p>

NJBUFF: Number of journal buffers

Class	SYSTEM
Default	The default for OS/390, z/OS, and VM is calculated using the formula (NSERVS + NSUBTKS + 1). The default for VSE is one, because VSE does not support multiple journal buffers.
Setting	NJBUFF is a parameter that is passed by the operating system when Model 204 is invoked by the system manager; it cannot be reset.

In OS/390 and z/OS, the NJBUFF parameter can be set explicitly only via the PARM field in the EXEC card. If set, its value should be (NSERVS + NSUBTKS + 1). See the *Model 204 System Manager's Guide*.

In VM, the NJBUFF parameter is stacked in the EXEC that issues all the FILEDEFS.

Meaning The NJBUFF parameter has a dual role of allocating the number of channel programs and journal buffers.

- If NJBUFF=1, only one channel program and one buffer are allocated.
- If you specify a number other than 1 or (NSERVS + NSUBTKS + 1), Model 204 still sets NJBUFF to (NSERVS + NSUBTKS + 1) for the run.

However, if you do specify another number, the number of channel programs for the journal dataset is set to that other number. This might degrade performance if the other number is significantly lower than (NSERVS + NSUBTKS + 1).

- Setting NJBUFF to a value greater than 1 invokes multiple journal buffers and ensures that a free buffer is always available for the journal.

OPSYS: Operating system

Class SYSTEM

Default None

Setting View-only

Meaning The operating system under which Model 204 is running

Valid settings of OPSYS are:

Setting	Meaning
X'800000'	Obsolete bit; ignore.
X'400000'	OS/390 or z/OS (any version). Bit is set whenever environment is z/OS.
X'200000'	IUCV communication available and in use.
X'008000'	VSE operating system
X'004000'	VSE 370 support
X'002000'	VSE 31-bit addressing supported
X'001000'	VM (any version). Bit is set whenever environment is VM. VM Release 7 or earlier, if not combined with X'000400'.

Setting	Meaning
X'000800'	z/OS/ESA (z/OS SP2 or later). This option can be summed with X'400000' or X'800000'.
X'000400'	VM/ESA (CMS Release 8 or later). This option can be summed with X'001000'.
X'000200'	Fujitsu FACOM OSIV, F4 Edition, 40 OS.
X'000100'	Hitachi HITAC VOS3.
X'000080'	CMS on an XC mode machine

RESLTHR: APSY resident load threshold

Class	SYSTEM
Default	-1
Setting	On User 0's parameter line or reset by system manager
Meaning	<p>The RESLTHR parameter specifies the minimum number of times an APSY precompiled procedure must be loaded into a user's server before Model 204 attempts to make the procedure's NTBL and QTBL resident and shareable in virtual storage. Once a procedure's NTBL and QTBL have been made resident, subsequent evaluation of that procedure no longer requires loading those tables into the user's server.</p> <p>The RESLTHR parameter is similar to the RESTHRSH parameter, which sets the minimum number of times a procedure must be swapped, along with the evaluating user, to CCASERVR before the procedure's NTBL and QTBL will be made resident. In a system with little or no server swapping, the number of APSY LOADs (and therefore RESLTHR) provides a better metric for determining whether to make a procedure's NTBL and QTBL resident. Resident requests may reduce CCATEMP I/O by as much as 20 percent and the number of bytes transferred per server I/O by as much as 30 percent.</p>

SCHDOPT: MP/204 scheduler operation and accounting

Class	SYSTEM
Default	0
Setting	On User 0's parameter line or reset by system manager.
Meaning	Control of the main task scheduler operation and accounting for sites with the MP/204 multiprocessing feature

If SCHDOPT is set to X'01', Model 204 tracks the main task scheduler overhead and generates the SCHDCPU statistic. This setting is used when the system manager does not want MP users to be charged for scheduler overhead.

If SCHDOPT is set to X'02', Model 204 forces all processing that can be performed in parallel to run in an offload subtask, even if the main task is not busy. Forced off loading tends to increase CPU usage by the Model 204 Online, but might improve response time, if the processor complex has sufficient excess CPU capacity.

If SCHDOPT is set to X'04', Model 204 defers closing disk pages on a critical file resource and other short-lived waits until the thread is actually swapped. Usually, when a swappable wait is done, Model 204 closes all still-open disk pages for that thread as quickly as possible. However, critical file resource waits tend to be short-lived, making this prompt page closing less efficient.

Valid settings of SCHDOPT are:

Setting	Meaning
X'00'	No scheduler overhead tracking, no forced offload. This is the default.
X'01'	Scheduler overhead tracking, no forced offload.
X'02'	Forced offload, no scheduler overhead tracking.
X'03'	Both scheduler overhead tracking and forced offload.
X'04'	Enable DKPRF - fast DKPR or fast logical read duplicated on next page.

Note: Setting SCHDOPT to 0, the default, changes the way since-last statistics are computed. This change can make it appear that users are using more CPU time than is actually the case. If you want Model 204 to compute user since-last statistics without scheduler overhead, reset SCHDOPT to 1.

SCHDOPT=4 means that deferred pages are released only at the time the user is swapped out. For all other SCHDOPT settings, deferred pages are released when the user is entering swappable wait state, even if server swap never takes place. If there is a chance that swappable wait does not take a lot of time, this option may be usable, because it saves several page reads. You should experiment with this option.

The SCHDOPT parameter is available under OS/390, z/OS, and VM.

Because the difference between doing a fast read of a deferred release page and a regular logical page read is not that great, CCA recommends that you adjust the SCHDOPT parameter setting only if you work in an MP/204 environment. Adjusting it does not have significant impact in other environments.

SYSOPT2: System options

Class	SYSTEM
Default	0
Setting	On User 0's parameter line or reset by system manager
Meaning	<ul style="list-style-type: none">• With SYSOPT2 X'80' <i>off</i>, all dynamic allocations default to use the existing TIOT option unless the XTIIOT option is specified on the DEFINE or ALLOCATE command.• With SYSOPT2 X'80' <i>on</i>, all dynamic allocations that specify the OLD and DIRECT file options use the XTIIOT option, unless the TIOT option is specified on the DEFINE or ALLOCATE command. <p>With the use of dynamic allocation and the XTIIOT option, only the amount of processor storage limits the number of allocated datasets.</p> <p>This parameter applies to only OS/390 and z/OS.</p>

TEMPCUR: Current CCATEMP pages in use

Class	SYSTEM
Default	None
Setting	View-only
Meaning	<p>The TEMPCUR parameter tracks the number of CCATEMP pages currently in use.</p> <p>To ensure that the statistics for TEMPCUR, TEMPMAX, and TEMPSIZE counts are calculated on the same basis, each count includes the first two pages of CCATEMP: the first is the File Control Table; the second is reserved.</p>

TEMPCURE: CCATEMP pages currently in use in the expansion area

Class	SYSTEM
Default	None
Setting	View-only
Meaning	<p>The TEMPCURE parameter tracks the expansion area of CCATEMP by counting the pages currently in use.</p>

TEMPCURS: CCATEMP pages currently in use in the small page area

Class	SYSTEM
Default	None
Setting	View-only
Meaning	The TEMPCURS parameter tracks the small page area of CCATEMP by counting the pages currently in use.

TEMPHIE: Highest page number used in CCATEMP expansion area

Class	SYSTEM
Default	None
Setting	View-only
Meaning	<p>The TEMPHIE parameter tracks the highest CCATEMP page number used in the expansion area. The expansion area pages start at 65,536 (X'10000').</p> <p>The TEMPHIE value is reported on the audit trail during EOJ processing and written to the system statistic block.</p>

TEMPHIS: Highest page number used in CCATEMP small page area

Class	SYSTEM
Default	None
Setting	View-only
Meaning	<p>The TEMPHIS parameter tracks the highest CCATEMP page number used in the small page area. The small page area pages range from 0 to 65,535 (X'FFFF').</p> <p>The TEMPHIS value is reported on the audit trail during EOJ processing and written to the system statistic block.</p>

TEMPMAX: CCATEMP pages high-water mark

Class	SYSTEM
Default	None

Setting	View-only
Meaning	<p>The TEMPMAX parameter tracks the maximum number of CCATEMP pages used in the run.</p> <p>To ensure that the statistics for TEMPMAX, TEMPCUR, and TEMPSIZE counts are calculated on the same basis, each count includes the first two pages of CCATEMP: the first is the File Control Table; the second is reserved.</p>

TEMPPAGE: Number of CCATEMP pages in storage

Class	SYSTEM
Default	0
Setting	On User 0's parameter line
Meaning	<p>Use the TEMPPAGE parameter to place CCATEMP pages for your job into dataspace storage. The number specified for the TEMPPAGE parameter determines how many CCATEMP pages are available for the run. Using the TEMPPAGE parameter also indicates that the pages must be placed in storage rather than on disk.</p> <p>When you specify the TEMPPAGE parameter, even if you have a CCATEMP DD record in the job, the disk file is not accessed and the pages are placed in storage. The default setting of zero means that your job will use a CCATEMP disk file.</p> <p>This parameter applies to only OS/390 and z/OS.</p>

TEMPSIZE: Highest CCATEMP page count

Class	SYSTEM
Default	None
Setting	View-only
Meaning	<p>The TEMPSIZE parameter tracks the highest CCATEMP page number available in the run. CCATEMP page numbers start from zero.</p> <p>To ensure that the statistics for TEMPSIZE, TEMPCUR, and TEMPMAX, counts are calculated on the same basis, each count includes the first two pages of CCATEMP: the first is the File Control Table; the second is reserved.</p>

UFEOHASH: Optimization of FOR EACH OCCURRENCE loops

Class	SYSTEM
Default	0
Setting	On User 0's parameter line
Meaning	The UFEOHASH value represents the number of hash cells that are allocated in the FEO hash table to optimize FOR EACH OCCURRENCE (FEO) loops that are run against unlocked records (FDWOL).

The UFEOHASH parameter can be set to a value between 0 and 65535, inclusive.

- When the UFEOHASH parameter is set to zero, Model 204 does not optimize unlocked FEO loops, which was the behavior of Model 204 prior to Version 5.1. FEO loops that are non-optimized scan from the beginning of the record for each occurrence of the field. This can result in significant CPU overhead for FEO loops run against fields with a large number of occurrences.
- A nonzero UFEOHASH value causes Model 204 to build an FEO hash table. As the value of UFEOHASH is used as a hash value divisor, CCA recommends that you set UFEOHASH to a prime number. A number such as 1021 might virtually eliminate non-optimized FEO loops for unlocked records at the cost of only 8,184 bytes of virtual storage.

Understanding the FEO hash table

An FEO table entry contains a file number and record number for a record. Every Table B record is tracked in the FEO table. Whenever a record is modified, the FEO table entry for that record is also updated. FEO processing remembers the FEO table value for the record it is processing. The current table information for a record is checked before each new scan for another field occurrence. If the table information is unchanged, the next scan starts on the record where the previous scan ended. If the table information changed, the next scan must begin at the start of the record.

It is possible that more than one record can map to the same hash cell in the FEO table. Consequently, an update to another record, can cause a scan from the beginning of the record. Each time a record is scanned from the beginning in response to a change in the value in the FEO table, the UFEOMISS statistic is increased by one.

This optimization is disabled in single-user jobs.

Usage	Setting the value of UFEOHASH should take into account the number of concurrent updates that happen in the job.
--------------	---

Updates to records in different files may hash to the same FEO slot, as can updates to different records in the same file. These are called **hash collisions**.

As the value of UFEOHASH is increased, the likelihood of these hash collisions is reduced. Up to a certain point, increasing the value of UFEOHASH tends to improve performance by reducing the number of FEO loops that are not optimized, which are also referred to as unlocked FEO misses. See “UFEOMISS: Number of unsuccessful optimizations for FEO” for related information.

UFEOMISS: Number of unsuccessful optimizations for FEO

Class SYSTEM

Default 0

Setting View-only

Meaning The UFEOMISS parameter indicates the number of times Model 204 tried to optimize an unlocked FOR EACH OCCURRENCE loop, but could not because of updating activity. A high value for the UFEOMISS parameter suggests that you might improve performance by setting the UFEOHASH parameter to a higher value.

The UFEOMISS parameter is always zero, if the UFEOHASH parameter is zero. See “UFEOHASH: Optimization of FOR EACH OCCURRENCE loops” on page 1-95 for related information.

Enhanced commands

The following commands will be copied to the *Command Reference Manual* for Version 5.1 commercial release.

BUMP MODULE: Logging out a user from an external module

See “Model 204 commands relating to ECF” on page 3-4 for a description.

CHECKPOINT: Handling a checkpoint

The following CHECKPOINT commands may be issued by any system manager or from the operator’s console:

Syntax

```
CHECKPOINT [ [ ABORT | MESSAGE ] |  
            [ [ SET | UNSET | END ] EXTENDED QUIESCE ] ]
```

Where:

- CHECKPOINT command with no arguments requests that Model 204 perform a checkpoint. May be issued by any user who is logged in.
- ABORT keyword aborts a pending request for a checkpoint.
- MESSAGE keyword displays the status of the most recent checkpoint.
- SET EXTENDED QUIESCE keywords place the Online into an extended quiesce immediately after the next successful checkpoint. Once placed into an extended quiesce, no file updating can take place until the extended quiesce ends. The checkpoint that is taken at the end of Model 204 initialization or recovery is not available for extended quiesce processing.

The command is ignored if the Online is already in an extended quiesce. Issuing the command multiple times is the same as issuing it once.

If you define a ring stream journal that has an OFFLOAD dataset defined, then when the extended quiesce begins, Model 204 issues internally an OFFLOAD STREAM command for the ring stream journal.
- UNSET EXTENDED QUIESCE keywords reverse the effect of the SET EXTENDED QUIESCE option. The command is ineffective if the:
 - Online is in an extended quiesce state
 - CHECKPOINT SET EXTENDED QUIESCE command has not been issued

Issuing this command multiple times is the same as issuing it once.

Issuing a CHECKPOINT UNSET EXTENDED QUIESCE command when the Online is not yet available for extended quiesce processing, results in the following error:

M204. 2612: CHECKPOINT COMMAND UNSUCCESSFUL - reasons

- END EXTENDED QUIESCE keywords terminate an extended quiesce. This command also restarts the checkpoint pseudo-subtask and file updating can resume in the Online.

If the parameter CPQZSECS=*nnnn* is specified and neither a CHECKPOINT SET EXTENDED QUIESCE nor CHECKPOINT UNSET EXTENDED QUIESCE command is issued within *nnnn* seconds of the start of an extended quiesce, the action specified on the parameter CPQZACTN is initiated.

- See “CPQZSECS: Maximum duration, in seconds, of an extended quiesce” on page 1-77 for more details.
- See “CPQZACTN: Action to take when the CPQZSECS time limit expires” on page 1-76 for more details.

Usage

Both the CHKABORT command and the CHKMSG command are still valid.

The security required for the ABORT and MESSAGE keywords is the same as for the CHKABORT and the CHKMSG commands.

When a CHECKPOINT command with an END, SET, or UNSET keyword is issued in a valid context, the following message is displayed:

M204. 2611: CHECKPOINT SET/UNSET/END COMMAND SUCCESSFUL

When a CHECKPOINT command with a SET, UNSET, or END keyword is issued in an invalid context, the following message is displayed:

M204. 2612: CHECKPOINT SET/UNSET/END COMMAND UNSUCCESSFUL - reason

CHKMSG: Returning the checkpoint status

Privileges	System manager
Function	Displays the status of the most recent checkpoint
Syntax	CHKMSG
Usage	Output from the CHKMSG command can include status information on an extended quiesce.

MONITOR CFR: Monitoring critical file resources

Privileges	System administrator
-------------------	----------------------

Function To display each user that currently holds an enqueue on a critical file resource.

Syntax MONI TOR CFR

The MONITOR command now supports the CFR option, so you can display the following information regarding critical resource enqueues. The display consists of:

- User number
- File
- Time of acquisition
- Lock type—share or exclusive

Since the CFR resource is a fleeting enqueue, an empty list is commonly displayed. The display of the following critical file resource use is possible only if CFRLOOK=1 is specified on User 0's parameter line.

Column heading in display	Critical file resource held
DIRECT	Table B
EXIST	Existence bitmap
INDEX	Ordered index
RECENQ	Record enqueueing table

Usage If you issue a MONITOR CFR command, but have not set CFRLOOK=1, the following message is displayed:

```
M204. 1925: CONFLI CT OPTI ON REQUI RES PARAMETER  
CFRLOOK= 1
```

Example You can use the MONITOR CFR command to detect critical file resource conflicts in long-running, complex searches against large Ordered Indexes. You can issue a BUMP *userid* command or STOPU command or correct the inefficient program.

For example, if USERNO 2 executes a program that includes either of the following User Language statements, all values of the Ordered Index for the field referenced are checked. Processing the statement holds some critical file resources for the file for a long time; update users are delayed until the search completes.

```
FD ordered-numeric-field-1 IS GT
```

or

```
FD ordered-character-field-1 IS LIKE *ABC
```

If at the same time, USERNO 3 uses a program that includes the following User Language syntax, this user will be delayed until USERNO 2's search completes.

CHANGE ordered-character-field-2 TO 'ABC'

If a system manager enters the following MONITOR CFR command, the following results might be displayed. The results illustrate USERNO 2 accessing the VEHICLES file and USERNO 6 accessing the DSNLIST file.

MONITOR CFR

USERNO	USERID	FILE	RECENQ	EXIST	INDEX	DIRECT
2	SUPERKLUGE	VEHICLES		S>16:41:53	S>16:41:53	
6	CCA01	DSNLIST			18:59:45	

Note: The S> characters before a timestamp indicates a shared lock type. A white space before a timestamp indicates an exclusive lock.

MONITOR: An Online system

Privileges	System administrator
Function	Monitor the status of checkpoint extended quiesce
Syntax	MONITOR
Usage	Two new wait codes have been added. See "Additional wait types" on page 2-13 and also "\$WAIT function" on page 1-49.

Displaying the RUNG queue

Various MONITOR commands display the scheduler queues in which each user currently resides: REDY, BLKI, BLKO, SWPG, WTSV and OFFQ. There is another queue, the running or RUNG queue. When you issue one of these MONITOR commands, these queues appear as column heads. Following the column heads, the accumulated information for a period of time is displayed. Next the detail of which users are in which scheduler queue is displayed.

- Prior to Version 5.1, the user who issued the MONITOR command was described in the detail section as REDY (ready).
- Starting in Version 5.1, the user who issues the MONITOR command is counted as a running user and is shown in the detail section as being in the RUNG queue. In an MP/204 environment, multiple users may be running simultaneously; those users, if any, are also shown as being in the RUNG queue.

MONITOR: Checkpoints

Privileges System Administrator

Function Enables system administrators to verify the date and time of the last successful checkpoint. Other valuable information, such as the number of time-outs since the successful checkpoint, the number of users preventing checkpoints, and the current size of CHKPOINT, can be displayed.

As of Version 5.1, you can also track checkpoint extended quiesce state information.

Syntax MONI TOR { CHECKPOI NT | CHKP } [USERLI ST] [SL] [EVERY *n*]

Usage New information about the checkpoint extended quiesce status is available.

Examples The following examples illustrate the messages regarding extended quiesce that you might see after issuing a MONITOR CHECKPOINT command.

The following output reflects that the system entered and exited a checkpoint extended quiesce interval.

```
*** LATEST SUCCESSFUL CHECKPOINT COMPLETED AT:
01. 239. 13: 57: 18. 37
*** 0 CHECKPOINT HAVE SINCE TIMED OUT
*** 2 RECORDS CURRENTLY IN CHKPOINT STREAM
*** 1 CHECKPOINTS CURRENTLY IN CHKPOINT STREAM
*** 0 USERS CURRENTLY HAVE CHECKPOINT INHIBITED
*** SYSTEM ENTERED EXTENDED QUIESCE AT: 08/27/01 13: 57: 18,
SYSTEM EXITED EXTENDED QUIESCE AT 08/27/01 13: 57: 49, REASON = END
COMMAND ISSUED BY USER NUMBER: 2
```

The following output reflects that the system entered and remains in a checkpoint extended quiesce interval. No updates are allowed.

```
*** LATEST SUCCESSFUL CHECKPOINT COMPLETED AT:
01. 239. 13: 57: 18. 37
*** 0 CHECKPOINT HAVE SINCE TIMED OUT
*** 2 RECORDS CURRENTLY IN CHKPOINT STREAM
*** 1 CHECKPOINTS CURRENTLY IN CHKPOINT STREAM
*** 0 USERS CURRENTLY HAVE CHECKPOINT INHIBITED
*** SYSTEM ENTERED EXTENDED QUIESCE AT: 08/27/01 13: 57: 18,
ALL FILE UPDATING REMAINS SUSPENDED
```

MONITOR DATASPACE: Display dataspace and hiperspace

Privileges System administrator

Function Enables the system administrator to display the use of dataspace and hiperspaces when tracking the use of the CCATemp and CCASERV In Storage feature

Syntax MONITOR DATASPACE

Example The DATASPACE option of the MONITOR command displays dataspace and hiperspace information as shown in the following example.

NAME	DATASPACE TYPE	4K PAGES	PAGE HWM	EXTRA DATASPACE
CCASERV	HI PERSPACE - SCROLL	912	392	
CCATemp	DATASPACE	3474	2396	
CCAAPSY	HI PERSPACE - CACHE	52	36	

NAME	READS	WRITES	PAGES READ	PAGES WRITN	SLOWRD	SLOWWR	PAGEF
CCASERV	12	21	357	687	0	364	0
CCATemp	1561	1575	2357	2378	0	0	0
CCAAPSY	2	3	13	33	0	33	0

The explanation of the headings in the previous example are:

Heading	Meaning
NAME	Dataspace or hiperspace name
DATASPACE TYPE	Type may be one of the following: <ul style="list-style-type: none"> Dataspace Hiperspace scroll Hiperspace cache Please see "DSPOPT: Data- and hiperspace options" on page 1-79 for explanation of terms.
4K PAGES	Number of 4K pages in a dataspace or a hiperspace
PAGE HWM	The highest 4K-page number used in a dataspace or a hiperspace
EXTRA DATASPACE	List of additional dataspace names allocated to accommodate the total number of pages. Each dataspace or hiperspace may not have more than 524288 4K pages. Additional dataspace or hiperspace names are constructed as CCA00001, CCA00002
READS	Number of read operations against a dataspace or a hiperspace
WRITES	Number of write operations against a dataspace or a hiperspace
PAGES READ	Number of 4K pages read from a dataspace or a hiperspace
PAGES WRITN	Number of 4K pages written to a dataspace or a hiperspace
SLOWRD	Number of 4K pages read from a dataspace or a hiperspace without the Move Page instruction. Please see "DSPOPT: Data- and hiperspace options" on page 1-79 for more explanation.

Heading	Meaning
SLOWWR	Number of 4K pages written to a dataspace or a hiperspace without the Move Page instruction. Please see "DSPOPT: Data- and hiperspace options" on page 1-79 for more explanation.
PAGEF	Number of page faults when reading 4K page from a cache hiperspace. The statistics indicates how many times Model 204 had to do a disk read due to a 4K page fault in a cache hiperspace

Usage

Among things monitored are page usage high-water marks, number of calls to the dataspace data movement routines, the quantity of data moved to and from the dataspaces, and page-fault statistics for dataspace data movement.

See also, "Sample output from a MONITOR DATASPACE command" on page 1-26.

MONITOR SERV: Display server number, SERVSZ and LSERVER

The MONITOR command provides information regarding servers defined to the Online. SERVSZ and LSERVER are displayed for each server defined and, if the server is currently occupied by a user, standard user information is also provided. The following syntax is supported:

MONI TOR SERV

MONI TOR SERV SL

MONITOR (s1,s2,s3...) SERV displays information for s1,s2,s3... only

MONI TOR CHKP USERLI ST SERV

MONI TOR CHKP USERLI ST SERV SL

Only servers, in order by definition, are reported. If NSERVS=20, then only twenty lines of output are produced. If any server is occupied by a user, standard user information is also provided:

USER	SVR	USERID	SERVSZ	LSERVER	FUNC	CNCT	CPU	SEQIO	QUE	WT	FLGS
	1		111000	77400					WTUS		
	2		222000	117000					WTUS		
5	3	CCAD01	128800	28792	EVAL	320	0.308	25	REDY		
1	4	CCAD02	777000	97296	DI SP	1	0.000	8	BLKI	3	60
	5		85000	85000					WTUS		
6	6	SUPERKLUGE	666000	117000		25	0.021	3814	RUNG		
9	7	CCAB08	777000	747000	EVAL	0	0.202	3	BLKI	24	68
8	8	USER01	840000	47016	EDI T	94	0.003	81	BLKI	3	66

This display shows all servers (NSERVS=8), in order by definition, defined to this Online. Included on each line are the SERVSZ for each server and the current number of bytes in use - LSERVER. Three servers, 1, 2, and 5, are waiting for user WTUS.

MONITOR: The subsystem

The MONITOR SUBSYSTEM command now displays the number of CCATEMP pages required to save precompiled procedures. When you include the PROCCT or ALL option in the command, the following line is displayed:

```
CCATEMP PAGES USED FOR PRECOMPILED PROCEDURES:  n
```

When the MONITOR SUBSYSTEM command is issued with options that result in more than one line of information per subsystem, a blank line is displayed between each subsystem, to assist in readability.

MSGCTL: Setting message output

You can now specify the following options as a *msg-option* on the MSGCTL command:

Option	Action
AUDITMS	Causes the specified error message to appear on the audit trail as an MS line.
NOAUDIT	Suppresses auditing of the specified error message.

Caution: CCA strongly recommends that you do not change or suppress the COUNT option that is assigned to some messages. The User Language compiler depends on using the COUNT option to track errors in your program. If no errors can be reported, your program will seem to compile only to abend during the run. Also, the lack of messages in your Model 204 journal, audit trail, or other job output can result in the inability to trace and diagnose problems.

RENAME FILE: Renaming a file

Privileges File manager

Function Change the name of a file

Syntax RENAME FILE *old-filename new-filename*

Where

- The *old-filename* specifies the name you want to discontinue using for this file.
- The *new-filename* specifies the name you want to use for the file, instead of *old-filename*.

Example You can use the following sequence of commands to rename a file named CANINES to DOGS.

```
ALLOCATE DOGS LIKE CANINES /* optional ALLOCATE */  
/* under only OS/390, */
```

BUMP FILE CANINES	/* z/OS and VM	* /
STOP FILE CANINES	/* optional	* /
STOP FILE DOGS		
RENAME FILE CANINES DOGS		
FREE CANINES		
START DOGS		
OPEN DOGS		

After the above sequence is successfully completed, the file, CANINES, is closed. The file will now exist as file named DOGS. It is the user's responsibility to change the corresponding JCL, if needed, to the proper DDNAME.

The DKRD and DKWR statistics are updated to indicate the number of pages that were renamed.

Usage

The RENAME FILE command lets you change the name of a file without using a DUMP or RESTORE command.

Before you issue a RENAME FILE command, the file you want to rename must be open and a stop issued against both the old and new file names. Only you can have the old and new file names open while the rename is processing. You must ensure that other users are not using the file. You can use the BUMP FILE command to remove user access to the file.

The file you rename should not be an MVS temporary dataset since the dataset name is not available for dynamic allocation of the new DDNAME. In addition, CMS formatted files are not supported.

The RENAME FILE command requires 49,868 bytes of below-the-line, temporary storage. You may need to adjust the SPCORE parameter accordingly.

After processing a RENAME FILE command, the DKRD and DKWR statistics reflect the number of renamed pages.

Note: The rename takes place outside of recovery and a discontinuity is logged across which ROLLBACK recovery cannot occur. Should RENAME FILE processing fail to complete, the file is not usable until another successful RENAME FILE command is completed. See the *Model 204 File Manager's Guide* for a full discussion of file discontinuities.

VIEW SERVSIZ: Display the size of the server

You can display the size of the server, in bytes, in which you currently reside. If multiple server sizes were defined for an Online, the results of VIEW SERVSIZ may be different for the same user as that user is swapped between different sized servers.

VIEW LSERVER: Display number of bytes in use

You can display the number of bytes currently in use in your server. The number will be less than or equal to SERVSZ.

New and changed statistics in Version 5.1

Additional statistics have been added to the end of the system, user, and since-last statistics blocks. Table 1-3 lists the new statistics for Version 5.1. Each statistic includes the description of what the statistic reports, followed by the position in the System-final, user-logout, user-since-last, SMF-logout, and/or SMF-since-last journal record layout as they apply.

If you have written any functions or other tools that use the STSYS, STSSYM, STUSER, or STUSYM COPY members, you must reassemble them. System-partial and user-partial offsets are the same as system-final and user-logout offsets, respectively.

Table 1-3. New statistics for Version 5.1

Statistic	Specifies...
APSYLD	Number of APSY loads
APSYLDD	Number of APSY loads from a dataspace
APSYLDT	Number of tiny APSY loads
DKPRF	Number of DKPRs that were satisfied by pending or deferred close buffers
DKSRR	Number of times a page that was expected to be in the buffer pool could not be located, which necessitates a physical I/O to disk
DKSRRFND	Number of times that a page that was expected to be in the buffer pool was located there, which eliminates the need for a physical I/O
ECCALL	ECF - Number of CALLs System-final(X'290') User-logout(X'160') User-SL(X'154') SMF-logout(X'174') SMF-SL(X'1A8')
ECCNCT	ECF - Elapsed time for external module System-final(X'2A8') User-logout(X'178') User-SL(X'16C') SMF-logout(X'18C') SMF-SL(X'1C0')
ECCTOUT	ECF - Number of CALLs that timed out (module or subtask unavailable) System-final(X'2A4') User-logout(X'174') User-SL(X'168') SMF-logout(X'188') SMF-SL(X'1BC')

Table 1-3. New statistics for Version 5.1 (continued)

Statistic	Specifies...
ECCWAITM	ECF - Number of calls that waited for module to become available System-final(X'294') User-logout(X'164') User-SL(X'158') SMF-logout(X'178') SMF-SL(X'1AC')
ECCWAITS	ECF - Number of CALLs that waited for a subtask to become available System-final(X'298') User-logout(X'168') User-SL(X'15C') SMF-logout(X'17C') SMF-SL(X'1B0')
ECDELETE	ECF - Number of DELETES System-final(X'28C') User-logout(X'15C') User-SL(X'150') SMF-logout(X'170') SMF-SL(X'1A4')
ECLOAD	ECF - Number of LOADs System-final(X'288') User-logout(X'158') User-SL(X'14C') SMF-logout(X'16C') SMF-SL(X'1A0')
ECNAMMAX	ECF - High-water mark of call names defined System-final(X'2B0')
ECMODMAX	ECF - High-water mark of modules loaded System-final(X'2AC')
ECTSKMAX	ECF - High-water mark of subtasks active System-final(X'2B4')
ECTWAITM	ECF - Time spent waiting for module to become available System-final(X'29C') User-logout(X'16C') User-SL(X'160') SMF-logout(X'180') SMF-SL(X'1B4')

Table 1-3. New statistics for Version 5.1 (continued)

Statistic	Specifies...
ECTWAITS	ECF - Time spent waiting for a subtask to become available System-final(X'2A0') User-logout(X'170') User-SL(X'164') SMF-logout(X'184') SMF-SL(X'1B8')
GTBLRS	Number of GTBL rearrangements required to add a global object System-final(X'2BC') User-logout('180') User-SL(X'174') SMF-logout(X'194') SMF-SL(X'1C8')
GTBLRU	Number of GTBL rearrangements required to add a global string variable System-final(X'2B8') User-logout('17C') User-SL(X'170') SMF-logout(X'190') SMF-SL(X'1C4')
TEMX	High-water mark of CCATEMP expansion area System-final(X'2C4')
TSMX	High-water mark of CCATMEP small page area System-final(X'2C0')

Since-last statistics

Statistics are kept for each User Language request, host language interface call, and a number of Model 204 system commands. When each request or command completes, a since-last statistics line is written. The line includes the ACCOUNT parameter at the beginning, followed by the LAST parameter:

Several MONITOR commands produce displays containing the user's last or current activity in the FUNC column. When this activity completes, statistics are also written to CCAAUDIT and the activity is described in ST lines as 'LAST=activity'.

Syntax ST \$\$\$ USERID=' *userid*' ACCOUNT=' *account name*'
LAST=' *acty*'

Where *acty* can be one of the following. The CREA, INCR, and INIT activity types are new in Version 5.1:

Table 1-4. Activity types

Option	Meaning
BLDX	Z command
CMPL	Compilation of a User Language request
CPYP	COPY PROCEDURE command
CREA	CREATE FILE <i>filename</i> command
DISP	DISPLAY PROCEDURE command
DUMP	DUMP file command
EDIT	Editing a request
EVAL	Evaluation of a User Language request
EXEC	Execution of an SQL request
FLDC	Compilation of a FLOD procedure
FLDE	Evaluation of a FLOD procedure
IMPC	Implied commit
INCR	IN FILE <i>filename</i> INCREASE DATASETS command
INIT	IN FILE <i>filename</i> INITIALIZE command
LOAD	Loading a precompiled (ASPY) procedure
PREP	Compilation of an SQL request
REST	RESTORE file command

2

Compatibility Issues

In this chapter

This chapter contains installation, migration, and compatibility information pertinent to Model 204 Version 5.1. Please read through the material for the latest information on contents and installation before you start the actual installation.

- Recovery incompatibilities
- Installation and reinstallation compatibility issues
- Sizing considerations
- Commands and parameters incompatibilities
- User Language incompatibilities
- Altered state information
- All operating systems
- For OS/390 and z/OS operating systems
- For VM operating system
- For VSE operating system

Recovery incompatibilities

RESTART recovery incompatibility

The new RESTART recovery feature, which automates secondary recovery, introduces upward incompatibility issues. If for your current version of Model 204, you maintain JCL or EXECs that automatically rotate CHKPOINT and RESTART datasets for secondary recovery, or you maintain RESTART ROLL BACK-only jobs with no CHKPOINT dataset definition, these jobs must be purged or modified for use with Model 204 Version 5.1. You need to educate your staff regarding the changes introduced in RESTART recovery.

Changes described in “Operational and performance improvements in recovery” on page 1-8 cause the following upward incompatibilities:

- You no longer change the CHKPOINT and RESTART datasets when running secondary recovery.
- A CHKPOINT dataset is now required for ROLL BACK-only processing.

RESTART ROLL FORWARD recovery is version specific

Version 5.1 ROLL FORWARD recovery is *not* compatible with Model 204 journals from previous releases. All ROLL FORWARD journal records, types 1–6, have been reformatted. If ROLL FORWARD processing is attempted using a pre-Version 5.1 journal using a Version 5.1 Online (Batch204), processing will be terminated with the following error:

```
M204. 2501:  RELEASE  I NCOMPATABI LI TY
```

ROLL BACK processing remains compatible with previous releases. CCA recommends that you backup all files prior to installing Version 5.1. **Also, since the format of all update journal records is now eight bytes larger than previous releases, the journal may need additional space.**

Installation and reinstallation compatibility issues

Softspy

If you plan to use SoftSpy, you must install the SoftSpy distributed with Model 204 Version 5.1. Earlier versions of Softspy do not run with Model 204 Version 5.1.

Additional information in CCAPRINT

At Model 204 initialization, a listing of all Early Warnings that have been applied to the module being executed is produced. It is comparable to the output of the following command:

```
DI SPLAY EW ALL
```

The output is directed to CCAPRINT in Online, Batch204, and IFAM4 jobs. A listing of all maintenance currently applied to the active load module or phase is displayed. Messages of the following type are produced in CCAPRINT, depending on the maintenance history at your site.

Examples NO EARLY WARNINGS HAVE BEEN APPLIED
EARLY WARNING 1 TO 4 APPLIED
EARLY WARNING 6 APPLIED
EARLY WARNINGS 9 TO 12 APPLIED

This listing helps both you and CCA Customer Support reduce the time required for problem resolution. You should have both CCAPRINT and CCAAUDIT available when you report problems to CCA Customer Support.

Subsystems available at initialization

If the CCASYS file is marked as full, FISTAT=X'08', during Online initialization, subsystems are now available and the following message is no longer issued:

```
M204. 1457 UNABLE TO SCAN LIST OF SUBSYSTEM NAMES
```

CCACAT field attribute change

All CCACAT field KEY attributes were changed to ORDERED. Please see the installation instructions for your operating system.

Starting in Model 204 Version 5.1, all KEY field attributes in the CCACAT file are being changed to ORDERED NUMERIC or ORDERED CHARACTER, as appropriate. This change is part of the reinstallation process and does not affect any data that is currently stored in the CCACAT catalog file.

Use of the ORDERED attribute speeds up data retrieval. It is also recommended for use with all fields which participate in SQL selection qualification, aggregate functions and queries using DISTINCT processing.

Do not use the MERGCUST job

For OS/390 and z/OS only, CCA recommends that you bypass using the MERGCUST job, because of the multiple changes to INSPARMS in Version 5.1. Instead, CCA recommends that you edit the Version 5.1 INSPARMS member and make changes using your old INSPARMS as a guide.

Sizing considerations

CCA recommends that you run your applications to examine the since-last statistics. Make your adjustments based on the following known increases and the decreases described in “Performance enhancements” on page 1-57.

Increasing LHEAP

In this release of Model 204, use of LHEAP may be increased in some cases. The increase may be 40 to 50 percent. The LHEAP parameter is used by only SQL threads: IODEV=13 and IODEV=19.

Changes in QTBL and VTBL requirements

Increasing QTBL

An QTBL increase of three entries may be needed to support application subsystem users.

Increasing VTBL

The VTBL record variables now require an additional eight bytes to support the subscripted field extraction performance improvement. This increase in VTBL will be seen when storing or updating a record. See “Subscripted field extraction” on page 1-61.

Some VTBL entries related to \$functions are now reusable. This enhancement may reduce your VTBL requirements.

Each FIND ALL VALUES or FOR EACH VALUE OF 'fieldname' statement generates a value set header of 48 bytes. This is an increase of 28 bytes.

Increase in STBL

If you set the MAXINCL parameter to a value greater than five, and you run a request with \$RDPROC call, the request will use more STBL space.

Increase in server size

Server sizes have generally increased by 48 bytes.

PDL usage increases

For Version 5.1, be prepared to increase the use of PDL by 276 bytes for the VALUE IN clause for FIND ALL VALUES, as compared to Version 4.2.

CCAJRNL size increase

Beginning in Version 5.1, Model 204 can detect if an update unit was improperly written or duplicated. The additional information in the records may cause a modest increase in the size of CCAJRNL. See “In Chapter 15: Using a System Journal and Statistics” on page 4-5.

CCATEMP usage

If a USE PROCEDURE command or \$BLDPROC User Language statement specifies a temporary procedure name, the output is directed to CCATEMP. Previously, if the operation caused CCATEMP to fill, the user was not notified. Now, the request is cancelled and the last page of the new procedure is kept to assist the programmer in debugging the application. The following message is issued:

```
M204. 0441: CCATEMP FULL: "USE PROCEDURE" COMMAND
```

or

```
M204. 0441: CCATEMP FULL: $BLDPROC
```

Note: This means that after the request is cancelled and the last page of the new procedure is kept. The procedure does not contain everything up to the point of failure.

The actual CCATEMP requirements did not increase; the change is: the user is notified.

Increase in file page buffer minimum

Starting in Version 5.1, the MINBUF parameter is automatically set to its smallest allowed value, 35. See “MINBUF: File page buffer minimum” on page 1-86 for a description and formula to calculate the minimum allowed.

Commands and parameters incompatibilities

REDEFINE command and INVISIBLE fields behavior change

The behavior of the REDEFINE command changed. Prior to Version 5.1, when you redefined an invisible field with options that added a new index, such as KEY, NUMERIC RANGE, or ORDERED, the redefinition for the field was accepted. However, the new index was not actually built (and could not be because the field was invisible). Now, if you redefine an invisible field with options that specify a new index for the field, the redefinition is rejected with the following message:

```
M204. 0411: CONFLI CTING ATTRI BUTES: I NVI SI BLE FI ELD  
WI TH NEW I NDEX
```

The following exception applies: in the case where no records have ever been stored in the file, and that case only, redefining an invisible field may specify options that add a new index for the field.

The behavior of the REDEFINE command does not change when used with invisible fields and options that do not add a new index for the field.

Backpaging count now accurate

In Version 5.1, the NBKPG parameter reflects the actual number of pages to use in backpaging. Prior to Version 5.1, the number of pages was always one more than the specified value. For example, if NBKPG=5 then, 'P -5' retrieved the oldest page. In Version 5.1, 'P -5' equals 'P -4'. You cannot retrieve the sixth backpage.

Changes to CHKABORT and CHKMSG commands

The CHKABORT and CHKMSG commands were changed so that any outstanding transaction is ended and the USE=IGNORE attribute applies to the command.

MSGCTL command behavior change

Starting in Version 5.1, the AUDITAD, AUDITMS, AUDITRK, and COUNT options are mutually exclusive. If multiple options are specified, the last one entered prevails.

If you remove or cancel the message type assigned to a message, NOAUDITAD, NOAUDITRK, or NOCOUNT, the default message type is AUDITMS.

Prior to Version 5.1, the MSGCTL message options were not mutually exclusive. If multiple options were entered for a message, the order of precedence was COUNT, AUDITRK, and then AUDITAD. If you want to

disallow the new Version 5.1 message options and allow MSGCTL command processing to operate as it did prior to Version 5.1, you can specify CUSTOM=7 in the CCAIN parameters or in a RESET command.

The AUDITMS and NOAUDIT options are new in Version 5.1. See “MSGCTL: Setting message output” on page 1-104 for details.

User Language incompatibilities

Changes to pattern matching

All characters X'00' through X'FF' are valid in a pattern presented to the User Language pattern matcher and the \$CHKPAT function.

As a result, all characters are treated as valid, literal characters. The following error messages are no longer invoked for these characters.

M204. 1688: error type IN PATTERN 'pattern' AT CHARACTER char

M204. 1689: error type IN PATTERN 'pattern' AT CHARACTER char

Continuing use of the escape character

The escape character, which is the exclamation point (!), allows a pattern matching character to be interpreted as a literal character rather than as a pattern character. The escape character affects only the next character.

Pattern character	Usage as a character
*	Wildcard
+	Placeholder for only one character
,	Or
()	Set beginning and ending
-	Range
/	Repeat
!	Escape
=	Hexadecimal
#	Numeric digit
@	Alphabetic

The eleven special pattern characters listed in the previous table continue to invoke special pattern operations, unless preceded by the escape character (!).

Note: Patterns containing embedded escape characters preceding special characters function as they did in earlier releases of Model 204.

\$FSTERR and \$ERRMSG changes

The \$FSTERR and \$ERRMSG functions return the first counting error and the most recent counting error in this session, respectively. These values are

usually preserved across requests. However, prior to Version 5.1 the User Language OPEN statement incorrectly cleared \$FSTERR and \$ERRMSG (in contrast, the OPEN command correctly left \$FSTERR and \$ERRMSG unchanged). This was changed in Version 5.1: an OPEN statement no longer clears \$FSTERR and \$ERRMSG. To clear the messages accessed by \$FSTERR and \$ERRMSG, use the new \$ERRCLR function. See “\$ERRCLR function” on page 1-44.

IF statements generating compile errors

Prior to Version 5.1, an incorrectly coded IF statement might not generate errors.

Beginning in Model 204 Version 5.1, the following incorrectly coded IF statement generates errors. For example:

```
IF %N =
```

The following errors are invoked:

```
*** 1 M204. 0052: I LLEGAL TERM
*** M204. 1042: COMPI LATI ON ERRORS
```

FIND statement performance enhancement

When a FIND statement refers back to a previous foundset and the foundset is empty, then the loop for the FIND statement is skipped. Performance for the FIND is enhanced because no retrieval is performed for the criteria specified in the FIND statement.

Each FIND statement, however, for this particular case, requires one more word (four bytes) of QTBL space.

Using Language Environment mathematics \$functions

Beginning with Version 5.1, if you want to use the Model 204 mathematics \$functions, you can install either the FORTRAN runtime libraries or use the IBM LE runtime libraries. Model 204 now includes an interface to the LE libraries. As in prior releases of Model 204, you are not required to use the mathematics \$functions. Consult the Model 204 installation guide for your operating system.

CCA recommends that you use the FORTRAN library for applications depending on mathematical functions performance.

Changes required to user-written \$functions

In Version 5.1, \$function operations were optimized to create CPU, QTBL, and VTBL savings. Due to these changes the RESULT macro does not work in Model 204 Version 5.1. Change any use of the RESULT macro in \$functions to use the standard LEAVENUM, LEAVEF0 and LEAVESTR macros.

Altered state information

Changes in information messages affecting location offset

Starting with Model 204 Version 5.1, the following information messages include the M204.*nnnn*: prefix. They appear in MONITOR VTAM output and in the termination section of CCAJRNL or CCAAUDIT.

M204. 0079: INPUT BUFFER HIGH WATERMARK - %C
M204. 1053: BUFFER USAGE STATISTICS FOR VTAMNAME %C
M204. 1055: NUMBER OF INPUT BUFFER WAITS - %C
M204. 1056: NUMBER OF WRITE WAITS - %C
M204. 1057: %C OUT OF %C INPUT BUFFERS ARE ACTIVE
M204. 1058: %C OUT OF %C OUTPUT BUFFERS ARE ACTIVE
M204: 1059: NUMBER OF TERMINALS IN OUTPUT BUFFER WAIT QUEUE IS %C
M204: 1890: NUMBER OF WRITE BUFFER WAITS - %C
M204: 1891: WRITE BUFFER HIGH WATERMARK - %C
M204: 1892: NUMBER OF CLSDST WAITS - %C
M204: 1893: NUMBER OF RESPONSE WAITS - %C
M204: 1895: NUMBER OF OPNDST WAITS - %C
M204: 1896: NUMBER OF STORAGE SHORT RETRIES - %C

If you run tasks that look for these messages based on their position, you will get unexpected results.

MONITOR and MONITOR SL commands output reformatted

The output from the MONITOR and MONITOR SL commands has been slightly reformatted to accommodate larger statistics. Specifically, the columns for the statistics DKRD, DKWR, UDD, OUT, FINDS and RECDS have each been expanded by 1 byte. This causes the entire line to be right shifted such that RQTM is shifted right by a total of 6 bytes, as shown in Figure 2-1 on page 2-12.

The CNCT column was expanded from 5 to 6 bytes without changing its line format. If CNCT is 6 bytes, there is no space between the left most byte of CNCT and the right most byte of FUNC.

New format:

```

USER SVR   BUF FLS   PCPU  SMPLS   RUNG   REDY   BLKI   WTSV   BLKO   SWPG
        67   4 0. 177

        6 new spaces distributed   no space
USER SVR USERID   P CUR   SLICE AGE FUNC CNCT   CPU  SEQIO QUE  WT FLGS
CPU  DKRD  DKWR   UDD   OUT  SLIC FINDS  RECDS  PCPU      RQTM
SCRN  SVRD SVPAGES SUBSYSTEM  PROC-FILE PROC
      COMMI TS      BACKOUTS  UPDTIME(MS)  LNGUPDATES  LNGUPDTIME(MS)
  
```

Old format:

```

USER SVR   BUF FLS   PCPU  SMPLS   RUNG   REDY   BLKI   WTSV   BLKO   SWPG
        26   2963  49 0. 733   15   0. 000  0. 000  22. 866  0. 000  4. 000  0. 000
USER SVR USERID   P CUR   SLICE AGE FUNC CNCT   CPU  SEQIO QUE  WT FLGS
CPU  DKRD  DKWR   UDD   OUT  SLIC FINDS  RECDS  PCPU      RQTM
SCRN  SVRD SVPAGES SUBSYSTEM  PROC-FILE PROC
      COMMI TS      BACKOUTS  UPDTIME(MS)  LNGUPDATES  LNGUPDTIME(MS)
  
```

Figure 2-1. Comparing MONITOR and MONITOR SL commands display with the former display

PRIORITY command output reformatted

The output of the PRIORITY command now includes a header line which indicates the meanings of the statistics in the user lines that follow. Also, the user detail lines have been reformatted slightly. User and server numbers now occupy up to five characters; previously, they occupied exactly three.

Messages paired for more information

Message M204.0146: filename - ROLL BACK INFORMATION IS OBSOLETE is now followed by message M204.2565, which provides additional information that may be useful in determining the reason why the file could not be rolled back. The text of the new message is:

```

M204.2565: filename=yy.ddd hh:mm:ss.tt,
RESTART=yy.ddd hh:mm:ss.tt
  
```

See "0146" on page 5-2 and "2565" on page 5-30 for a full explanation of these messages.

Displaying the running users

Starting in Version 5.1, the user who issues a MONITOR command against the scheduler queues is described as running (RUNG), not just ready (REDY). See “MONITOR: An Online system” on page 1-100.

Returning the checkpoint status

Output from the CHKMSG command can include status information about an extended quiesce; for example:

```
CHECKPOINT COMPLETED ON 01.297 13:18:44.15
SYSTEM ENTERED EXTENDED QUIESCE AT: 10/24/01
13:18:44, SYSTEM EXITED
EXTENDED QUIESCE AT: 10/24/01 13:18:55, REASON = END
COMMAND ISSUED BY
USER NUMBER: 3
```

Message for the Operator

While ROLL FORWARD is processing, the following message is issued:

```
M204.1992: RECOVERY: PROCESSING ROLL FORWARD BLOCK#
blocknumber date-time stamp
```

The message gives the processing sequence number and date-time stamp. It is issued at the Operator’s console for each occurrence of an hour change in the date-time stamp of the CCARF update records.

You could receive a maximum of 24 such messages in a 24-hour period.

Additional wait types

The following wait types are introduced in Version 5.1:

Type	Explanation
30	User(s) waiting, with \$WAIT(<i>nn</i> , 'SWAP'), for a user-defined ECB to be posted by a \$POST function from a different user.
31	User(s) waiting, with \$WAIT(<i>nn</i> , 'NOSWAP'), for a user-defined ECB to be posted by a \$POST function from a different user.
43	ECF to load or delete a module.
44	External module to become free.
45	ECF subtask to become free.
46	External module to run.

Type	Explanation
47	User(s) waiting, with \$WAIT('CPQZ'), for the extended quiesce ECB to be posted by the successful completion of a Model 204, system-wide checkpoint.
48	User(s) waiting, with \$WAIT('QZSIG'), for the end of extended quiesce.
49	User(s) waiting, at end of extended quiesce, for count of \$WAIT('CPQZ') and \$WAIT('QZSIG') users to go to zero.
50	User is waiting for HSM to recall an archived dataset.

All operating systems

Assembling source

High Level Assembler 1.4 (HLASM) is required to correctly assemble Model 204 source.

If your site has assembler programs that reference the Model 204 macro library, you must reassemble with each new release of Model 204.

In Model 204 Version 5.1, the length of the statistic block has changed, necessitating reassembly of your programs that reference statistics blocks.

Installing Connect ★

If you purchase Connect ★, your contract determines the number of seats or threads you may define. In Version 5.1, the contract number is enforced. If you define more Connect ★ threads than you have contracted for, an error at Online start-up time is generated indicating your valid limit. If you wish to increase this limit, or if you think the message is incorrect, please contact CCA Sales or Customer Support. A new CPUID zap will be provided to adjust your level of service.

AUDIT204 and processing journals

Journals must be processed with AUDIT204 from the release that created the journal. Failure to do so causes the statistics to be improperly labeled.

Obsolete parameter and statistic

In Version 5.1, the LDKBMRW parameter becomes obsolete. The DKSRW statistic is no longer used.

Global table high water marks

As of Version 5.1, the GTBL usage high water marks, as reported in since-last statistics, include all data in GTBL, even the global object data pointers. Prior to Version 5.1, global object data pointers were not counted in the GTBL usage high water marks. From Version 4.2 to Version 5.1, the reported high water marks increased by 32, even if you do not use the Version 5.1 hashed GTBL feature and your actual GTBL usage does not increase.

IFAM application programs

IFAM1, IFAM2, and IFAM4 application programs must be relinked in Version 5.1.

No M204XSVC should be linked in for IFAM1. CCA always recommends to install and use an SVC for IFAM1 rather than linking in M204XSVC, which must be authorized.

Note: IFIF, linked into IFAM2 applications, was assembled with AMODE(31), RMODE(ANY) attributes.

For OS/390 and z/OS operating systems

Introducing Model 204 support for z/OS

Model 204 can run under IBM's operating system z/OS. All functionality in Model 204 Version 5.1 works on OS/390 V2.10 as well as z/OS.

Model 204 and 64-bit z/Architecture

Model 204 operates safely under z/OS in 64-bit mode using IOS Branch Entry or EXCP options. Version 5.1 has improvements that allow customers using IOS Branch Entry to avoid a performance penalty using real storage above the 2-gigabyte line.

Reinstalling CRAM

You must reinstall CRAM when you install Model 204 Version 5.1. You must use the CRAMINS job generated by the JCL for Model 204 Version 5.1. Earlier versions of the CRAMINS job are incompatible with this version of Model 204, because of a change in the AMODE setting.

Under OS/390 and z/OS, CRAMINS, the CRAM installation job, links IGCLM244 AMODE(31).

Installing CRAM and/or M204XSVC

You can install CRAM and/or M204XSVC with or without using SVCs. To make a decision for your site, please consult the *Model 204 OS/390 and z/OS Installation Guide* to review the benefits and limitations of each option.

Changes to INSPARMs in Version 5.1

The following INSPARMs task parameters have a different default setting in Version 5.1, or they have been removed from the INSPARMs task parameters, because they are automatically included in the ONLINE module.

INSTALL-DICT=I

Use this default to create the Dictionary/204 subsystem and allocate the SQL support subsystems, CCACATREPT and CCATSF. These are not required, but can provide report and generation aids when administering SQL accessed Model 204 databases.

For existing Dictionary/204 users, change the default I (Install) to R (Reinstall). If you do not wish to use subsystems or Dictionary/204, change this to N for No.

INSTALL-SQL=I

Includes the Remote SQL object code, RSQL, rather than the former default EVCP code in the Model 204 Online. RSQL is significantly larger than EVCP. This will also cause job SQLINST to be generated, which builds a new SQL system catalog, CCACAT, and restores procedure files for the SQL subsystem.

If you have already installed previous versions of these features, change this default to R for the reinstall job, SQLRINST. Otherwise, set this parameter to the previous default N for No. EVCP will be used in the link job.

HRZN=Y

You no longer need a decryption key to install and use Horizon. By default, one Horizon thread, IODEV=27, may be defined in your ONLINE module. If you purchase Horizon, the number of threads you can use is unlimited.

Horizon allows the SNA Communications Server/TCPIP necessary for Connect ★ and customer written distributed application programs.

PQO=Y

PQO no longer requires a decryption key and by default is part of your ONLINE module generation. If you have not yet purchased the Parallel Query Option, the IODEV=51 thread definition is limited to one. Existing customers will have no limit for definition of this thread type.

SQL-COBOL=Y

If you indicate that the RSQL object code should be used when generating your ONLINE module (INSTALL-SQL=I or R) and CRAM is installed, this new default allows Advantage/SQL use of the free SQL batch COBOL support. Set SQL-COBOL=N, if you do not want or will not use this feature.

SQL-CONNECT*=Y

Setting SQL-CONNECT*=Y allows for two free threads for each of the following:

- Remote SQL threads (IODEV=19)
- Remote User Language threads (IODEV=49)

The Y setting generates a note indicating Connect ★ Workstation software must be installed on Client PCs. Remote SQL is required in the Online. If no Connect ★ capability is desired, set this parameter to the value N.

TCPIP

TCPIP is no longer a selectable option and is automatically included in the Online generation with Horizon.

SNA Communications Server

SNA Communications Server is no longer a selectable option and is automatically included in the Online generation.

Limited support for EXCPVR

EXCPVR is not supported for z/OS in 64-bit mode. CCA recommends using IOS Branch Entry. See the *Model 204 OS/390 and z/OS Installation Guide* for details.

CICS Transaction Server support

CCA has updated the CICS interface to include support for CICS Transaction Server for OS/390 (Release 1.3) and CICS Transaction Server for z/OS (Release 2.1).

CICS Transaction Server beginning with Release 1.3 includes the ability to execute server Java applications from its own subsystem. Thus, CICS users can execute server Java code alongside existing CICS applications written in languages such as COBOL and PL/I. You can integrate Model 204 data and transactions with your CICS-based e-initiatives.

MQ/204 enhancements

MQ/204, the interface from Model 204 to IBM's MQSeries, has been enhanced in Model 204 Version 5.1. Additionally, the MQ/204 documentation has been significantly expanded. Users of MQ/204 are encouraged to refer to the latest edition of the *Model 204 MQ/204 Reference*.

Beginning in Version 5.1, you can specify the REPORT option on various MQ/204 statements as a number or as a %variable containing a numeric value. Previously, the REPORT option could be specified by only keywords.

For VM operating system

31-bit BATCH2 / IFAM2 processing

Handling 31-bit IFAM2 applications

In Model 204 Version 5.1, BATCH2 and IFAM2 are now 31-bit modules under VM. To run an IFAM2 program in 31-bit mode, you must change your compile and link procedures as appropriate for the application. Consult the documentation for your compiler. For example:

To create	Take this action...
VS COBOL II program	Compile the program with the REENTRANT and RESIDENT options.
LE COBOL program	Specify the RMODE(ANY) option on the compile step.
31-bit COBOL	Link with the AMODE(31) and RMODE(ANY) options.

Handling 24-bit IFAM2 applications

No changes are required to run 24-bit IFAM2 application programs.

Dynamic allocation in CMS

Dynamic allocation in CMS and allocation using M204UTIL now supports OS formatted minidisks, including full volume minidisks with VTOC crossing cylinder boundary and multi cylinder VTOC. There is still a restriction for creating a VTOC using M204UTIL to be in cylinder boundaries.

Version 5.1 does not allow allocation of new datasets on volumes with indexed VTOC. Error message "NEW ALLOCATION NOT ALLOWED ON INDEXED VTOC" will be issued and return code 24 set for dynamic allocation on indexed VTOC volumes.

Changes to M204GEN command in Version 5.1

The following options were added to the M204GEN command to support Freeway/204. For a description of this feature see "Freeway/204" on page 1-24. Because they are included in the Online generation, decryption keys are no longer required for Horizon, PQO, and Connect ★, or SNA Communications Server and TCP/IP.

Because Horizon, Parallel Query Option, and Remote SQL are installed by default, note the additional options to the M204GEN command. If you do not want to try these features at this time, use the options to negate the defaults:

- HRZN / NOHRZN

If you accept the default HRZN, one Horizon thread, IODEV=27 is defined to your ONLINE module. To exclude Horizon from your ONLINE module, specify NOHRZN.

- PQO / NOPQO

If you accept the default PQO, one IODEV=51 thread is defined to your ONLINE module. To exclude the Paralle Query Option from your ONLINE module, specify NOPQO.

- RSQL / EVCP

If you accept the default RSQL, the Remote SQL object code is included in your ONLINE module generation. This allows for two free threads for each of the following:

- Remote SQL threads (IODEV=19)
- Remote User Language threads (IODEV=49)

To use Connect ★, you must install the Connect ★ workstation software on client PCs. If no Connect ★ capability is desired, specify EVCP.

For VSE operating system

CICS Transaction Server 1.1 support

The CICS interface for VSE includes support for CICS Transaction Server for VSE/ESA Release 1.1.

Identifying AUDIT204 utility users

The printed report generated by the REPORT USERID function of the AUDIT204 utility now includes an extra header line for each group of USERID records:

USERID: *useri d_bei ng_report ed_on*

This is an aid in determining which USERID is responsible for Model 204 transactions.

Horizon for TCP/IP and Horizon/LE for TCP/IP limitations

Horizon for TCP/IP and Horizon/LE for TCP/IP, as described in “Horizon for TCP/IP and Horizon/LE for TCP/IP” on page 1-72, are not available for VSE. The VSE operating system continues to support the Horizon functionality of earlier releases.

Limitations with sequential data files

Currently, the current journal location is not included when running under VSE. This is due to IBM restrictions with NOTE/POINT with sequential data files under VSE.

3

Model 204 External Call Facility

In this chapter

- Overview
- Model 204 commands relating to ECF
- ECF User Language statements
- Tracking ECF
- Return codes
- Introducing ECF User 0 parameters
- Subtask and load module management
- Restrictions and cautions
- ECF examples

Overview

The External Call Facility (ECF) is a method for programs written in Model 204 User Language to invoke external, non-Model 204 modules, such as non-IFAM COBOL modules. Data can be passed between Model 204 and external modules. External modules can open non-Model 204 datasets, read or write to them, and close them. In addition, multiple modules can be called multiple times in a single run.

ECF is available only under OS/390 and z/OS.

Loading an external module

Each external module that is called using ECF is dynamically loaded once into memory from a nominated load library or from //STEPLIB. The load is done by an authorized user executing the EXTERNAL LOAD statement; for example, by User 0 during startup. The load is issued from a dedicated ECF subtask that is used exclusively for module loads and deletes.

After an external module is loaded, a *call-name* of up to 48 characters is associated with it. An association is set up by an authorized user executing the EXTERNAL NAME statement; typically by User 0 during startup. More than one call name can be associated with the same load module name.

Callers of the external module always refer to the call (logical) name rather than the load module (physical) name. Thus, if a load module name or location changes, only the setup statements need to be changed; no application code changes are required.

Calling an external module

A previously loaded external module is called by a user executing the EXTERNAL CALL statement. This invokes a separate ECF subtask to run the external module, and causes the user to enter a swappable, bumpable wait state. When the external module completes, control returns to the user. There are no restrictions on the length of time the external module can run.

Data is passed between Model 204 and an external module via Model 204 image. A copy of the image is made when the module is called, so that the caller need not remain in the server while the external module runs. The external module can modify the copy of the image. The modified image is returned to the caller, unless the EXTERNAL MODULE statement specified PARMTYPE=INPUT.

Avoiding data corruption and incorrect results

External modules must not access any dataset that Model 204 or any other external module has already opened. *Failure to observe this could result in data corruption or abends in either the external module or Model 204.*

External modules invoked using ECF run under separate OS/390 and z/OS subtasks in the same address space as the Model 204 Online that invoked the module. This means that external modules run in parallel with Model 204 (and each other), even if you do not use MP/204 (multiprocessor).

Most errors that can occur in the external module are isolated from Model 204. However, it is still possible for programming errors in external modules to corrupt or overwrite storage belonging to Model 204. *A programming error in your external module could result in data corruption, abends, or incorrect results in the external module or in Model 204.*

Stopping an external load module

Authorized users can stop an external module by executing:

- EXTERNAL STOP statement, usage of a load module can be disabled by an authorized user. It can be enabled again by the EXTERNAL START statement.
- EXTERNAL DELETE statement, a load module can be removed from storage by an authorized user. An EXTERNAL DELETE statement must not be used for Language Environment (LE) modules, because OS/390 and z/OS does not support the use of DELETE of Language Environment main programs and an abend can result.

ECF statistics and messages

See Table 1-3 on page 1-107 for the System-final, user-logout, user-since-last, SMF-logout, and SMF-since-last statistics that relate to ECF.

See Chapter 5 for the messages that relate to ECF.

Model 204 commands relating to ECF

The BUMP command was modified to bump all users running a specified external module.

BUMP MODULE: Logging out a user from an external module

Function Logs out a particular user or a set of users

Privileges System administrator or operator

Syntax `BUMP { userid | usernumber | ALL | FILE filename
| SUBSYSTEM subsysname | QUEUEMANAGER qmname
| MODULE module-name }`

Where The arguments are:

Argument	Specifies...
userid	Character string that identifies a particular Model 204 user. Users who have logged in under this userid are logged out.
usernumber	A particular user by a numeric value.
ALL	All current users are logged out.
FILE filename	Name of a Model 204 file.
SUBSYSTEM subsysname	Name of a Model 204 application subsystem.
QUEUEMANAGER qmname	Name of a queue manager defined with a DEFINE QUEUEMANAGER command to support MQSeries.
<i>module-name</i>	The particular external module whose users are to be bumped.

Note: Only users who are running the specified external module at the time the BUMP MODULE command is issued are bumped; subsequent users of the module are not affected. To prevent subsequent use of the module, use the EXTERNAL STOP statement.

ECF User Language statements

You must check \$STATUS and \$STATUSD return codes after each ECF statement.

EXTERNAL CALL statement

Function Calls an external module that was previously loaded using an EXTERNAL LOAD statement. The call name used was previously specified in an EXTERNAL NAME statement.

Privileges Any user

Syntax EXTERNAL CALL *call-name* WITH *image-1*
[, *image-2* , . . . *image-7*]

Where The arguments are:

Argument	Specifies...
call-name	Logical name, either a literal or %variable, of the external module to invoke.
image-1,...image-7	Name(s) of a previously defined image(s) to pass to the external module.

Usage You may specify from one to seven images separated by commas.

Note: A return code of zero does *not* mean that the external module performed as it was designed; it simply means that the module was successfully invoked and completed without an ABEND.

EXTERNAL DELETE statement

Function Removes a previously loaded external module from storage.

Privileges System manager or User 0

Syntax EXTERNAL DELETE *module-name*

Where The argument, a literal, or a %variable, is:

Argument	Specifies...
module-name	Module name of a previously loaded external module.

Usage Do not use the EXTERNAL DELETE statement for Language Environment modules.

EXTERNAL LOAD statement

Function Loads an external module into storage. The module, previously defined by an EXTERNAL MODULE statement, is loaded and available to all users.

Privileges System manager or User 0

Syntax EXTERNAL LOAD *module-name*

Where The argument, a literal, or a %variable, is:

Argument	Specifies...
module-name	Name of the external module, a PDS member name.

EXTERNAL MODULE statement

Function Defines an external module for later loading.

Privileges System manager or User 0

Syntax EXTERNAL MODULE *module-name* [DDNAME=*ddname*]
[PARMTYPE=INPUT | OUTPUT]
[PARMSE=*value*]
[PARMMODE=[24 | 31]]
[REENTRANT | AFFINITY]

Where The arguments, either literals or %variables, are:

Argument	Specifies...
<i>module-name</i>	Name of the external module, a PDS member name. Required.
<i>ddname</i>	DDNAME of the PDS where the load module is located. If omitted, the standard search order or //STEPLIB is used. Optional.
PARMTYPE	Parameter (image) type passed to the external module on an EXTERNAL CALL statement. Optional. <ul style="list-style-type: none">INPUT means that any changes made by the external module to the parameter are discarded.OUTPUT, the default, means that any changes are retained.

Argument	Specifies...
PARMSIZE	<p>Required size of the parameter (image) passed to the external module on all EXTERNAL CALL statements.</p> <p>If the actual parameter size does not match the required size, the EXTERNAL CALL statement fails.</p> <p>If PARMSIZE is not specified, the parameter size is not checked.</p>
PARMMODE	<p>Where storage to hold the copy of the parameter area is allocated. Optional.</p> <ul style="list-style-type: none"> • 24 means allocate in 24-bit, below-the-line storage. • 31, the default, means allocate in 31-bit, above-the-line storage. <p>Note: Do not specify PARMMODE=24 unless the module has a specific requirement for this setting.</p>
REENTRANT	<p>Module that can be used by more than one user at a time. This optional argument takes effect only if the module was link-edited with the REENTRANT attribute. Do not specify REENTRANT unless the module has a specific requirement for this setting.</p>
AFFINITY	<p>Module always runs on the same subtask. Conversely, that subtask runs only that module. Do not specify AFFINITY unless the module has a specific requirement for this setting.</p>

Usage You can use an EXTERNAL MODULE statement to define a module using either:

- New name and attributes.
- Existing name. Attributes associated with an existing name are replaced by new attributes.

Subtask affinity, which is specified by the AFFINITY keyword, is required in some situations. For example, when an external module opens a dataset on one call, but does not close it till a subsequent call. Subtask affinity is required because OS/390 and z/OS require that you open and close a dataset from the same subtask.

The AFFINITY keyword is incompatible with the REENTRANT keyword. A compilation error is generated, if both are specified for the same module. You specify one or the other or neither, but not both.

EXTERNAL NAME statement

Function Use the EXTERNAL NAME statement to:

- Associate a logical call name with the name of a module that was previously defined by issuing an EXTERNAL MODULE statement

- Remove a previously set up association. A module can have multiple call names.

Privileges System manager or User 0

Syntax EXTERNAL NAME *call - name* FOR *module - name*
EXTERNAL NAME *call - name* REMOVE

Where The arguments, either literals or %variables, are:

Argument	Specifies...
<i>call-name</i>	Logical name to associate with an external module. Up to 48 characters.
<i>module-name</i>	Name of a previously loaded external module.

EXTERNAL START statement

Function Enables further calls to an ECF module.

Privileges System manager or User 0

Syntax EXTERNAL START *module - name*

Where The argument, either a literal or %variable, is:

Argument	Specifies...
<i>module-name</i>	Name of a previously loaded external module.

Usage Initially, a module is in the START state. You need not issue an EXTERNAL START statement unless you want to reverse a prior EXTERNAL STOP statement.

EXTERNAL STOP statement

Function Stops further calls to an ECF module. Currently executing calls either complete or, if the FORCE option is used, abend.

Privileges System manager or User 0

Syntax EXTERNAL STOP *module - name* [FORCE]

Where The arguments, which can be either literals or %variables, are:

Argument	Specifies...
module-name	Name of a previously loaded external module.
FORCE	Users currently executing the specified module are bumped.

Usage The following table describes how an EXTERNAL STOP command is evaluated.

When EXTERNAL STOP command is issued	Then
If no users are executing the module	Module is immediately marked <i>stopped</i> .
If one or more users are executing the module	Module is marked <i>draining</i> until the last user finishes executing the module, then it is marked <i>stopped</i> .
If FORCE was specified	Current users of the module are bumped.
If FORCE was not specified	Current users of the module are allowed to complete.

In all cases, subsequent attempts by any user to call the module with an EXTERNAL CALL statement result in a \$STATUS of 8.

Note: CCA does *not recommend* using the FORCE option, as Model 204 cannot ensure that the external module is terminated cleanly.

Tracking ECF

Wait types for ECF

See “Additional wait types” on page 2-13 for a listing of the wait types for the External Call Facility.

New ECF statistics in Version 5.1

In the *Model 204 System Manager’s Guide*, tables list the statistics in numerical order as they appear in the journal record layouts for System-final, user-logout, user-since-last, SMF-logout, and/or SMF-since-last statistics. The new statistics introduced with ECF are listed in Table 1-3 on page 1-107.

Return codes

All EXTERNAL statements set \$STATUS and \$STATUSD.

Table 3-1. Return codes for ECF

\$STATUS	\$STATUSD	Meaning
0	0	ECF function completed without error
1	0	ECF inactive
2	0	Not authorized
3	1	Invalid module name
	2	Invalid call name
	3	Invalid DDNAME
	4	Image inactive
	5	Invalid PARMSIZE
4	0	Call name not defined
6	0	Module not defined
7	0	Module not loaded
8	0	Module unavailable (draining or stopped)
10	0	Module not deleted
		System busy; timed out
	1	Module unavailable
20	2	No subtask available
		Load or delete failed
30	1	DDNAME not present
	2	DDNAME open failed
	3	DDNAME close failed
	5	Load or delete failed; see "\$ECFSTAT function" on page 3-12
	6	Internal ECF ABEND; see "\$ECFSTAT function" on page 3-12
	7	Internal ECF ABEND; Load or delete subtask terminated by the operating system
40		Module failed
	1	Module gave nonzero return code; see "\$ECFSTAT function" on page 3-12

Table 3-1. Return codes for ECF (continued)

\$STATUS	\$STATUSD	Meaning
	2	Module ABEND; see "\$ECFSTAT function" on page 3-12
	3	Insufficient memory available to allocate a buffer for the parameter area
	4	Actual parameter size is not equal to PARMSIZE
	5	When the parameter area, as updated by the external module, was being copied back to the original image(s), ECF detected that the size of one of the images had been changed. This status can occur only if an image contains a variable array whose size is changed by the external module.
	6	ECF subtask terminated by the operating system
50		ECF external table full
	1	ECF ECMODS table full; increase ECMODS User 0 parameter
	2	ECF ECNAMES table full; increase ECNAMES User 0 parameter

\$ECFSTAT function

Function Returns the detailed completion code from the previous EXTERNAL statement.

Syntax The \$ECFSTAT function returns:

```
' RC=cccc, COMPLETION CODE=t ww, REASON CODE=rrrrrrrr'
```

Where Arguments can be specified as

Argument	Represents...
<i>cccc</i>	Completion code in decimal format
<i>t</i>	ABEND type, either S for SYSTEM or U for USER
<i>ww</i>	One of the following ABEND values: <ul style="list-style-type: none"> • SYSTEM values are three hexadecimal digits followed by a blank • USER values are four decimal digits
<i>rrrrrrrr</i>	Reason code, eight hexadecimal digits

Usage If ECF is not active or the user has not executed any EXTERNAL CALL, EXTERNAL DELETE, or EXTERNAL LOAD statements, a NULL string is returned.

Introducing ECF User 0 parameters

The following User 0 parameters are used by ECF.

A system manager can reset only the ECWAIT parameter; the other ECF parameters cannot be reset.

Parameter	Purpose	Documented on...
ECISUBS	Subtasks for running external modules initially attached	page 1-80
ECMODS	External modules to load	page 1-80
ECMSUBS	Number of ECF subtasks for running modules	page 1-80
ECNAMES	Number of external call name	page 1-81
ECPRIV	ECF privileges	page 1-81
ECPSIZE	Size of initial parameter area allocated by ECF	page 1-81
ECWAIT	ECF wait time	page 1-82

Subtask and load module management

Some OS/390 and z/OS overhead accrues in loading an external module into storage and attaching an OS/390 and z/OS subtask under which it runs. ECF avoids incurring this cost for every EXTERNAL CALL statement by managing the load modules and subtasks as described in the following sections.

Subtasks assignment

At system startup, the following subtasks are started:

- One subtask for loading and deleting modules
- One or more subtasks, specified by the ECISUBS parameter, for executing external modules

When a user issues an EXTERNAL CALL statement, an unused ECF subtask is selected on which to run the external module, up to the limit of ECMSUBS. Users unable to get a subtask enter a wait of up to ECWAIT milliseconds for a subtask to become available. When an in-use ECF subtask becomes available, because the module that was running under it ends, it is assigned to a user waiting for a subtask. If a user's wait time expires before a subtask is available, a no-subtask-available failure is returned.

Subtask affinity

Normally, when an external module is called via the EXTERNAL CALL statement, ECF selects any free subtask on which to execute the module. Although this is appropriate for most external modules, some modules might need to always be executed on the same subtask. This is known as **subtask affinity**. Subtask affinity for a module is specified by the AFFINITY option on the EXTERNAL MODULE statement. The AFFINITY option is incompatible with the REENTRANT option.

- If the AFFINITY option is not specified, the module does not have subtask affinity. This is the default.
- If the AFFINITY option is specified, the module has subtask affinity.

When the first EXTERNAL CALL of the module is executed, a dedicated subtask for the module is attached. A two-way association between the module and the subtask is established. That module will run on only that subtask, not on any other subtask. That subtask is used to run only that module, not any other module.

If a module has subtask affinity, and the dedicated subtask for that module is abnormally terminated for any reason, the previous dedicated subtask is detached, and a new dedicated subtask for the module is attached. A subtask could be abnormally terminated if the module it was running has an abend, or the user running the module is bumped. Therefore, specifying AFFINITY does

not guarantee that the same subtask is always used for that module for the duration of a Model 204 Online.

If you use modules with subtask affinity, you should ensure that ECISUBS and ECMSUBS are appropriately set. In particular, ECISUBS and ECMSUBS must be greater than or equal to the number of modules with subtask affinity.

- The ECISUBS parameter specifies the number of subtasks attached during system initialization. These subtasks are never used for modules with subtask affinity; the first EXTERNAL CALL of a module with subtask affinity always results in a new subtask being attached (subject to ECMSUBS).
- The ECMSUBS parameter specifies the maximum number of ECF subtasks used to run modules. The number of subtasks used for modules with subtask affinity *plus* the number of subtasks used for modules without subtask affinity will be a maximum of ECMSUBS.

Load modules

An external module is usually loaded into storage just once. The exception is when an EXTERNAL LOAD statement, possibly preceded by an EXTERNAL DELETE statement, is used to reload a previously-loaded module.

Normally, one Model 204 user at a time is allowed to issue an EXTERNAL CALL statement for the module. Other users who attempt to call the module while it is in use, enter a wait-state of up to ECWAIT milliseconds for the module to become available. When a module is freed, because the current execution of it ends, it is assigned to one of the users waiting for it. If a user's wait time expires before the module is assigned, a module-in-use failure is returned.

If a load module was link-edited with the REENTRANT (also written, RENT) attribute, and if the EXTERNAL MODULE statement that defined the module characteristics specified the REENTRANT option, then multiple Model 204 users are allowed to simultaneously issue an EXTERNAL CALL statement for the module—subject to subtask availability.

Fulfillment order

An EXTERNAL CALL statement can invoke an external module only if both an ECF subtask is free, and for a serially reusable module, if the module is not in use by another user. The check that the module is not in use by another user is done before the allocation of a subtask so that the most restrictive condition is checked first.

Restrictions and cautions

The following restrictions and cautions apply to the use of ECF:

- ECF loads an external module only once, when the EXTERNAL LOAD statement is issued. Thereafter, every user who issues an EXTERNAL CALL statement for that module uses the same copy of the module. Therefore, you must write your module to initialize itself properly on every call.
- ECF passes the copy of the parameter area to the external module using standard OS/390 and z/OS linkage conventions. Your external module must support these conventions to receive the parameters.
- ECF does not provide any special initialization of the environment when you issue an EXTERNAL CALL statement; it merely branches to the in-memory copy of the module. Therefore, your module must perform any required initialization and termination of its run-time environment. In particular, this means that an external module, written in any language, should be written as a main routine and not a subroutine.
- Only one copy of a load module can be in memory at a time. This is an OS/390 and z/OS restriction.
- Externally called modules must be AMODE(31) and either RMODE(ANY) or RMODE(31).
- Externally called modules must not attempt to retain any context information from one call to another. You must write the modules so that each call executes independently of any other.
- For efficiency purposes, ECF does not use OS/390 and z/OS to maintain the usage counts or status information for load modules. Control is passed by direct branch rather than use of the OS/390 and z/OS ATTACH, LINK, or XCTL macros. Therefore, one external module must not attempt to load or attach another external module, or attempt to reference code or data in another external module.
- The definition of the parameter area in Model 204—the image definition—must agree with the definition of the parameter area in the external module (in a COBOL program, the LINKAGE SECTION). If they do not agree, it is possible for the module to modify the wrong storage.

To prevent this, Model 204 checks that the external module put its results in *only* the assigned area—not somewhere else. Model 204 checks the area immediately past the end of the assigned area. If the unassigned area was used, Model 204 displays the following message and restarts the user.

```
M204. 2563: MODULE=name RETURNED MORE THAN length
BYTES
```

ECF examples

This section illustrates, in various languages, how to write, compile, link and invoke an external module that adds two numbers together and returns the sum. For clarity, the sample code omits error handling, other standard elements, and some JCL elements.

COBOL sample Number 1

This example uses the Language Environment enabled compiler, IBM COBOL FOR OS/390 and z/OS AND VM.

COBOL program

```
//COBSAMP EXEC PGM=IGYCRCTL, PARM=(NOSEQ, RENT)
//SYSLIN DD DSN=YOUR. OBJLIB(COBSAMP), DISP=SHR
//SYSIN DD *
IDENTIFICATION DIVISION.
PROGRAM-ID. COBSAMP.
DATA DIVISION.
LINKAGE SECTION.
01 M204-PARMS.
   03 NUMBER-ONE PIC S9(7) COMP-3.
   03 NUMBER-TWO PIC S9(9) BINARY.
   03 NUMBER-SUM PIC S9(9) BINARY.
PROCEDURE DIVISION USING M204-PARMS.
   COMPUTE NUMBER-SUM EQUAL NUMBER-ONE + NUMBER-TWO
   MOVE ZERO TO RETURN-CODE
   GOBACK.
```

Language Environment Options

```
//ASMUOPT EXEC PGM=ASMA90, PARM='NOXREF'
//SYSLIB DD DSN=SYS1.MACLIB, DISP=SHR
// DD DSN=CEE.SCEEMAC, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=YOUR. OBJLIB(CEEUOPT), DISP=SHR
//SYSIN DD *
CEEUOPT CSECT
CEEUOPT AMODE ANY
CEEUOPT RMODE ANY
PRINT ON, NOGEN
CEEXOPT
   ABTERMENC=(ABEND),
   RTEREUS=(ON)
END
```

Linkedit

```
//LINK EXEC PGM=IEWL, PARM='LIST, MAP'
```



```
//SYSPRI NT DD SYSOUT=*
//SYSLI B DD DSN=CEE. SCEELKED, DI SP=SHR
//OBJLI B DD DSN=YOUR. OBJLI B, DI SP=SHR
//SYSLMOD DD DSN=YOUR. LOADLI B, DI SP=SHR
//SYSLI N DD *
I NCLUDE OBJLI B(CEEUOPT)
I NCLUDE OBJLI B(COBSAMP)
NAME COBSAMP(R)
```

Calling from User Language

```
//ONLI NE EXEC PGM=ONLI NE
//STEPLI B DD DSN=M2O4. LOADLI B, DI SP=SHR
// DD DSN=YOUR. LOADLI B, DI SP=SHR
// DD DSN=CEE. SCEERUN, DI SP=SHR
```

```
BEGI N
I MAGE P ARMS
    NUMBER. ONE. PACKED I S PACKED LEN 4
    NUMBER. TWO. BI NARY I S BI NARY LEN 4
    NUMBER. SUM. BI NARY I S BI NARY LEN 4
END I MAGE
```

```
EXTERNAL MODULE COBSAMP
EXTERNAL LOAD COBSAMP
EXTERNAL NAME MOD_COBSAMP FOR COBSAMP
```

```
PREPARE I MAGE P ARMS
%PARMS: NUMBER. ONE. PACKED = 1
%PARMS: NUMBER. TWO. BI NARY = 2
EXTERNAL CALL MOD_COBSAMP WI TH P ARMS
PRI NT %PARMS: NUMBER. ONE. PACKED WI TH ' + ' WI TH -
      %PARMS: NUMBER. TWO. BI NARY WI TH ' = ' WI TH -
      %PARMS: NUMBER. SUM. BI NARY
END
```

Comments.

COBOL	Corresponds to User Language
PI C S9(7) COMP- 3	PACKED LEN 4
PI C S9(9) BI NARY	BI NARY LEN 4

- The Language Environment option ABTERMENC=(ABEND) *must* be specified.
- Specify the Language Environment option RTEREUS=(ON) and code GOBACK instead of STOP RUN , as illustrated, to make the runtime environment reusable and to improve performance.
- Setting Language Environment options is described in IBM Manual SC28-1939 *Language Environment for OS/390 and z/OS & VM Programing*

Guide. In the code in the previous section, “Calling from User Language”, Language Environment options are set by linking CEEUOPT with the COBOL module. Other methods are also available and you should check with your site’s Language Environment administrator to determine the appropriate method to use.

COBOL sample Number 2

This example illustrates the use of multiple images using the Language Environment enabled compiler, IBM COBOL FOR OS/390 and z/OS AND VM.

Note: Ordinarily, multiple images are only used if the parameters to be passed cannot fit into a single image. Images are limited in size to 32767 bytes.

COBOL program

```
I DENT I F I C A T I O N   D I V I S I O N .
PROGRAM- I D.   COBSAM2.
DATA   D I V I S I O N .
LINKAGE SECTI O N .
01  M204- PARMS1.
    03  NUMBER- ONE                P I C  S9( 7)  COMP- 3.
01  M204- PARMS2.
    03  NUMBER- TWO                P I C  S9( 9)  B I N A R Y .
01  M204- PARMS3.
    03  NUMBER- SUM                P I C  S9( 9)  B I N A R Y .
PROCEDURE D I V I S I O N   U S I N G  M204- PARMS1,  M204- PARMS2,  M204- PARMS3.
    COMPUTE NUMBER- SUM EQUAL NUMBER- ONE + NUMBER- TWO
    MOVE ZERO TO RETURN- CODE
    GOBACK.
```

Calling from User Language

```
BEG I N
I M A G E  P A R M S 1
    N U M B E R . O N E . P A C K E D   I S   P A C K E D   L E N   4
E N D   I M A G E
I M A G E  P A R M S 2
    N U M B E R . T W O . B I N A R Y   I S   B I N A R Y   L E N   4
E N D   I M A G E
I M A G E  P A R M S 3
    N U M B E R . S U M . B I N A R Y   I S   B I N A R Y   L E N   4
E N D   I M A G E

E X T E R N A L   M O D U L E   C O B S A M 2
E X T E R N A L   L O A D   C O B S A M 2
E X T E R N A L   N A M E   M O D _ C O B S A M 2   F O R   C O B S A M 2

P R E P A R E   I M A G E   P A R M S 1
P R E P A R E   I M A G E   P A R M S 2
P R E P A R E   I M A G E   P A R M S 3
```

```

%PARMS1: NUMBER. ONE. PACKED = 1
%PARMS2: NUMBER. TWO. BINARY = 2
EXTERNAL CALL MOD_COBSAM2 WITH PARMS1, PARMS2, PARMS3
PRINT %PARMS1: NUMBER. ONE. PACKED WITH ' + ' WITH -
      %PARMS2: NUMBER. TWO. BINARY WITH ' = ' WITH -
      %PARMS3: NUMBER. SUM. BINARY
END

```

SAS/C sample

This example uses the SAS/C compiler, not the IBM C compiler.

SAS/C program

```

//C          EXEC PGM=LC370B
//STEPLIB   DD   DSN=SASC. LOAD, DISP=SHR
//SYSLIB    DD   DSN=SASC. MACLIB, DISP=SHR
//SYSLIN    DD   DSN=YOUR. OBJLIB(SASCSAMP) , DISP=SHR
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
typedef struct PARMAREA {
    int A;
    int B;
    int C;
} parmarea;

int main(int argc, char **argv) {
    parmarea *pptr;
    if (argc != 2) {
        /* the parameter list was not in OS format */
        return 1000;
    }
    pptr = (parmarea *) argv[1];
    pptr->C = pptr->A + pptr->B;
    return 0;
}

```

Linkedit

```

//LC          EXEC PGM=IEWL, PARM=' AMODE(31) , RMODE(ANY) '
//SYSLIB     DD   DSN=SASC. STDLIB, DISP=SHR
//           DD   DSN=SASC. BASELIB, DISP=SHR
//OBJLIB     DD   DSN=YOUR. OBJLIB, DISP=SHR
//SYSLMOD    DD   DSN=YOUR. LOADLIB, DISP=SHR
//SYSPRINT   DD   SYSOUT=*
//SYSLIN     DD   *
    INCLUDE   OBJLIB(SASCSAMP)
    ENTRY     $MAINC
    NAME      SASCSAMP(R)
//*

```

Calling from User Language

```
//ONLINE EXEC PGM=ONLINE
//STEPLIB DD DSN=M204. LOADLIB, DISP=SHR
// DD DSN=YOUR. LOADLIB, DISP=SHR
```

```
BEGIN
IMAGE PARMS
    NUMBER. ONE. BINARY IS BINARY LEN 4
    NUMBER. TWO. BINARY IS BINARY LEN 4

    NUMBER. SUM. BINARY IS BINARY LEN 4
END IMAGE
```

```
EXTERNAL MODULE SASCSAMP
EXTERNAL LOAD SASCSAMP
EXTERNAL NAME MOD_SASCSAMP FOR SASCSAMP
```

```
PREPARE IMAGE PARMS
%PARMS: NUMBER. ONE. BINARY = 1
%PARMS: NUMBER. TWO. BINARY = 2
EXTERNAL CALL MOD_SASCSAMP WITH PARMS
PRINT %PARMS: NUMBER. ONE. BINARY WITH ' + ' WITH -
      %PARMS: NUMBER. TWO. BINARY WITH ' = ' WITH -
      %PARMS: NUMBER. SUM. BINARY
END
```

Comments.

SAS/C	Corresponds to User Language
int	BINARY LEN 4

- The SAS/C program must be coded as a main program, not a subroutine.
- To enable SAS/C to accept parameters in standard OS format, the entry point must be defined as \$MAINC and the parameters handled as illustrated. This technique is described in the *SAS/C Compiler and Library User's Guide*.

Assembler sample

This example uses the non-Language Environment assembler.

Assembler program

```
//ASMSAMP EXEC PGM=ASMA90
//SYSLIB DD DSN=SYS1. MACLIB, DISP=SHR
//SYSLIN DD DSN=YOUR. OBJLIB(ASMSAMP), DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ASMSAMP CSECT
```

```

ASMSAMP  AMODE 31
ASMSAMP  RMODE ANY
          USI NG *, 15
          L    1, 0(1)      GET ADDRESS OF I MAGE COPY
          XC   WORKAREA, WORKAREA      CLEAR WORK AREA
          MVC  WORKAREA+4(4), 0(1)     GET FI RST NUMBER  (PACKED)
          CVB  0, WORKAREA      GET FI RST NUMBER  (BI NARY)
          A    0, 4(1)         ADD SECON D NUMBER (BI NARY)
          ST   0, 8(1)         STORE SUM          (BI NARY)
          XR   15, 15          ZERO RETURN CODE
          BR   14              RETURN
          DS   OD
WORKAREA DS   PL8              WORKI NG STORAGE
          END   ASMSAMP

```

Linkedit

```

//LASM    EXEC PGM=I EWL
//OBJLI B DD   DSN=YOUR. OBJLI B, DI SP=SHR
//SYSLMOD DD   DSN=YOUR. LOADLI B, DI SP=SHR
//SYSPRI NT DD  SYSOUT=*
//SYSLI N  DD   *
INCLUDE  OBJLI B(ASMSAMP)
NAME ASMSAMP(R)

```

Calling from User Language

```

//ONLI NE EXEC PGM=ONLI NE
//STEPLI B DD   DSN=M204. LOADLI B, DI SP=SHR
//          DD   DSN=YOUR. LOADLI B, DI SP=SHR

```

```

BEGI N
I MAGE PARMS
    NUMBER. ONE. PACKED I S PACKED LEN 4
    NUMBER. TWO. BI NARY I S BI NARY LEN 4
    NUMBER. SUM. BI NARY I S BI NARY LEN 4
END I MAGE

```

```

EXTERNAL MODULE ASMSAMP
EXTERNAL LOAD ASMSAMP
EXTERNAL NAME MOD_ASMSAMP FOR ASMSAMP

```

```

PREPARE I MAGE PARMS
%PARMS: NUMBER. ONE. PACKED = 1
%PARMS: NUMBER. TWO. BI NARY = 2
EXTERNAL CALL MOD_ASMSAMP WI TH PARMS
CALL CHECK. STATUS
PRI NT %PARMS: NUMBER. ONE. PACKED WI TH ' + ' WI TH -
       %PARMS: NUMBER. TWO. BI NARY WI TH ' = ' WI TH -
       %PARMS: NUMBER. SUM. BI NARY
END

```

Comments

Assembler	Corresponds to User Language
DS PL4	PACKED LEN 4
DS F	BI NARY LEN 4

4

Model 204 V5R1 Revisions to Existing Documents

In this chapter

- Revisions to Command Reference Manual
- Revisions to Horizon: Intersystem Processing Guide
- Revisions to System Manager's Guide
- Revisions to the User Language Manual

Revisions to *Command Reference Manual*

SECTRLOG: Security trusted login environments

In Chapter 4, the following text was added to the SECTRLOG parameter description.

TPROCESS applications cannot use the SECTRLOG parameter. Doing so results in the following message:

```
M204. 1786:  PROCESS- TO- PROCESS NOT SUPPORTED ON THIS  
THREAD
```

Updating options for the MSGCTRL command

In Chapter 9 replace the following Action... descriptions for the options TERM and NOTERM in Table 9-12 as follows.

Option	Action
TERM	Causes the specified message to be displayed on the user's terminal or for single user jobs in CCAPRINT.
NOTERM	Does not display the specified message on the user's terminal or for single user jobs in CCAPRINT. Note: Model 204 passes return codes back in login and logout messages; if you set the NOTERM option for login and logout messages, you cannot get the return code.

Revisions to *Horizon: Intersystem Processing Guide*

Version 5.1 documentation was consolidated so that the syntax and discussion of the following commands are described in only the *Command Reference Manual*.

- DEFINE LINK
- DEFINE PROCESS
- DEFINE PROCESSGROUP
- MONITOR LINK
- MONITOR PROCESS
- MONITOR PROCESSGROUP

This change in location affected the *Model 204 Horizon: Intersystem Processing Guide* and the *Model 204 SQL Connectivity Guide*. These manuals still explain the design and considerations for network conversations, but they no longer have DEFINE and MONITOR command syntax and details for the arguments. Corrections and updates to the DEFINE and MONITOR command were made at the same time.

Changes documented for the DEFINE LINK command

Prior to Version 4.2, when you defined a link, the PROTOCOL argument accepted LU62C for a CNOS connection and LU62 for a non-CNOS connection. Beginning in Version 4.2, both types of connection were established by the LU62 value. (PROTOCOL=LU62)

When you examine the DEFINE LINK command, you cannot tell which type of connection was made. However, when you issue a MONITOR LINK command you can tell that the CNOS link has a CNOS session, denoted by an X in the FLGS column, as shown in the following example.

MONITOR LINK LULK1

LOCAL I . D.	MAXSES	BNDSES	CONVS	FLGS	TRAN/PROTO
DVCNMB8	50	2	0	A	VTAM/LU62

USER	PROCESS	SENDS	RECVS	FLGS	PROCESSGRP	REMOTE I . D
		0	0	B		CI CSRE5
		0	0	BX		CI CSRE5

Revisions to *System Manager's Guide*

In Chapter 3: Defining the User Environment (CCAIN)

Implementing CRAM XDM usage for OS/390 and z/OS operating systems

To implement CRAM-XDM, take the following steps:

1. Install Model 204 following the instructions in the *Model 204 OS/390 and z/OS Installation Guide V5R1* or higher.
2. You must include a PPT entry for the M204XDM module. Specifying non-swappable and noncancelable is recommended.
3. In the M204XDM job, set the PARM= parameter on the EXEC statement to the subsystem name used in the CRAMINS installation job, which assembles IGCLM244.
4. You must submit the M204XDM job prior to all Onlines. If M204XDM is unavailable and the Online specifies XMEMOPT=X'80', the Online cannot initialize.
5. Set the appropriate XMEMOPT options in your Model 204 Online CCAIN stream. To use CRAM XDM, include X'80' in the XMEMOPT setting and set XMEMSVC to the SVC number used in the XSVCINS installation job. If you linked M204XSVC into your Online instead of installing it as an SVC, do not set XMEMSVC.

Warning: If you make CRAM XDM cancelable and the operator cancels Model 204 XDM, CSA storage is orphaned. **To reclaim the orphaned storage, you must IPL OS/390 or z/OS.**

In Chapter 12: Establishing and Maintaining Security

Login delays

In addition to the LOGTRY and LOGFAIL parameter options, an automatic security feature is automatically invoked for threads on which users make repeated attempts to log in and fail due to invalid ID-password combinations.

When login is required, if an invalid user ID and password combination is supplied, an internal count used in the LOGTRY comparison is incremented. On the fourth-failed login attempt, the login process for this thread begins to slow down. The error count is used as the basis for a calculation that increases according to the function $N*(N-1)$, where N is the number of failed login attempts.

A valid user ID and password combination immediately clears the error count and eliminates the delay. During login service delays, Model 204 also drops the

login service priority level from 255 to 0, and marks the offending thread as pending login due to errors.

LOGFAIL actions still occur. The user can be bumped by a user with operator or system manager privileges. Even if a bump is performed, the failing error count is not reset until a valid login on that thread occurs. Other Model 204 users and Model 204 itself are not affected by the time delay on a failed login, except for time delay of a pending login tying up a thread.

To monitor the system for unsuccessful logins, use the MONITOR, LOGWHO, and STATUS commands. A wait type (WT) of 23 from the MONITOR command indicates that a login for the thread is pending; flags (FLGS) indicate that the user is swappable and bumpable.

In Chapter 15: Using a System Journal and Statistics

The following section was changed for Version 5.1 to reflect the addition of information to CCAJRNL records to detect whether an update unit was improperly written or duplicated. See “CAJRNL size increase” on page 2-6.

Recovery entries (Types 1–6)

Entry types X'01' through X'06' contain information used for recovery. They follow the same general format as all other journal entries. There is *no* type X'07' journal entry.

The format of the recovery entries is meaningful only in the context of the Model 204 roll forward algorithm. However, entry types might be of interest in some statistics gathering applications, as follows:

Type	Contains recovery information from...
1	Discontinuity
2	Begin update unit. More than one type 2 entry can be written for each update unit. The update unit number appears as a 4-byte binary number at offset 16(10).
3	End update unit. The update unit number appears in the same format and location as the type 2 entry.
4	File state information: opened, closed, dumped, file parameter list (FPL) page pre-image logged or open at the time of a checkpoint
5	Checkpoint
6	File update

Revisions to the *User Language Manual*

In Chapter 3: Basic User Language Statements and Commands

In *Chapter 3: Basic User Language Statements and Commands*, in the section on “Loop statements,” introductory paragraphs are rewritten as follows:

Once a set of records or values has been located, the found set can be referred to by using the FOR statement.

The FOR statement introduces a loop. The complete set of statements within the loop is performed once for each record or value located. If no record or values are found, the statements in the loop are skipped.

When a FIND statement refers back to a previous found set and the found set is empty, then the loop for the FIND statement is skipped. Performance for the FIND processing is enhanced because no retrieval is performed.

In Chapter 4: Record Retrievals

In *Chapter 4, Record Retrievals*, in the section on “Pattern matching,” the Escape character subsection is rewritten as follows:

Escape character

The escape character (!) allows a character to be interpreted as a literal character rather than as a pattern matching character. The escape character affects only the next character and is needed for the eleven pattern matching characters—* + ' () - / ! = # @—when you want to treat them as valid literal characters.

The escape character works as shown in the following example.

If pattern is...	Sample matching values
! (800!) - *	(800)-244-3344
J+++.	JUMP. JILT. J%&\$.

In Chapter 11: Flow Control in User Language

In *Chapter 11, Flow Control in User Language*, the “Index loops” section was altered to describe the correct behavior of *exp1*, *exp2*, and *exp3* within a FOR loop.

Index loops

The index loop statement executes a series of statements for each element of an array, for particular dimensions of the array, or for particular elements of the array. In addition to being used with arrays, index loops can be used any place

where a set of statements is to be executed a number of times. If the loop is controlled by a number rather than by a value or a FOR loop, an index loop can be used.

Syntax The format of the index loop statement is:

```
FOR %variable FROM exp1 TO exp2 BY exp3
```

- Where**
- *exp1* indicates the first element to be processed by the index loop. *Exp1* is not allowed to change during processing.
 - *exp2* indicates the last element to be processed by the index loop. *Exp2* is not allowed to change during processing.
 - *exp3* indicates the number by which the element number is to be incremented each time the loop is performed. For example, BY 2 might be used to process every other array element.

The TO and BY phrases can appear in any order. If BY is omitted, the default increment is 1.

Index loop processing

The expressions specified in the index loop are evaluated only once before the first pass through the loop. *Exp1* and *exp2* are evaluated before the loop, so changes will not affect the loop. However, *exp3* can change within the loop.

Model 204 performs the test for index loop completion before the loop body. The increment is performed before each pass of the loop except the first. A branch is allowed out of the loop, but not into it.

In Chapter 18: Program Communications Facilities

In the section titled “TPROCESS (terminal process) communication” on page 18-4, the following text was added:

TPROCESS applications cannot use the SECTRLOG trusted login parameter. Using trusted login results in the following message:

```
M204. 1786: PROCESS- TO- PROCESS NOT SUPPORTED ON THIS  
THREAD
```

In Chapter 20: Global Features

The following section of *Chapter 20: Global Features* was written to clarify the behavior of the various CLEAR statements.

REMEMBER statement

The purpose of the REMEMBER statement is to store the processing position in a FOR loop. Each REMEMBER statement creates a GTBL entry of variable

length. See the chapter on Global Features in the *Model 204 User Language Manual* for detailed descriptions of GTBL entries. If you remember a position as GLOBAL, the entry remains in GTBL after the current request ends. Model 204 clears nonglobal REMEMBER positions at the end of each request.

Syntax REMEMBER [GLOBAL] *position_name*
[IN *foundsortset_name* | ON *list_name*]

- Where**
- The GLOBAL option retains the REMEMBER position after the current request terminates. If GLOBAL is not specified, the position is temporary and is cleared from GTBL after the current request terminates.
 - *position_name* is a unique object name you assign to the REMEMBER position.
 - The *foundsortset_name* or *list_name* must be identical to the list_name or label specified in the FOR loop:
 - If you are processing a found set or a sorted set, IN *foundsortset_name* is the label of the FIND statement that generated it.
 - If you are processing a list, ON *list_name* is the name you gave to the list when you originally declared it.

Usage notes The following rules apply to the use of the REMEMBER statement:

- Can appear only within a FOR loop, but not a nested FOR loop. The file context must be identical to the context of the FOR loop.
- Is invalid in ad hoc group context, if used with the GLOBAL option.
- The REMEMBER statement is incompatible with FOR EACH RECORD IN ORDER BY clauses.

POSITION statement

A POSITION statement recalls a remembered position, so you can resume FOR processing in a list or found set that was terminated at an earlier time. For example, if you remembered the position at record number 5, then your foundset is positioned at record number 6 for further processing.

Syntax POSI TI ON { FOUNDSET *foundsortset_name*
| LI ST *list_name* }
[AT] *position_name*

- Where**
- The *foundsortset_name* or *list_name* must be identical to the list or label name used in the FOR statement that follows.
 - If you are about to resume the processing of a found set, *foundsortset_name* is the label of the FIND statement that generated it.

- If you are about to resume the processing of a list, *list_name* is the name you gave to the list when you originally declared it.
- *position_name* is a name that you assigned to the remembered position when you stored it.

Clearing the GTBL work area

When you log off, all global objects and global variables are cleared. If you want to clear some or all global objects or global variables from GTBL during your working session, prior to logging out you can issue:

- CLEAR statement
- CLEARG and/or CLEARGO commands
- \$DELG function call to selectively delete global variables
- UTABLE command that changes the size of FTBL, XTBL, or GTBL clears all global objects of any type
- RESET command for the GTBLEHASH and GTBLPCT parameters

Using the CLEARG and CLEARGO commands

Use this command	To Remove...
CLEARG	Only global string variables.
CLEARGO	All global objects: images, screens, menus, found sets, lists, and temporary and permanent positions from GTBL; it does <i>not</i> clear global string variables.

The CLEARGO command takes no arguments. See the *Model 204 Command Reference Manual* for syntax and detailed information on the CLEARG and CLEARGO commands.

Using the \$DELG function

You can use the \$DELG function to delete global variables created by either the \$SETG function or \$INCRG function.

For syntax and usage of the \$DELG function, see the *Model 204 User Language Manual*.

The \$DELG function deletes a single or group of similar global variables, which releases and compacts the GTBL space for reuse.

Using the CLEAR statement

You can use the CLEAR statement to clear global objects of the same type, all global variables, or individual global objects, as shown in the following table:

The statement	Clears...
CLEAR [<i>type-of-object</i>] OBJECTS	A class of global objects or all global objects. Global variables are not cleared, because they are not objects.
CLEAR GLOBAL	Individual global objects from GTBL.
CLEAR GLOBALS	All global variables. You cannot selectively clear individual global variables with the CLEAR statement.

Syntax The format for the CLEAR statement is:

```
CLEAR { [ [ ALL | TEMP | LISTFDST | POSITION ] [ GLOBAL ] ]
OBJECTS
| GLOBAL { IMAGE | SCREEN | MENU | LIST | FOUNDSET
| POSITION [ PERM | TEMP ] }
| ' objectname' | %variable }
| GLOBALS
```

- Where**
- *ALL* clears all permanent and temporary global objects, including: found sets, lists, images, menus, positions, screens, and sorted sets from GTBL.
 - *TEMP* clears only global objects explicitly declared as TEMP in your request, including: images, menus, and screens from GTBL.
 - *LISTFDST* clears all global found sets, lists, and sorted sets from GTBL.
 - *POSITION* clears all permanent and temporary positions from GTBL.
 - *OBJECTS* specifies CLEAR command is operating on the object type you specified or all types of objects.
 - *GLOBAL* clears a specific global object from GTBL. You cannot clear a global variable created with the \$SETG function using this form.
 - *objectname* is the literal name of the specific global object to be cleared from GTBL. Enclose *objectname* in single quotation marks.
 - *%variable* contains a value that specifies the global object to be cleared from GTBL.
 - *GLOBALS* clears all global variables created with the \$SETG function from GTBL. This form does not clear any global objects.

Usage notes The following statements clear all permanent and temporary global found sets, images, lists, menus, positions, screens, and sorted sets:

```
CLEAR OBJECTS
```

```
CLEAR ALL OBJECTS
```

```
CLEAR ALL GLOBAL OBJECTS
```

```
CLEAR GLOBAL OBJECTS
```

Clearing global found sets and lists

You can clear global found sets and lists in GTBL without logging off using the following examples:

- Clear a specific global list or found set by issuing one of these statements:

```
CLEAR GLOBAL LIST ' object name'
```

```
CLEAR GLOBAL LIST %variable
```

```
CLEAR GLOBAL FOUNDSET ' object name'
```

```
CLEAR GLOBAL FOUNDSET %variable
```

- Clear all global found sets and lists with the statement:

```
CLEAR LISTFDST [ GLOBAL ] OBJECTS
```

(Global found sets and lists are among the global objects cleared by the CLEARGO command.)

- Close a file or group with which a global list or found set is associated. This includes the file close processing done when exiting or stopping a subsystem.
- Issue a UTABLE command that changes the size of FTBL, XTBL, or GTBL.

If you clear a global list or global found set, then any request that uses the global and does not have the FIND statement that creates the global object, receives either of the following error messages:

```
M204. 0301 REFERENCED STATEMENT LABEL UNDEFINED
```

```
M204. 0311 UNACCEPTABLE STATEMENT REFERENCE
```

In Chapter 27: User Language Functions

In *Chapter 27: User Language Functions*, the “\$CHKPAT” section is rewritten as follows:

\$CHKPAT function

The \$CHKPAT function verifies the syntax of a pattern. \$CHKPAT returns a null string, if the pattern is syntactically correct, or an error message, if it is not.

Without the \$CHKPAT function, pattern syntax errors caused cancellation of the request or required the coding of awkward ON units.

The \$CHKPAT function supports language-sensitive specification of patterns through an optional second parameter, such as the \$ALPHA and \$ALPHANUM functions.

All characters X'00' through X'FF' are valid in a pattern presented to the User Language pattern matcher. The \$CHKPAT function no longer invokes either of the following messages when it encounters these characters.

M204. 1688: *errortype* IN PATTERN '*pattern*' AT
CHARACTER *char*

M204. 1689: *errortype* IN PATTERN '*pattern*' AT
CHARACTER *char*

5

New and Revised Messages

In this chapter

- New and revised Model 204 numbered messages
- New Model 204 unnumbered messages
- New SQL messages

New and revised Model 204 numbered messages

0146 filename - error description

Model 204 has detected the error indicated by message M204.0146 for the specified file. The possible error conditions are:

Error condition	Explanation of condition
DEFERRED UPDATE CHKP MISSING	Model 204 cannot locate a record corresponding to the roll-back-to-checkpoint in (one of) <i>filename</i> 's deferred update dataset(s).
DEFERRED UPDATE READ ERROR	Model 204 encountered an I/O error while trying to locate the roll-back-to-checkpoint in (one of) <i>filename</i> 's deferred update dataset(s).
DEFERRED UPDATES MISSING	File was being updated in deferred update mode by the run being recovered and Model 204 cannot open the deferred update dataset(s).
IS ON A READ- ONLY DEVICE	(Model 204 CMS version only) <i>Filename</i> is on a device for which the Model 204 service machine has read-only access.
MISSING	Model 204 cannot open <i>filename</i> 's dataset(s).
ROLL BACK INFORMATION IS OBSOLETE	Either <i>filename</i> was updated after the run being recovered ended, or the wrong RESTART dataset was specified. Further information is given in M204.2565, which follow this message.

The action taken by Model 204 depends on the value specified on the ERROR option of the RESTART command.

ERROR option specified	Model 204 action
ERROR CONTINUE	Recovery continues.
ERROR OPERATOR or no ERROR	Operator is prompted to decide whether or not recovery should continue. If recovery continues, files that cannot be recovered are bypassed.
ERROR STOP	Recovery does not continue.

The following table lists possible reasons for failure to open files during recovery, and what action to take.

Reason	Corrective action you can take...
DEFERRED UPDATE CHKP MISSING	Check for an incorrect DD, DLBL, or FILEDEF statement for the deferred update dataset(s) for this file.

Reason	Corrective action you can take...
DEFERRED UPDATE READ ERROR	Attempt to copy the deferred update dataset(s) for this file to another physical location. If the read error persists, this file cannot participate in this recovery.
DEFERRED UPDATES MISSING	Check for an incorrect or missing DD, DLBL, or FILEDEF statement for the deferred update dataset(s) for this file.
IS ON A READ- ONLY DEVICE	(Model 204 CMS version only.) Correct the access mode for the device in question.
MISSING	Make sure the DD, DLBL, or FILEDEF statements are correct for this file.
ROLL BACK INFORMATION IS OBSOLETE	<p>If all files report this reason, check that the correct checkpoint dataset was specified as the RESTART dataset for recovery.</p> <p>If only one or a few files report this reason, it means that they were updated after the end of the job being recovered. Such files cannot participate in this recovery.</p> <p>Message M204.2565, which follows this message, provides information which may assist in determining when the files were updated.</p>

System manager response: Correct the error and rerun Model 204 recovery.

Operator response: Notify the system manager of this message.

E 0 4 OPR

0169 BUG.. WHILE REAPPLYING RF ENTRY TYPE entry type, SUBTYPE entry subtype, FROM UPDATE UNIT update unit number TO FILE filename [RECORD: nnn]

A Model 204 system error occurred. Model 204 encountered an error while attempting to reapply an update during roll forward. When available, a record number is included in this message.

System manager response: Contact CCA Customer Support. Please have the following documentation available:

- Complete text of the error message
- Audit trail from both this run and the previous run
- Snap

If follow-up diagnosis is necessary, CCA Customer Support requires copies of the following items:

- CCARF or CCAGEN dataset
- RESTART dataset, if this is RESTART recovery

E 0 4 DUMP SNAP AUDITAD

0232 FILE OR GROUP ISN'T OPEN - name

This message can be generated for any of these reasons:

- A CLOSE command for a file or group not currently open was issued.
- An IN *filename* clause or IN *groupname* clause was issued as part of a User Language statement. The specified file or group was not currently open. The compilation continues, but the request is not evaluated.
- The OUTPUT file context specified for the COPY PROC command was a file or group not currently open.

Response: In the first case, no response is required. In the second case, make sure that all files accessed in the IN clause are open or eliminate those which are not open from the IN clause and retry. For the third case, open the file.

E 0 4 ECHO AUDITER

0434 NJBUFF RESET TO %C

The NJBUFF parameter was reset to an invalid value. NJBUFF must be equal to:

- One (1), or
- Value of the NSERVS parameter, plus NSUBTKS parameter, plus 1.

NJBUFF specifies the number of journal buffers allocated by Model 204 during initialization.

Response: Notify the system manager.

System manager response: Reset the NJBUFF parameter to a valid value and rerun Model 204.

I 0 0 AUDITAD

0441 CCATEMP FULL: operation | pooltype

The *operation* being performed required an extra page from CCATEMP, but there were no free pages available. CCATEMP is defined by the system manager, and provides a set of work pages that are assigned on request to various users. If a request gets this message, it might be because some other user is holding a large number of CCATEMP pages.

The message specifies either the *operation* being performed, or the *pooltype* (section) of CCATEMP that was full.

operation can be one of:

- \$BLDPROC
- "USE PROCEDURE" COMMAND
- WHILE INITIALIZING SORT

- FOR FRV KEYS
- WHILE GENERATING RECORDS TO BE SORTED
- WHILE GENERATING ORDERED INDEX SORTED RECORDS
- WHILE PERFORMING SORT

pooltype can be one of:

- SMALL MODEL PAGE POOL
- SMALL MODEL PAGE POOL AND EXPANSION PAGE POOL

The current request is cancelled.

If the *operation* was \$BLDPROC or "USE PROCEDURE" COMMAND, then the temporary procedure that was being written is deleted, except for its very last CCATEMP page, which is retained for information purposes.

In some cases the user is restarted and, if the user was in the process of updating any files, Model 204 marks them physically inconsistent.

CCATEMP pages below 64K are allocated from the small model pool. CCATEMP pages above 64K are allocated from the expansion area pool.

The *pooltype* specified in the message indicates whether you exceeded the capacity of the small model pool or both the small model and expansion area pools.

Response: Notify your system manager.

System manager response: Recover the physically inconsistent files using one of the following procedures:

- ROLL BACK/ROLL FORWARD processing
- File reorganization procedure described in the *Model 204 File Manager's Guide*
- The RESTORE command for restoring previously dumped copies

This message is informational, but it indicates a need to increase the space allocation for CCATEMP. As the size of CCATEMP is fixed at initialization, it is necessary to bring down Model 204 to increase it.

E 60 60 CANCEL AUDITAD

0473 UNABLE TO INITIALIZE <subsystem name> or <SUBSYSTEMS>

If the message reports a specific subsystem, that subsystem could not be initialized because of an error.

If the message is issued by User 0 during Model 204 initialization and reports SUBSYSTEMS, then APSY initialization failed, and subsystems will not be available during this run.

Response: Notify the system manager.

System manager response: If the message reports a specific subsystem, look at the audit trail or issue the TEST subsystem command to determine which error caused subsystem initialization to fail.

If the message reports SUBSYSTEMS, then the server tables for User 0 are not large enough to initialize subsystem handling. Change the User 0 parameters so that its server table sizes are at least as large as the sizes listed in the description of message M204.1457, and then restart Model 204.

E 0 4 AUDITER

0555 EXTRA NUMERIC FIELD IGNORED FOR FIELDNAME= fieldname IN TABLE B RECORD NO.= record-number

The field name specified in the message is a NUMERIC RANGE field. A User Language request, FLOD instruction, or Host Language Interface function illegally attempts to store a second value for the field name in a record. The attempt fails because two values for NUMERIC RANGE fields cannot exist at once.

The message may list the field name only, not the Table B record number.

Response: Delete the old field name, add the new field name, and try again.

E 40 40 AUDITMS

0556 EXTRA NUMERIC FIELD IGNORED FOR FIELDNAME= fieldname IN TABLE B RECORD NO.= record-number

A NUMERIC RANGE field can exist only once in each record. The request is attempting to insert a second NUMERIC RANGE field for the field name in the record.

The message may list the field name only, not the Table B record number.

Response: Correct the statement and retry.

E 98 98 AUDITMS

0572 NEW VALUE NULL, PAD, OR TOO LONG FOR fieldname

The string to be stored in a preallocated *fieldname* is one of the following:

- Longer than the maximum length of the field
- Null (or entirely equal to the PAD character specified for the field)
- Terminates with a pad character.

Response: Correct the statement and retry.

E 0 4 CANCEL AUDITER

0584 FILE IS IN USE: dataset name

This message appears when an Online run of Model 204 encounters a file enqueueing conflict.

Response: Try the procedure or command later, or notify your system manager if you suspect the run of Model 204 that had the enqueue has crashed.

Operator response: Notify the system manager if there are multiple occurrences of this message

E 0 4 OPR

0621 recovery-status

This informational message specifies the recovery status of the file being opened. The recovery status is indicated as follows:

- FILE *filename* REGENERATED TO CHECKPOINT OF *dd mmm yyyy hh:mm:ss.th*
- FILE *filename* REGENERATED TO DISCONTINUITY OF *dd mmm yyyy hh:mm:ss.th*
- FILE *filename* ROLLED BACK TO CHECKPOINT OF *dd mmm yyyy hh:mm:ss.th*
- FILE *filename* ROLLED BACK TO DISCONTINUITY OF *dd mmm yyyy hh:mm:ss.th*

where:

- *filename* specifies the file being opened.
- *dd mmm yyyy* is the day, month and year.
- *hh:mm:ss.th* is the time in hours, minutes, seconds, and thousandths of a second.

The message is generated if the FISTAT parameter indicates that the file has been recovered (X'10').

I 0 0 AUDITMS

0622 recovery-status

This message specifies the recovery status of the file being opened. The possible status messages are:

- UPDATE *nn* OF *dd mmm yyyy hh:mm:ss.th* WAS HIGHEST UPDATE REAPPLIED TO *filename* BY *recovery method*
- UPDATE *nn* OF *dd mmm yyyy hh:mm:ss.th* WAS LOWEST UPDATE BACKED OUT OF *filename* BY *recovery method*

- UPDATE *nn* OF *dd mmm yyyy hh:mm:ss.th* WAS LOWEST UPDATE PARTIALLY REAPPLIED TO *filename* BY *recovery method*

Where:

- *nn* is the update unit number.
- *dd mmm yyyy* is the day, month and year.
- *hh:mm:ss.th* is the time in hours, minutes, seconds, and thousandths of a second.
- *filename* is the name of the file being opened.
- *recovery method* is either ROLL FORWARD or REGENERATE.

The message is generated only if the FISTAT parameter indicates that the file has been recovered (X'10').

I 0 0 AUDITMS

0669 SERVER INITIALIZATION ERROR: \$STATUS=nn \$STATUSD=nn

A request for a server program, such as, Connect ★, a Horizon process, or PQO failed.

Response: Consult the *Horizon: Intersystem Processing Guide* for an explanation of the STATUS and STATUSD codes.

E 0 4 AUDITMS

0710 TOO MANY OCCURRENCES FOR FIELDNAME= fieldname IN TABLE B RECORD NO.= record-number

A File Load program attempted to add too many occurrences of *fieldname* in the preallocated section. No occurrence was added, and the execution of FASTLOAD continues.

File manager response: The preallocated section can only contain as many occurrences of a field as specified in the OCCURS attribute in the field description. If you want additional occurrences accepted into the Model 204 file, you must reorganize the file and increase the number given in the OCCURS clause for *fieldname*, or omit the OCCURS clause (and optional accompanying LENGTH clause) to allow this field to have a variable number of occurrences on each record.

E 0 4 AUDITER

0711 INCOMPRESSIBLE VALUE FOR FIELDNAME=%C IN TABLE B RECORD NO.=%C

An attempt was made by a File Load program to update a field in the preallocated section of a Model 204 file with a value which is not compressible to a Model 204 binary or floating point value. The update of *fieldname* is ignored, and the execution of the File Load program continues.

File manager response: There is probably an error in the input data. If you want *fieldname* to be able to accept incompressible values, reorganize the file and remove *fieldname* from the preallocated section by taking the OCCURS attribute out of the field description, or by giving the field the BINARY, CODED and OCCURS attributes.

E 0 4 AUDITER

0712 VALUE TOO LONG FOR FIELDNAME= fieldname IN TABLE B RECORD NO.= record-number

Where:

- *fieldname* specifies the name of the field within the record.
- *record-number* specifies the Model 204 record number that incurred the error.

A File Load program has encountered an attempt to update *fieldname*, a field in the preallocated section, with a value that exceeds the defined length for *fieldname*. The update to *fieldname* did not occur, but the File Load program continues.

File manager response: You can do one of the following:

- Correct the FLOD program or input file.
- Increase the size of the preallocated field which is specified in the LENGTH attribute of the field description.
- Leave this field out of the preallocated section by not specifying the OCCURS attribute.

Note: Changes to the preallocated section require reorganizing the file.

E 0 4 AUDITER

0858 GROUP group name

or

GROUP group name OPENED (-- NO UPDATES ALLOWED)

This is an informational message.

The first format indicates that Model 204 processed the named group. It may be preceded by other related messages.

The second format indicates that Model 204 successfully opened all the files that make up the specified group.

I 0 0 AUDITMS

0955 IFAM2 CRAM OPEN ERROR, IFAM HALTED- CHANNEL= channelname

An attempt has been made to open a channel that is already open.

This error could also occur if XMEMOPT=X'8X' has not been specified and the system has been configured to use XDM CRAM.

Response: Contact the system manager.

System manager response: The most common reason for a CRAM open failure is using a CRAM channel name that is already open. This can happen if there are multiple copies of Model 204 running in the same system. Review the error codes listed with message M204.0956. If the problem persists, contact CCA Customer Support and have the following information available:

- Snap
- Audit trail

Operator response: Issue the IFAMOPEN command to recover.

E 0 4 DUMP VIEWERR SNAP AUDITAD OPR

1000 Variable-name VARIABLE CANNOT BE RESOLVED. VALUE = value

The *value* of a *variable-name* variable (:%variable) is not the name of any variable defined for the request being evaluated.

The *value* should be the name of a screen or menu item or, if an initial value is being declared for a variable, the *value* must be defined prior to its use.

Response: If *value* starts with a percent sign (%), remove the percent sign. Otherwise, make the value the name of a screen or menu item. If an initial value is being declared for a variable, define the variable prior to its use in the initial clause. Retry the request.

E 0 4 CANCEL AUDITER

1006 FILE filename MUST BE DEFINED WITH A DATASET TYPE OF DIRECT OR SEQUENTIAL

The parameter DIRECT or SEQUENTIAL is missing, or an incorrect dataset type (file organization method) has been specified in your DEFINE DATASET command.

System manager's response: Check the DEFINE DATASET command to make sure that all the required parameters were entered correctly. Retry the command again. If this error occurs often, contact Customer Support.

E 0 4 AUDITER

1063 FILE OR GROUP name NOT OPEN, COMMAND REJECTED

You issued a DEFAULT command or used an IN clause for a command that requires a file or group name. The file or group is not currently open.

Response: Correct the file or group name specified. If it is correct, open the file or group before retrying the command.

E 0 4 AUDITER

1069 ALLOCATE/FREE FAILED WITH RETURN CODE return_code ERROR REASON CODE value INFORMATION REASON CODE value

An error was detected during a dynamic allocation (SVC 99). The return and reason codes are described in the IBM manual *MVS Programming: Authorized Assembler Services Guide, GC28-1763*, chapter 26.

Response: Correct the error and retry the command.

E 0 4 AUDITER

1076 DO YOU REALLY WANT TO action

You are being asked to verify the action specified in the message. An explanation of the question often appears on the preceding line of output.

- For Online APSY users, the defaults listed are provided.
- From Online non-APSY users, this question requires an answer of Y (yes) or N (no). Any response other than Y or N is ignored and generates the following:

M204. 1078: PLEASE REPLY "Y" OR "N".

Context: following a circumstance

When following a request to reset FTBL, XTBL, or GTBL, this message gives you a chance to quit before deleting the data in those tables. The default is Y (yes).

The message DO YOU REALLY WANT TO CLEAR THE PHYSICALLY BROKEN BIT? following a reset of FISTAT is checking whether you want to change the state of a file Model 204 has marked as physically broken. Specifying Y resets FISTAT to the specified value; specifying N does not change the setting of FISTAT. The default is Y.

The message DO YOU REALLY WANT TO EDIT INTO EXISTING PROCEDURE? implies that your action to save a procedure will overwrite an already existing procedure with the same name. Specifying Y, the default, overwrites the existing procedure. Specifying N does not overwrite the existing procedure, and you are given a chance to save the procedure under another name.

Context: following a message

Following...	Indicates that you...	Specify Yes to...	Specify No to...
M204.0569	Exceeded the default maximum for the number of Table B records to search.	Perform a direct search of Table B	(Default) Cancel your request.
M204.0602	Attempted to access a file that another user has in exclusive mode.	Attempt to access the file again.	(Default) Cancel your request.
M204.0846	Attempted to access a group which another user has in exclusive mode	Attempt to access the group again	(Default) Cancel your request
M204.1140	Attempted to create a new procedure with the PROCEDURE command, but one with that name already exists.	(Default) Overwrite the old procedure	Give yourself a chance to rename the new one
M204.1216	Attempted to access records which another user has in exclusive mode.	Attempt to access the records again	(Default) Cancel your request.
M204.1227	Used a table to control the sharing of data below the file level is full	Attempt to allocate space again.	(Default) Cancel your request.
M204.1332	Tried to exceed a prespecified maximum, such as MCPU.	Cancel the request	(Default) Try the request again.

Response: Decide which action you wish to take and reply with either Y (yes) or N (no).

P 0 0 AUDITAD

1190 MAXBUF SET TO SAME VALUE AS MINBUF = value

The explicit or default value for the MAXBUF parameter was less than the explicit or default value for the MINBUF parameter, so MAXBUF has been automatically reset to the MINBUF value.

System manager response: This is an informational message and no action is required.

I 0 0 AUDITMS

1250 END OF INPUT DATA

No more lines could be read from the user's primary input device. This message normally appears if User 0's input (CCAIN) does not contain an EOJ command. In this case, the entire Model 204 run is terminated as though an EOJ command were present. For users other than User 0, this message is followed by message:

M204. 1023: USER SUSPENDED.

System manager response: If this message appears prematurely for User 0 in a multiuser ONLINE run, use a HALT or *SLEEP command to delay further processing of User 0's input. If this message appears for a terminal user, contact Model 204 Customer Support and have the audit trail available.

E 0 4 AUDITER

1289 (LGTBL - GTBLHASH*10) MUST BE AT LEAST 288 FOR SUBSYSTEM PROCESSING

You cannot use application subsystems when the amount of free space in the global variable table (GTBL) is less than 288 bytes. Model 204 initialization will terminate abnormally.

System manager response: If you do not require application subsystems in this run, turn off the SYSOPT=X'01' setting. Otherwise, set the LGTBL and GTBLHASH parameters on the User 0 parameter line so that (LGTBL minus (GTBLHASH times 10)) is at least 288 bytes. Rerun Model 204.

E 0 4 AUDITER

1300 RESTART COMMAND REQUIRES CHECKPOINT LOGGING - RUN TERMINATED

One of the following errors has occurred:

1. RESTART recovery requires a CHKPOINT (or CHKPNT) definition.
2. ROLL FORWARD was specified on the RESTART command and the X'01' bit of the RCVOPT parameter was not set.
3. The CHKPOINT dataset has failed to open.

System Manager response

1. Define a CHKPOINT (or CHKPNT) dataset.
2. Set the RCVOPT parameter's X'01' bit on.
3. Allocate a new CHKPOINT dataset.

Operator response

Notify the system manager.

E 52 52 AUDITAD OPR

1332 LONG REQUEST - EXCEEDED %C <= %C

This informational message accompanies a message that indicates that the parameter value exceeds or equals maximum values specified for the following parameters:

Parameter	Maximum number of...
MBSCAN	Direct, Table B searches
MCNCT	Second per request of wall clock time
MCPU	Milliseconds per request of CPU time
MDKRD	Disk reads per request
MDKWR	Disk writes per request
MOUT	Lines printed per request
MUDD	Output lines written to a directed output, USE, dataset per request
SQLBSCAN	Read/write operations without user restart
SQLLPLIM	Read/write operations without swapping

The message DO YOU REALLY WANT TO CONTINUE? follows.

Response: Reply Y or N to the DO YOU REALLY WANT TO CONTINUE? message. The default is N. A reply of Y continues the request until the value or values are again exceeded. A reply of N terminates the request. To prevent the problem in the future, reset the appropriate parameter(s).

E 0 4 AUDITER

1342 UNABLE TO RUN CCASYS PROCEDURE: procname | errortype

This message is issued if an error occurs when Model 204 tries to include one of the CCASYS procedures used to control the application subsystem facility. An additional explanatory message may be produced.

The error may occur for the following reasons:

- Compilation errors have occurred.
- One or more server tables is too small.
- GTBL is full.
- The attention key was hit.

Response: Check that the server tables are at least as large as the sizes listed in the description of message M204.1457.

E 0 4 AUDITER

1427 INVALID STOPPING POINT CLAUSE

The syntax of the stopping point specified after the keyword 'TO' for some file in the REGENERATE command is invalid.

Check the syntax against that outlined in the *Model 204 Command Reference Manual*, or remove the TO clause and allow media recovery to use the default stopping point (last update).

If ONEPASS was specified, the following TO clauses are invalid:

- TO LAST CHECKPOINT
- TO LAST CHECKPOINT BEFORE ...
- TO LAST UPDATE BEFORE ...

Rerun media recovery with the corrected syntax.

E 0 4 AUDITER

1457 UNABLE TO SCAN LIST OF SUBSYSTEM NAMES

Model 204 is unable to check the command you entered against the list of application subsystems defined at your installation. The command is not executed. This message can be caused by any of the following:

- Subsystem processing is not enabled for this run, because the system manager did not set the SYSOPT=X'01' bit. Subsystem processing cannot be enabled without restarting Model 204.
- Subsystem processing is not enabled for this run, because an error occurred during Model 204 initialization. This is indicated in the audit trail by message M204.0473, issued by User 0 during initialization. Subsystem processing cannot be enabled without restarting Model 204.
- There is insufficient space in the server tables of the user who issued the command.
- A new Model 204 release was installed, but a matching Dictionary/204 upgrade was either omitted or not performed properly.
- There is some problem with the CCASYS file.

Response: Subsystems cannot be started unless there are 288 unused bytes in the global variable table (GTBL). The following sequence of commands may assist you in verifying and creating adequate GTBL space:

Command	Purpose
CLEARG	Deletes all global string variables from GTBL

Command	Purpose
CLEARGO	Deletes all global objects from GTBL
VIEW LGTBL	Displays the allocated size of GTBL
VIEW GTBLHASH	Displays the number of GTBL hash buckets
UTABLE LGTBL=<value>	Resizes GTBL to the indicated size

If GTBLHASH is zero, LGTBL=288 is sufficient. However, if GTBLHASH is nonzero, LGTBL should be at least $(288 + (10 * GTBLHASH))$. The extra space represents 8 bytes of control information per hash bucket, plus an allowance because hashing cannot distribute values exactly evenly.

If this error occurs when GTBL is large enough, use the VIEW command to verify that the sizes of the server tables are at least this size:

Server table	Size
LNTBL	50
LQTBL	120
LSTBL	250
LVTBL	50
LFSCB	256

If the values of these parameters are adequate when this message is issued, contact the system manager.

System manager response: Look at the audit trail for an earlier message identifying the cause of the error. Verify that CCASYS was installed properly. Was Dictionary/204 correctly installed or upgraded for this release? If CCASYS is physically inconsistent, reorganize the file. The reorganization should include a reload of data and the procedures stored in CCASYS.

E 0 4 AUDITER

1718 FRN IN GROUP CONTEXT REQUIRES “MEMBER” CLAUSE

The FOR RECORD NUMBER statement may not be used in group context unless the MEMBER clause is also used.

Response: Either change the request to use a file context on the FOR RECORD NUMBER statement or add a MEMBER clause to the group context FOR RECORD NUMBER and retry the request. Refer to the *Model 204 Command Reference Manual* for more information on Model 204 files and groups.

E 0 4 ECHO AUDITER

1743 entity-name NOT FOUND

An attempt to open an entity failed because the definition for this entity does not exist.

Response: Verify the spelling of the entity name. If the name is spelled correctly, then the entity was not defined.

- A Model 204 file or group is defined via the CREATE command.
- An external dataset must have a DD statement in the JCL (or, in CMS, a FILEDEF in the EXEC).
- A DEFINE command is required for a VSAM dataset or a process.
- For connectivity to Model 204, issue a DEFINE PROCESSGROUP or DEFINE REMOTE command to identify a remote node (Horizon for VTAM) or an IP address (Horizon for TCP/IP).

Contact the system manager for the entity definition.

System manager response: Define the entity, if necessary.

E 0 4 SAVE VIEWERR NOTERM

1925 %C OPTION REQUIRES PARAMETER CFRLOOK=1

A MONITOR command was issued that requires Critical File Resource Statistics to be collected.

The displayed keyword of the MONITOR command is valid only if *conflict counts* are being kept.

Response: Add the CFRLOOK=1 parameter setting to your CCAIN stream.

System manager response: Reset CFRLOOK=1.

E 0 4 AUDITER

1992 RECOVERY: PROCESSING ROLL FORWARD BLOCK# %X %C

This message is informational only.

It details the sequence number and date/time stamp of the current journal block being processed by roll forward. The message will be issued once for each changing hour.

I 0 0 OPR

2009 UNEXPECTED CONVERSATION STATE state-name

A CCA-supplied transaction program issued an APPC verb that is not appropriate in its current conversation state.

System manager response: Forward the CCA audit trail to CCA Customer Support.

E 0 4 OPR

2079 MINBUF RESET TO (NSERVS + NSUBTKS) * MAXOBUF + 15 =nnnn

The minimum valid value for MINBUF is the number of servers (the value of the NSERVS parameter) plus the maximum number of pseudo-subtasks (the value of NSUBTKS parameter), times the maximum number of open disk buffers per server (the value of the MAXOBUF parameter), plus fifteen.

However, a smaller value was specified for MINBUF or the MINBUF default value was too small. The value of MINBUF was automatically reset to its minimum allowable value as indicated by this message.

This is an informational message. No action is required.

I 0 0 AUDITMS

2115 DATA RECEIVED WHILE WAITING FOR CEB RESPONSE - RH=req-header RU=req-unit

Horizon has received data from the session partner after initiating the end of the session, but before the end of the session has been acknowledged by the session partner.

- *Req-header* shows the 3-byte SNA Communications Server request header, in hexadecimal.
- *Req-unit* shows the first 16 bytes of the SNA Communications Server request unit, in hexadecimal.

This is a normal event that can be expected to occur sometimes at the end of a session. No action is required; Horizon discards the late data.

I 0 0 AUDITMS

2116 %F DOES NOT CONTAIN AN ORDERED INDEX

The REORG 01 command was issued for a file that does not contain an ordered index.

File manager response: Check to make sure that you specified the correct file.

E 0 4 AUDITER

2143 INSUFFICIENT MAIN STORAGE FOR SPCORE SPECIFICATION below 16MEG LINE, SPCORE = nnnnn, MINBUF = nnnnn, MAXBUF = nnnnn

The IOS branch entry feature is *not* in use, but the amount of SPCORE requested is unavailable below the 16 megabyte line.

Response: Contact the system manager.

System manager response: Either reduce the amount of spare storage requested (SPCORE), or reduce the size of some other data structure requiring storage below the 16 megabyte line, or use the IOS Branch Entry feature.

E 60 60 AUDITAD OPR

2253 SUBSYSTEM subsystem-name, record-type - RECORD CONTAINS INVALID DATA

A subsystem definition record contained invalid data. The possible values for the record type are:

- SCLASS HEADER
- FILE/GROUP
- SDEF
- UNSPECIFIED

System manager response: Check the subsystem definition in CCASYS, using SUBSYSMGMT. If the problem is not apparent, save the audit trail and contact CCA Customer Support.

E 0 4 AUDITER

2288 TCP/IP CLEANUP HAS CLOSED SOCKET socket-number FOR USER userid

This message documents the fact that an open socket has been closed automatically as part of a cleanup of TCP/IP resources owned by the thread.

I 0 0 AUDITAD

2368 INCLUDED FROM COMMAND LEVEL

or

INCLUDED FROM SUBSYSTEM name

This is an informational message that is generated when the failing statement is entered at the Model 204 command prompt.

I 0 0 NOTERM

2378 SECURITY TRUSTED LOGIN FEATURE DISABLED

The Security Trusted login feature was disabled, because this Model 204 job is not running with one of the following security interfaces: ACF2, Security Server (formerly RACF) or TOPSECRET. The Model 204 job starts normally, but cannot take advantage of the Trusted Login Feature.

A rare possibility is that there is not enough storage available to create a work area needed for the Trusted Login feature. If this happens, then the Model 204

job cannot finish the initialization process, since there is more storage to GETMAIN, such as the buffers.

System manager response: First, verify that the SECTRLOG was correctly specified on the proper IODEV. If it was, then determine why a Security Interface was not active. If there is a shortage of storage, then adjust the region size to allow for extra storage. Then rerun the job.

For more information on the Security Trust Login feature see the *Model 204 Security Interfaces Guide* and the *Model 204 System Manager Guide*.

Operator response: Inform the System Manager and the Security Officer.

E 0 4 AUDITAD NOTERM

2463 **COMMAND name_1 IS AN ALIAS OF name_2, RESETTING IT ALSO**

The RESET COMMAND command is being used to change the USE DD options and/or the command privileges for the command *name_1*. This command has *name_2* as an alias.

Response: None. The system automatically changes the alias to match.

I 0 0 AUDITMS

2478 **operation REJECTED, WOULD OVERWRITE CURRENTLY ACTIVE PROC**

A request to modify a currently active temporary procedure, that is, one that has been included or is the target of concurrent USE PROC or \$BLDPROC requests, was denied.

Operation specifies the type of action attempted: \$BLDPROC or USE PROC.

You can modify temporary procedures from within a User Language request by either the USE PROCEDURE command or the \$BLDPROC function. If the target procedure for either of these operations is currently active—that is, has been included or is the target of a concurrent USE PROCEDURE command or \$BLDPROC function, or is indirectly modified by one—then the request is rejected.

Response: Direct the output to an inactive temporary procedure or issue the USE PROCEDURE command from the command level, not from inside a User Language procedure. When using:

- \$BLDPROC function, you can specify any inactive temporary procedure.
- USE PROCEDURE command, the valid temporary procedures vary with the type of action requested. When the action keyword is:

Keyword	The temporary procedure specified
APPEND or REPLACE	Cannot be currently active.

Keyword	The temporary procedure specified
INSERT	Must have an ID number greater—that is, closer to zero—than the number specified in the USE PROCEDURE command.
SHIFT	Cannot be active.

E 0 4 ECHO AUDITER

2486 %C: TABLED FULL. PAGE ALLOCATED FROM TABLED RESERVE AREA

Table D of this file is full and FISTAT has been set to X'08'. This occurred because Model 204 allocated a page from the Table D reserve area. Refer to the DPGSRES parameter for more detail on Table D reserve area.

Processing continues. Marking the file full prevents other users from starting requests that update Table D, making it more likely that all requests in progress complete normally. Refer to the *Model 204 File Manager's Guide* for more detail.

File manager response: Increase the size of Table D.

E 0 0 AUDITMS

2489 IODEV PARAMETER EXPECTED

Model 204 finished processing User 0's parameter line and expected to find an IODEV parameter on the next line in the CCAIN input stream but did not find one.

System manager response: Make sure that an IODEV parameter is specified on the first line for User 1. Check to make sure that the CCAIN stream does not contain any other errors that could cause Model 204 to process User 1's parameters prematurely, for example, missed continuation in User 0's parameter line.

E 0 4 AUDITER

2500 SQL ERROR error-code error-message

SQL processing failed and an SQL error message was issued. The SQL message number and error message are identical to what is returned to the SQL client side and describe the actual error condition.

Response: Check in the *Model 204 SQL Error Codes Messages* or another SQL error message reference for details. Correct the SQL request accordingly and resubmit the request when the problem is fixed. For a file-related error notify the file manager. For a system-related error notify the system manager.

File manager response: None, unless the reported SQL message is related to file management issues.

System manager response: None, unless the reported SQL message is related to system management issues.

E 0 4 AUDITER

2501 RELEASE INCOMPATIBILITY

Version 5.1 supports ROLL FORWARD recovery for only Version 5.1 journals. The RESTART stream that was specified indicates the recovered run was from a pre-Version 5.1 release.

File manager response: Do one of the following:

1. Rerun ROLL BACK/ROLL FORWARD using a pre-Version 5.1 release.
2. Run only ROLL BACK under Version 5.1.

E 0 4 AUDITMS

2502 DFSHSM RECALL ERROR, DSNAME = dsname, RETURN CODE = rc, REASON CODE = rs

An ALLOCATE command attempted to dynamically allocate a dataset that had been migrated by DFSMSHSM, IBM's Storage Manager. Model 204 issued an HSM request to recall the dataset, but the request failed. The return and reason codes can be used to identify the cause of the failure.

A return code of 400, 401, or 806 indicates an ARCFMWE failure. This can be caused by a Model 204 internal error or an HSM error.

Any other return code indicates an ARCHRCAL failure. This indicates that HSM could not process the recall request. The meaning of return code *rc* and any associated reason code *rs* is described in message ARC11 *rc* which can be looked up in IBM manual *MVS System Messages Volume 1 (ABA - ASA) GC28-1784*.

Common return codes and their meaning are:

Return code	Meaning
3	Migration/Backup/Dump volume not available
8	Error cataloging dataset
51	Another DFSMSHSM function active for dataset, recall rejected
74	Dataset recall failed - The function is held

Operator response: Correct the problem with DFSMSHSM recall processing.

Response: Contact your system operator.

E 0 4 AUDITMS

2503 IBM SYSTEM INTERFACE MACRO macro ABENDED, ABEND CODE = code, REASON CODE = code

Model 204 has trapped an abend in the named IBM system interface macro. The abend code and abend reason code are displayed to help diagnose the cause of the abend.

System manager response: Contact CCA Customer Support.

E 0 4 AUDITMS

2509 errortype hexdump

Horizon uses this message to report two types of events:

- A receipt of request or a response unit for sessions that are in the process of termination and are therefore discarded.
- A failed attempt to BIND a session.

The general format of the message in the first case is

M204. 2509 *rutype* BEING DISCARDED FOR HS_ID = *xxxxxxxx*

or

M204. 2509 DISCARDING *rutype* FOR HS_ID = *xxxxxxxx*

where HS_ID is the hexadecimal Half Session identifier and *rutype* is one of the following:

- BIND RESPONSE
- CINIT (Control Initiate) REQUEST
- MESSAGE
- NOTIFY REQUEST
- TERMINATE REQUEST
- UNBIND RESPONSE

The general format of the message in the second case is:

M204. 2509 *rutype* FAILED | REJECTED SENSE=

where *rutype* is one of the following:

- BIND
- BIND RESPONSE

The sense codes are standard SNA codes which are listed in the IBM manual *SNA Formats, GA27-3136*.

The following table is a small selection of possible codes:

Code	Message	Check...
08050000	Session limit exceeded	DEFINE PROCESSGROUP INLIMIT parameter (or DEFINE SESSIONGROUP MAXIMUM and WINNERS parameters for CNOS).
08090000	Mode inconsistency (wrong state)	
080F6051	Security violation	Whether partner is opening the process with a valid user ID/password and check the LOGON and GUESTUSER parameters on the local and partner DEFINE PROCESSGROUP commands.
08150000	Function already active	
0835nnnn	Parameter error in BIND image	
084B0000	Resources are not available.	Storage to see if it is a problem.
08520001	Session activation request received while earlier one pending	

System manager response: The DISCARDED messages usually point to some earlier session failure, and no further action is usually necessary.

When there is a BIND failure, check the BIND parameters—the MODENAME parameter on the DEFINE PROCESSGROUP or DEFINE SESSIONGROUP—on both ends of the session. Refer also to the previous table on specific sense codes.

I 0 0 AUDITMS

2511 %C %C %C %C %C

This CCA debugging message is written only to the internal trace table. It is controlled by the DBGBIT parameter bit X'20000000'. If CCA Customer Support has not directed you to use this setting, then you will never see the message.

E 0 4 AUDITER

2512 ROLL BACK WILL USE THE FOLLOWING DATASET: %C

This message is informational only. It indicates that the Roll Back process will use the listed dataset for preimage processing. The values of %C can be RESTART or CHKPOINT.

I 0 0 OPR

2513 NON-ORDERED FIELD FOR VALUE IN CLAUSE

The field name mentioned in the VALUE IN clause must be Ordered.

E 0 4 AUDITER

2514 REMOTE VALUE IN CLAUSE NOT SUPPORTED

The VALUE IN clause is not supported in REMOTE context.

E 0 4 AUDITER

**2515 %C MPLOCK %C %C %C REGPAIR=%C USAGE=%C WORKREG=%C
COND=%C BUG=%C TRACE=%C**

This message is an internal Model 204 trace message meant for CCA Customer Support usage. Currently, this message is written only to the internal in-core-wrap-around trace table.

E 0 4 AUDITER

2516 MQ/204 OPTION ERROR: options REQUIRE options

On an MQ/204 statement, some options require that other options be specified.

Response: Correct the options on the statement in error.

E 0 4 ECHO AUDITER

2523 THE VALUE OF option CANNOT BE greater/less/longer THAN value

The value of an option on an MQ/204 statement or command, or an External Call Facility (ECF) statement, is invalid for the reason listed.

Response: Correct the specification of the option value.

E 0 4 ECHO AUDITER

2524 *value* IS AN INVALID *name*

On an MQ/204 statement or command, or an External Call Facility (ECF) statement, the value specified is invalid for the option identified by name.

Response: Correct the statement or command.

E 0 4 ECHO AUDITER

2527 option HAS ALREADY BEEN SPECIFIED

On an MQ/204 statement or command, or an External Call Facility (ECF) statement, the option listed has been specified more than once.

Response: Correct the statement or command.

E 0 4 ECHO AUDITER

2528 option IS AN INVALID OPTION

On an MQ/204 statement or command, or an External Call Facility (ECF) statement, an unrecognized option has been specified.

Response: Correct the statement or command.

E 0 4 ECHO AUDITER

2529 INVALID RUN-TIME OPTIONS VARIABLE

This message is issued when compiling certain MQ/204 statements, and is caused by one of the following conditions:

- A run-time options variable specified was not a string variable
- A run-time options variable specified was an array variable
- More than three run-time option variables were specified on the statement

User response: Correct the statement and retry the request.

E 0 4 ECHO AUDITER

2530 RESERVED FOR MQ/204 USE

This message will be used in an upcoming MQ/204 project.

E 0 4 ECHO AUDITER

2531 MQ/204 EVAL DEBUG: %C

This message can be issued when debugging MQ/204.

I 0 0 AUDITMS

2532 MQ/204 EVAL DEBUG END

This message can be issued when debugging MQ/204.

I 0 0 AUDITMS

2535 MQ/204 DISABLED (MQINTASK IS 0)

The current request has attempted to execute an CLOSE QUEUE, MODIFY QUEUE, MQBACK, MQCMIT, MQGET, MQPUT, MQPUT1, or OPEN QUEUE User Language statement or a START QUEUEMANAGER command has been issued, but MQ/204 is currently disabled. The MQINTASK parameter is set to zero and, therefore, no subtasks are available for communication with MQSeries.

Response: Contact your system manager.

System manager response: At least one MQ/204 subtask must be attached by an Online or Batch job which will use MQ/204. Review the MQ/204 requirements of the Online or Batch job and set the MQINTASK and MQMXTASK parameters accordingly.

E 0 4 CANCEL AUDITER

2536 MQ/204 OPERATIONS NOT ALLOWED DURING RECOVERY

The current request attempted to execute a CLOSE QUEUE, MODIFY QUEUE, MQBACK, MQCMIT, MQGET, MQPUT, MQPUT1, or OPEN QUEUE User Language statement or a START QUEUEMANAGER command was issued while ROLL FORWARD recovery was still in progress.

Response: Wait for recovery to complete and try again.

E 0 4 CANCEL AUDITER

2542 NO MQ/204 SUBTASKS AVAILABLE

A START QUEUEMANAGER command was issued, but no MQ/204 subtasks were available to verify that a connection to the queue manager could be established.

System manager response: Wait for users of MQ/204 to free up existing subtasks and try again. If necessary, increase the value of the MQMXTASK parameter or the MQINTASK parameter, or both.

E 0 4 AUDITER

2553 ECMSUBS LESS THAN ECISUBS; ECMSUBS RESET

The value of the ECMSUBS parameter (maximum number of ECF subtasks) was less than the value of the ECISUBS (initial number of ECF subtasks). The value of ECMSUBS has been reset to the value of ECISUBS.

This message can also indicate that a nonzero value was specified for ECISUBS, but ECMSUBS was allowed to default to zero.

System manager response: Review your system's ECF subtask requirements and reset ECISUBS and ECMSUBS accordingly.

E 0 4 AUDITMS

2555 ECF SUBTASK TYPE=type ECTASK=addr TCB=addr INITIALIZED

This informational message reports that an External Call Facility (ECF) subtask was initialized. The type is one of the following values:

Value	Subtask is used to...
0	Load or delete modules
1	Run external modules without subtask affinity
2	Run external modules with subtask affinity

I 0 0 SAVE VIEWERR AUDITRK NOTERM

2556 INSUFFICIENT STORAGE TO ALLOCATE control block name

The current request attempted to execute an External Call Facility (ECF) statement, but there was insufficient memory available to allocate the ECF control block named in the message. The request is canceled.

System manager response: Increase the value of the SPCORE parameter.

E 0 4 CANCEL AUDITER

2557 routine action name1=value1 name2=value2 name3=value3...

This informational message reports that an ECF low-level routine took the specified action with the values indicated.

I 0 0 AUDITRK NOTERM

2558 ECF function name1=value1 name2=value2

This informational message reports that an External Call Facility (ECF) statement was executed with the values indicated.

I 0 0 AUDITRK NOTERM

2560 ECF SUBTASK ECTASK=addr TCB=addr (MODULE=name) TERMINATED DUE TO cause

An External Call Facility (ECF) subtask was terminated while running the external module named in the message. The cause can be either:

- User bump
- Previous ABEND
- Operating system request

Response: Determine what action needs to be taken as a result of the abnormal termination of the named module.

I 0 0 AUDITRK NOTERM OPR

2561 ECF operation MODULE=name COMPLETION=code REASON=code RETURN=code ECTASK=addr TCB=addr

An External Call Facility (ECF) operation failed when processing the external module named in the message.

- *Operation* can be one of:

<i>Operation</i>	You issued ECF statement	OS/390 or z/OS responded by...
CALL	EXTERNAL CALL	ABEND external program.
DELETE	EXTERNAL DELETE	Issuing an error to DELETE request.
LOAD ABENDED	EXTERNAL LOAD	ABEND during processing.
LOAD FAILED	EXTERNAL LOAD	Issuing an error to LOAD request.

- *Completion code* has two formats:

Completion code	Type of code is...
<i>Snnn</i>	OS/390 or z/OS system code, described in <i>IBM MVS System Codes, GC28-1780</i> .
<i>Unnnn</i>	User ABEND code, issued by the external program that ABENDED.

The Completion code is a hexadecimal value for system codes and a decimal value for user ABEND codes.

- *Reason code* is a hexadecimal value.
- *Return code* is a decimal value.

The message always reports *either* a Return code *or* a Completion code and associated Reason code.

Response: Determine what action needs to be taken, based on the error code.

I 0 0 SAVE VIEWERR AUDITRK NOTERM

**2562 ECF operation MODULE=name DDNAME=name ADDR=addr
LENGTH=value EPA=addr**

An External Call Facility (ECF) external module was loaded or deleted. The following information about the program is reported:

Information	Specifies
<i>operation</i>	LOADED or DELETED
ADDR	Address in storage at which the module was loaded
EPA	Entry point address of the module
LENGTH	Length, in bytes, of the module

I 0 0 AUDITRK NOTERM

2563 MODULE=name RETURNED MORE THAN length BYTES

An EXTERNAL CALL statement passed a parameter area of *length* bytes to an external module. However, when the module executed, it modified storage that followed the copy of the parameter. After storage corruption has occurred, the user is restarted.

Response: Check that the definition of the parameter area in the User Language program, the image definition, matches the definition in the external module. For COBOL, check the LINKAGE SECTION.

E 0 4 AUDITMS

2565 filename=yy.ddd hh:mm:ss.tt, RESTART=yy.ddd hh:mm:ss.tt

A file can be rolled back only when using the checkpoint dataset (RESTART) from the last job that updated the file. This message, which follows M204. 0146: ROLL BACK INFORMATION IS OBSOLETE, is issued when the RESTART dataset is *not* from the last job that updated the file.

There are two possible causes for this error:

- The file was updated by another job after the run that you are trying to roll back.
- The RESTART [DD | DLBL | FILEDEF] command points to the wrong checkpoint dataset.

Comparing timestamps

The timestamps in this message may assist in identifying the reason for the ROLL BACK INFORMATION IS OBSOLETE message. Model 204 creates a file-specific timestamp at the *first* update to a file after *each* checkpoint. If a job does not use checkpointing, the timestamp is created at the first update of the job only.

The timestamp is written to both the checkpoint dataset, if checkpointing is active, and the file itself. During ROLL BACK processing, Model 204 checks that the most recent timestamp from the RESTART dataset is the same as that in the file. If they are different, message M204.0146 and this message are issued.

This is NOT a Model 204 internal error.

If the timestamp from the file is:

- Later than the timestamp from the RESTART dataset, then the file was updated by another job after the run that you are trying to roll back. The timestamp from the file may assist you to identify the other job. Remember that the timestamp represents the time of the first update after the last checkpoint in that job, or the first update of the job, if not checkpointing.
- Earlier than the timestamp from the RESTART dataset, then either you are using the wrong checkpoint dataset for RESTART; or—less likely—you restored the file from an old backup before attempting to run recovery.

System manager response: Correct the error and rerun Model 204 recovery.

Operator response: Contact your system manager.

E 0 4 AUDITMS

2567 REGS=%X %X %X %X %X %X %X %X %X %X %X %X %X %X %X

This message is intended for CCA Customer Support. The hexadecimal values of the current general registers are displayed.

E 0 4 AUDITMS

2568 ENTRY%C, PREV RL%C, 2ND PREV RL%C

This message is intended for CCA Customer Support.

The MODULE NAME and OFFSET of the current routine and the MODULE NAME and OFFSET of the routine that called it are displayed.

E 0 4 AUDITMS

2569 SESLOCK session lock number ALREADY LOCKED|UNLOCKED

A Horizon session lock has been obtained or released twice. This is a serious internal error.

User response: Notify your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

I 0 0 AUDITAD

2570 TCP/IP CLEANUP DEFERRED FOR SOCKET %C USER %C - %C

This is an informational message.

At the time of logout, a user owned a TCP/IP connection that was not cleaned up by the user thread because of one of the following preconditions:

- A local SESSTERM was already in progress.
- A SESSTERM request had already been received from the remote partner.
- The connection had already been terminated by the remote partner.
- A previous error on this connection had already triggered cleanup.

I 0 0 AUDITAD

2571 TCP/IP CLEANUP DEFERRED FOR SOCKET %C USER %C - LOCAL SESSTERM HAS BEEN PREVIOUSLY SCHEDULED

This is an informational message.

At the time of logout, a user owned a TCP/IP connection. The connection was not cleaned up by the user thread because a local SESSTERM had already been scheduled for this thread and will be performed by the TCP/IP Service PST.

I 0 0 AUDITAD

2572 TCP/IP CLEANUP FAILED FOR SOCKET %C USER %C - SESSION BLOCK COULD NOT BE LOCATED

This is a warning message that suggests an underlying problem in the cleanup of the user's connection at the time of logout.

User response: Collect the audit trail and the User Language procedures, before contacting CCA Customer Support.

I 0 0 AUDITAD

2573 function USER=user-ID COMP_CODE=completion code RSN_CODE=reason code QM=queue manager Q=queue EXT_QM=external queue manager name EXT_Q=external queue name LOCAL DYNAMIC QUEUE action

This informational message reports the results of an MQSeries API call that creates or deletes an MQSeries Local Permanent or Temporary Dynamic Queue.

Information	Represents
function	MQSeries API call (MQOPEN, MQCLOSE)
user-ID	User for whom the call was performed

Information	Represents
completion code	Completion code returned by the API call
reason code	Reason code returned by the API call
queue manager	Queue manager entity on which the call operated
queue	Queue entity, if any, on which the call operated
action	OPENED, CREATED, CLOSED WITH DELETE, or DELETE_PURGE

The external queue manager and queue names for the associated queue are the names by which the queue and queue manager are defined to MQSeries. They might not be the same as the names by which they are defined to Model 204.

E 0 4 AUDITMS

2574 THIS OPERATION IS NOT SUPPORTED FROM USER ZERO

An illegal operation, such as SNA Communications Server transfer control, has been attempted from User 0 (from a command in the CCAIN stream).

User response: Alter the CCAIN stream to remove the illegal command.

E 0 4 AUDITMS

2575 RENAME REJECTED, filename MUST BE STOPPED

Access to *filename* must be temporarily disabled by the system manager prior to the RENAME command. Users who already have the file open will not be affected by the stop, but no new opens will be allowed.

Response: Contact your system manager.

RENAME REJECTED, filename IS MEMBER OF GROUP groupname

The renamed file must not be a member of any open group.

Response: Close all appropriate open groups and redefine them.

RENAME REJECTED, filename CREATED WITH FRCVOPT=256

The renamed file must not be specified with FRCVOPT=256. A file rename creates a discontinuity and FRCVOPT disallows them.

Response: Reset FRCVOPT to remove the disallowed discontinuity option.

RENAME REJECTED, filename ACTIVE PROCEDURE INSIDE RENAMED FILE

The RENAME command must not be issued from an active procedure that is included in the file being renamed.

Response: Remove the procedure from the file being renamed and re-execute.

E 0 4 AUDITER

2576 FILE RENAME FROM filename1 TO filename2

A successful RENAME command was issued. You can no longer access the file using the original name, *filename1*. You must issue an OPEN FILE command to access the new file, *filename2*.

I 0 0 AUDITMS

2577 SUBSYSTEM NAME EXCEEDS 10 CHARACTERS IN LENGTH

The user specified a subsystem name that is longer than 10 characters. A subsystem name must be 10 characters or fewer.

User response: Check the subsystem name and try again.

E 0 4 AUDITMS

2578 SOCKET socket_number LINK link_name CLOSING

The named socket is in the process of closing. The name of its owning LINK is noted. This is an informational message.

I 0 0 AUDITRK NOTERM

2579 SOCKET socket_number LINK link_name CLOSED

The named socket has been closed. The name of its owning LINK is noted. This is an informational message.

I 0 0 AUDITRK NOTERM

2580 type FAILED R15=value R0=value

One of the following requests dealing with dataspace services failed:

- ADRSPACE CREATE
- ALESERV ADD
- ALSERV ADD
- DSPSERV CREATE

The values of register 15, the return code, and register 0, the reason code, are specified.

Response: Look up the return code and reason code to determine if a system parameter needs to be adjusted or if CCA Customer Support needs to be contacted.

If the messages says ALESERV or DSPSERV, the IBM manual to look in is: OS/390

MVS Programming: Authorized
Assembler Services Reference, Volume 1
(ALESERV-DYNALLOC)

Document Number GC28-1764

If the message says ADRSPACE or ALSERV, the IBM manual to look in is: VM/ESA.

CP Programming services

Document Number SC24-57-60

E 0 4 CANCEL AUDITER

2581 XMEMOPT=2 (IOS BRANCH) REQUIRED FOR XTLOT OPTION

The dynamic allocation option XTLOT may be specified only if Model 204 is running with IOS branch entry (CCAIN parameter: XMEMOPT=X'02').

Response: Either remove the XTLOT option or contact the system manager.

System manager response: Either remove the XTLOT option or restart Model 204 with XMEMOPT=X'02' and the appropriate SVC entry.

E 0 4 AUDITMS

2582 ERROR CREATING DATASPACE spacename -

{ NOT ON XC MODE VIRTUAL MACHINE
| CAN'T CREATE HIPERSPACE UNDER CMS
| NOT APF AUTHORIZED TO CREATE HIPERSPACE
| DATASPACE ALREADY EXISTS WITH SAME NAME }

System manager response: Correct the problem and rerun the job

E 0 4 AUDITMS

2583 PCBPTTRD NOT ZERO FOLLOWING PASSIVE SOCKET CLOSE

An internal error has occurred in the Horizon for TCP/IP thread management.

Response: Contact your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

E 0 4 SNAP AUDITER NOTERM OPR

2584 SOCD NOT FOUND

A socket control block has been unexpectedly deallocated. This is a serious internal error.

Response: Contact your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

E 0 4 SNAP AUDITER NOTERM OPR

2585 CLOSE ATTEMPTED FOR SOCKET WITHOUT SESD

A CLOSE_SOCKET request was attempted on a socket connection not recognized by the Horizon session manager. This is a serious internal error.

Response: Contact your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

E 0 4 SNAP AUDITER NOTERM OPR

2586 CLOSE ATTEMPTED AFTER API TERMINATION

A CLOSE_SOCKET request was attempted on a socket after the Horizon for TCP/IP had terminated. This is a serious internal error.

Response: Contact your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

E 0 4 SNAP AUDITER NOTERM OPR

2587 SOCKET socket-number WAS SET IN THE EXCEPTION FDS FOR link-name BUT THERE IS NO SESSION BLOCK CURRENTLY ASSOCIATED WITH THIS SOCKET

The Horizon for TCP/IP interface has received notification that a socket incurred an exception, but the Horizon session manager does not recognize this socket connection as extant. This may be the result of an application error, a temporary increase in system load, or an internal error in the Horizon for TCP/IP interface or in TCP/IP itself.

Response: Contact your system manager.

System manager response: Contact CCA Customer Support for further instructions.

I 0 0 AUDITAD

2588 SOCKET socket-number WAS SET IN THE READ FDS FOR link-name BUT THERE IS NO SESSION BLOCK CURRENTLY ASSOCIATED WITH THIS SOCKET

The Horizon for TCP/IP interface has received notification that a socket is ready for reading, but the Horizon session manager does not recognize this socket connection as extant. This may be the result of an application error, a temporary increase in system load, or an internal error in the Horizon for TCP/IP interface or in TCP/IP itself.

Response: Contact your system manager.

System manager response: Contact CCA Customer Support for further instructions.

I 0 0 AUDITAD

2589 SOCKET socket-number WAS SET IN THE WRITE FDS FOR link-name BUT THERE IS NO SESSION BLOCK CURRENTLY ASSOCIATED WITH THIS SOCKET

The Horizon for TCP/IP interface has received notification that a socket is ready to accept data, but the Horizon session manager does not recognize this socket connection as extant. This may be the result of an application error, a temporary increase in system load, or an internal error in the Horizon for TCP/IP interface, or in TCP/IP itself.

Response: Contact your system manager.

System manager response: Contact CCA Customer Support for further instructions.

I 0 0 AUDITAD

2590 SESD FOR SOCKET socket-number LINK link-name UNEXPECTEDLY DEALLOCATED BEFORE SOCKET DATA COULD BE READ

The Horizon for TCP/IP interface has received notification that a socket is ready for reading, but the Horizon session manager does not recognize this socket connection as extant. This may be the result of an application error, a temporary increase in system load, or an internal error in the Horizon for TCP/IP interface or in TCP/IP itself.

Response: Contact your system manager.

System manager response: Contact CCA Customer Support for further instructions.

I 0 0 AUDITAD

2591 SESD FOR SOCKET socket-number LINK link-name UNEXPECTEDLY DEALLOCATED AFTER SOCKET DATA WAS READ

The Horizon for TCP/IP interface has read data for a socket connection, but the Horizon session manager does not recognize this socket connection as extant. This may be the result of an application error, a temporary increase in system load, or an internal error in the Horizon for TCP/IP interface or in TCP/IP itself.

Response: Contact your system manager.

System manager response: Contact CCA Customer Support for further instructions.

I 0 0 AUDITAD

2592 PASSIVE SOCKET FOR LINK link-name HAS BEEN CLOSED

This message documents that the socket used to accept new connection requests for LINK *link-name* has been closed.

I 0 0 AUDITAD

2593 SESD FOR SOCKET socket-number WAS UNEXPECTEDLY REALLOCATED

A serious internal error has occurred in the Horizon session manager.

Response: Contact your system manager.

System manager response: Forward the resulting CCASNAP to CCA Customer Support.

E 0 4 SNAP AUDITER NOTERM OPR

2594 SOCKET socket-number RECEIVED A ZERO-LENGTH RECORD FROM REMOTE remote-id PORT port-number

The named remote partner has closed the named socket connection by sending a zero-length record.

I 0 0 AUDITAD

2595 PASSIVE SOCKET FOR LINK link-name HAS BEEN CLOSED TEMPORARILY DUE TO LOCAL RESOURCE SHORTAGE

The named link has reached its connection limit as specified by the CONNECTIONS parameter on the DEFINE LINK command. No new

connection requests can be accepted until an existing connection is deallocated.

Response: If this message occurs frequently, contact your system manager.

System manager response: If this message occurs frequently, increase the number of CONNECTIONS specified on the relevant DEFINE LINK.

I 0 0 AUDITAD

2596 PASSIVE SOCKET FOR LINK link-name HAS BEEN REOPENED: LOCAL RESOURCE SHORTAGE ABATED

The named link had reached its active connections limit and closed its passive socket. A connection active at that time has been deallocated. New connection requests can now be accepted.

I 0 0 AUDITAD

2598 SOCKET socket-number MOVED TO DEFERRED CLNUP QUEUE FOR LINK link-name

The named socket has been moved to the deferred cleanup queue of *link-name*. This is an informational message.

I 0 0 AUDITRK NOTERM

2599 TCP/IP PC CALL HANDLER "SNMPGPCN" NOT LINKED IN

Response: Relink the ONLINE module and include the SNMPGPCN deck from the IBM or INTERLINK library.

E 0 4 OPR

2600 SESSINIT REJECTED: SOCKET socket-number REMOTEID=remote-net-address PORT=remote-port LINK=local-link-name

A client's OPEN PROCESS request for a TCP/IP session initiation has been rejected by the remote peer.

Response: Retry the OPEN PROCESS request. If this fails, ensure that the remote peer is available to the network.

E 0 4 AUDITMS

2601 ECF MODULE NAME NOT status

A BUMP MODULE *module-name* command was entered, but the specification of the module name was invalid. *Status* is one of the following:

Status	Means the module name was not...
DEFINED	Yet defined. Module names are defined by the User Language EXTERNAL MODULE statement.
SPECIFIED	Identified
VALID	A valid module name. For example, the name is too long.

E 0 4 AUDITMS

2602 LATE DATA RECEIVED AFTER TIMEOUT OR BUMP -- RH=Req_header, RU=Req_unit

Horizon has received data from the session partner after a bump or timeout terminated the conversation prematurely.

- *Req_header* shows the 3-byte SNA Communications Server request header, in hexadecimal.
- *Req_unit* shows the first 16 bytes of the SNA Communications Server request unit, in hexadecimal.

This is a normal event that can be expected to occur when a user is bumped or a conversation times out. No action is required and Horizon discards the late data.

I 0 0 NOTERM OPR

2603 SOCKET ACCEPT REQUEST FOR LINK link-name COULD NOT BE ACCOMMODATED: NO SESSION BLOCK AVAILABLE

The Horizon for TCP/IP interface received notice of a socket connection request, but the Horizon session manager cannot allocate a session block for this request. The socket request has been accepted and closed. This error indicates too few connections were specified for the named Horizon for TCP/IP link.

Response: Contact your system manager.

System manager response: Increase the number of connections in the definition of the named link.

I 0 0 AUDITAD

2604 FLOD RECORD DEFINITION EXCEEDS RECORD LENGTH

The File Load record statements exceeded the length of the current record being processed. The File Load job is terminated and no further updates will be processed.

File Manager response: Correct the FLOD program before resubmitting the job.

E 64 64 AUDITER

2605 CHKPOINT TOO SMALL FOR ROLL FORWARD - number1 BLOCKS REQUIRED; number2 FOUND

This message is written before ROLL BACK processing and indicates that the CHKPOINT dataset is too small to successfully complete ROLL FORWARD processing. Model 204 will be immediately terminated.

System manager response: Increase the size of the CHKPOINT dataset and rerun Model 204 using the same job from the aborted recovery run.

Operator response: Notify the system manager of this problem.

E 52 52 AUDITAD OPR

2606 ONLY number IODEV numbers ARE LICENSED, RUN TERMINATED

The number of specified IODEVs has exceeded the number that your contract allows.

System Manager response: Either reduce the quantity of the specified IODEVs or contact CCA Customer Support to arrange for a greater supply.

E 0 4 AUDITMS

2607 CPQZACTN AND CPQZSECS ARE COREQUISITE PARAMETERS

System manager response: The two CCAIN User 0 parameters CPQZSECS and CPQZACTN must both be specified, if either is specified.

You will also get this message if both parameters are specified, but CPQZACTN is zero (0) and both RCVOPT bit X'01' is on and the NUSERS parameter is greater than one (1). Make appropriate changes to the parameters and resubmit the job.

CPQZSECS determines how many seconds to wait after the checkpoint extended quiesce period has started before the CPQZACTN action is taken. If the CHECKPOINT END EXTENDED QUIESCE command is not entered within this time period, then the action specified by the CPQZACTN parameter setting is taken.

E 0 4 AUDITER

2608 CPQZACTN VALUE IS INVALID OR MORE THAN ONE MUTUALLY EXCLUSIVE VALUE SET

There are four possible actions to specify for the CHECKPOINT EXTENDED QUIESCE timeout parameter; they are:

Setting	Action
X'80'	CONTINUE—Wait another CPQZSECS
X'40'	ENDEQ—Release all updating users, end quiesce
X'20'	OPERATOR—Ask Operator what to do
X'10'	EOJ—Terminate the run

The setting for the CPQZACTN parameter did not specify either a valid value or it specified more than one valid value.

System manager response: Change the setting of CPQZACTN and retry the job, if CPQZACTN was specified on User 0's parameter line. Or, if a RESET command was issued, reissue with a valid value.

E 0 4 AUDITER

2609 POSTING NOT ALLOWED FOR NAMED ECBS

The application attempted to post a named ECB, such as CPQZ or QZSIG. Named ECBs are posted by Model 204 internal routines only.

E 0 4 AUDITER

2610 SYSTEM ENTERED EXTENDED QUIESCE AT: yy/ddd hh:mm:ss, ALL FILE UPDATING REMAINS SUSPENDED

or

-- END EXTENDED QUIESCE PROCESSING IS DELAYED WITH nnnn USER(S) IN WAIT TYPE 47

or

-- END EXTENDED QUIESCE PROCESSING IS DELAYED WITH nnnn USER(S) IN WAIT TYPE 48}]

A Model 204 system that is in extended quiesce, presumably while file backups are being done, sends this informational message to the system operator every minute. The options indicate why a CHECKPOINT END EXTENDED QUIESCE command is not immediate.

I 0 0 NOTERM OPR

2611 CHECKPOINT [UNSET | SET | END] COMMAND SUCCESSFUL

Audited message indicating the successful completion of the stated command.

I 0 0 AUDITAD

**2612 CHECKPOINT COMMAND UNSUCCESSFUL -
{CHECKPOINTING NOT SPECIFIED FOR THIS RUN
| CCAIN PARM NUSERS =1
| IN SYSTEM INITIALIZATION, TERMINATION
| INVALID TRANSITION
| REDUNDANT TRANSITION SPECIFIED}**

The reason the previously issued command didn't complete successfully.

System manager response: Reissue in the appropriate context.

Operator response: Reissue in the appropriate context.

E 0 4 AUDITER

**2613 SYSTEM ENTERED EXTENDED QUIESCE AT: yy.ddd hh:mm:ss, SYSTEM
EXITED EXTENDED QUIESCE AT: yy.ddd hh:mm:ss, REASON = {END
COMMAND ISSUED BY USER: nnnnnn
| EOJ COMMAND
| CPQZACTN TIME OUT ACTION
| CHKPPST RESTART}**

Message sent to operator's console indicating termination of extended quiesce.

I 0 0 NOTERM OPR

**2614 EXTENDED QUIESCE IN PROGRESS -UPDATING SUSPENDED, PLEASE
WAIT**

Information message sent to a user who tried to issue a file updating command or start a user language updating request during extended quiesce; they are in a swappable wait during extended quiesce.

I 0 0 AUDITMS

2615 %C

Message printed without prefix and appended to output of the CHKMSG or MONITOR CHECKPOINT commands to display status of extended quiesce processing.

I 0 0 NOPREFIX AUDITAD

2616 %C COMMAND IGNORED - SYSTEM IN EXTENDED QUIESCE

Some form of the CHECKPOINT or CHECKPOINT ABORT command was issued while the system was in extended quiesce. This is an information message.

E 0 4 AUDITMS

2617 EXTENDED QUIESCE RING JOURNAL OFFLOAD FAILED

Ring Journaling with an offload stream is defined to the system, and at the beginning of extended quiesce processing the automatic offload failed; more information may be available in a prior message.

Operator response: Notify appropriate personnel.

E 0 4 AUDITER OPR

2618 USER ENTERED A NONSWAPPABLE SWAIT DURING EXTENDED QUIESCE PROCESSING -KJWHAT= X%X'

The user issued a file command and as a result went into a non swappable wait during extended quiesce processing.

System manager response: The User Language or IFAM process that displayed this message will continue with normal operation. No response is needed except to collect the displayed information if requested by CCA Customer Support.

E 0 4 AUDITER NOTERM

2619 USER IN \$WAIT('QZSIG') SWAIT RESTARTED - EXTENDED QUIESCE PROCESSING MAY BE AFFECTED

or

USER IN \$WAIT('CPQZ') SWAIT RESTARTED - EXTENDED QUIESCE PROCESSING MAY BE AFFECTED

or

CHKPPST PST RESTARTED - EXTENDED QUIESCE PROCESSING MAY BE AFFECTED

There is a problem in extended quiesce processing system. Message is audited as an AD line to prevent Online from coming down.

Operator response: Notify appropriate personnel.

E 0 4 AUDITAD OPR

2620 SUBSYSTEM name INVALID CCATEMP PAGE COUNT count

At the end of STOP SUBSYSTEM command processing, a check is made that all CCATEMP pages used for saved compilations were freed. This message is issued if it is determined that some pages were not freed. In normal processing, you do not see this message, but it could occur if, for example, a user was restarted during a previous attempt to stop the subsystem. The orphaned CCATEMP pages cannot be recovered.

System manager response: No response is required. However, if you see this message frequently, the cause should be investigated.

E 8 8 VIEWERR

2621 NO FILE TABLES DUMP(G)ED

A DUMP or DUMPG command was issued and, at termination, the highest table number dumped was 0.

File manager response: If a DUMP command was issued then the original file is corrupt and the backup created by the DUMP command invalid. Call Customer Support at CCA with documentation.

System manager response: If a DUMPG command was issued then CCAGRP is corrupt and the backup created by the DUMPG command invalid. Call Customer Support at CCA with documentation.

Operator response: Notify the file manager or system manager.

E 20 20 AUDITER

2622 HIGHEST CCATEMP PAGE # USED IN area = page number, HIGHEST AVAILBLE = page number

Model 204 is terminating and reports the highest number of CCATEMP pages used from each of the two CCATEMP areas:

- SMALL MODEL AREA - the first 64K (65,536) pages of CCATEMP
- EXPANSION AREA - the remaining pages in CCATEMP

This informational message is written during Model 204 termination.

I 0 0 AUDITAD NOTERM

2625 THE PRECEDING MESSAGE CAUSED A RESTART DUE TO REQUIRED WAIT AND OBTAINED RESOURCES

The preceding message was issued while user had open pages and terminal screen had no place to print the message. The required wait for user to clear the screen is not allowed while having open pages.

Response: Notify the system manager.

System manager response: The preceding message is the real source of the problem. Analyze the message and respond accordingly

Operator response: Inform the system manager.

E 0 4 VIEWER

2626 XMEMSVC MUST BE SET TO A VALID SVC # IF X'02' IS SET IN XMEMOPT

If the XMEMOPT parameter is set with X'02', which invokes IOS BRANCH ENTRY and M204XSVC is not linked into the ONLINE or BATCH204 load module, then XMEMSVC must be set to a valid SVC number.

System manager's response: Add the XMEMSVC setting on User 0's parameter line.

E 72 72 AUDITER

2628 uuuuuuuuuu aaaaaaaaaa LOGOUT yy mmm dd hh.mm

This message is sent to a user's terminal and the audit trail for a successful logout; if the CUSTOM parameter is set to option 6. If the CUSTOM=6 option is not set, the LOGOUT message number is 0353.

The field values for the message are:

- *uuuuuuuuuu* is the user's user ID.
- *aaaaaaaaaa* is the user's account.
- *yy mmm dd* is the current Gregorian date (for example, 02 MAR 31).
- *hh.mm* is the current time in hours and minutes.

I 0 0 AUDITAD NOTERM

New Model 204 unnumbered messages

EXTENDED QUIESCE WILL OFFLOAD CURRENT RING JOURNAL

This information message is displayed whenever a journal offload is done at the beginning of an extended quiesce. This message follows message 0843.

New SQL messages

- 2079 Invalid Inner Join specifications.**
- 2080 Invalid Cross Join specifications.**
- 2081 Invalid Natural Join specifications.**
- 2082 Invalid Left (outer) Join specifications.**
- 2083 Invalid Right (outer) Join specifications.**
- 2084 Invalid Full (outer) Join specifications.**
- 2085 Unknown Join operation.**
- 2086 Invalid join syntax.**
- 2087 Invalid scope of join qualifier(s).**
- 2088 Multiple parent tables in join qualifier(s).**
- 2089 A nested table % in an outer join is not supported.**
- 4016 Table reference to a label in a subquery is lost.**
- 4017 Nested table reference to a label in a subquery is lost**
- 4703 Rule of mapping FOREIGN KEY to MODEL 204 field % violated.**
- 5541 Column % maps to a Model 204 invisible field and may not be used in an ORDER BY clause.**
- 6315 SYSNAME cannot be specified with PRIMARY KEY SYSTEM.**

Index

Symbols

`$CENQCT` function
 monitoring the Resource Enqueue Table 1-58

`$CHKPAT` function
 changes to pattern matcher 2-9
 verifying syntax of a pattern 4-12

`$CHKPINF` function 1-39
 tracking checkpoints 1-39

`$ECBDGET` function
 and NECBS parameter 1-41
 syntax for 1-41

`$ECBDSET` function
 and NECBS parameter 1-42
 associating a string 1-15
 comparing to `$POST` 1-15
 syntax for 1-42

`$ECBTEST` function
 syntax for 1-43

`$ECFSTAT` function
 syntax for 3-12

`$ERRCLR` function
 clearing message text 1-44

`$ERRMSG` function
 coordinating with `$ERRCLR` 1-44

`$FSTERR` function
 coordinating with `$ERRCLR` 1-44

`$INCRG` function
 and `$DELG` function 4-9

`$POST` function
 associating a string 1-15
 comparing to `$ECBDSET` 1-15
 syntax for 1-45

`$POST('QZSIG')` function call
 limitation 1-22

`$SETG` function
 and `$DELG` function 4-9

`$STATUS` return codes
 ECF 3-11

`$STATUSD` return codes
 ECF 3-11

`$STATUSR` function
 continuing DO WHILE loops 1-47
 syntax 1-47

`$UNPOST` function

 syntax for 1-47

`$VALIDATE_NUMERIC_DATA` function 1-48
 detecting invalid numeric data 1-48

`$WAIT` function
 caution with NOSWAP option 1-50
 syntax for 1-49

Numerics

31-bit BATCH2 processing
 under VM 2-20

64-bit z/Architecture
 and Model 204 2-17

A

Advantage/SQL
 INS204 task parameter 2-18

aliases
 in SELECT LIST statement 1-66

ALLOCATE command
 obsolete form 1-32
 replacing dynamic allocation in RESTART
 processing 1-12
 with XTLOT or TLOT 1-31

APARs required
 eliminate IBM TCP/IP problems 1-72

applications
 rewriting for VALUE IN clause processing 1-34

APSY
 load statistics 1-29
 resident load threshold 1-90

APSY precompiled procedures
 APSYPAGE parameter 1-76
 in Storage feature 1-28

APSY Precompiled Procedures in Storage
 DSOPT parameter 1-29

APSYLD statistic
 tracking CCASYS procedures 1-29

APSYLDD statistic
 tracking dataspace loads 1-29

APSYLDT statistic
 tracking tiny loads 1-30

APSYPAGE parameter

- activating APSY Precompiled Procedures in Storage 1-28
- compared to TEMPPAGE 1-28
- pages for precompiled procedures 1-76
- arrays, %variable
 - index loop with 4-7
- asterisk (*)
 - wildcard in SELECT list statement 1-66
- AUDIT204 utility
 - identifying users 2-22
- AUDITMS option
 - for MSGCTL command 1-104
- AUDITOPT parameter
 - audit MQ/204 messages 1-76

B

- backpages
 - accurately counting 2-7
- BATCH2
 - backup requirements 1-22
 - IODEV required for backup 1-22
- BIND ERRNO 48
 - IBM TIME-WAIT state 1-75
- Block Comments
 - support for in XREF 1-56
- BUMP command
 - and login delays 4-5
- BUMP MODULE command
 - logging out of an external module 3-4
- BY phrase, in index loops 4-7

C

- cache hiperspace
 - DSPOPT parameter 1-29
- call name
 - associating an external program 3-7
 - definition of 3-2
- CCA Customer Support
 - evaluating your CCAPRINT and CCAAUDIT files 2-3
- CCA Web site
 - accessing most recent Model 204 updates 1-4
 - address 1-4
 - downloading zaps 1-5
 - Early Warnings 1-4
 - obtaining user name and password 1-4
- CCAAUDIT
 - messages location offset 2-11
- CCACAT file
 - field attribute change 2-3
- CCAJRNL

- message location offset 2-11
- CCAJRNL file
 - entries 4-5
 - recovery 4-5
 - user statistics 1-110
 - since-last 1-110
- CCAPRINT file
 - displaying Early Warning information 2-3
- CCARF dataset 1-11
 - NOTE/POINT facility 1-11
 - ROLL FORWARD processing 1-10
- CCASERVR in Storage
 - activating 1-25
 - DESPOPT parameter 1-29
 - managing server swapping 1-25
 - system programmers consideration 1-27
- CCASYS procedures
 - tracking with APSYLD 1-29
- CCATEMP in Storage
 - activating 1-25
 - managing server swapping 1-25
 - system programmers consideration 1-27
- CCATEMP statistics
 - expansion area pages 1-92
 - for small page area 1-93
 - highest page count 1-94
 - high-water mark 1-92, 1-93
 - small page area 1-93
 - small page area highest count 1-93
- CFR
 - monitoring 1-98
 - See also* critical file resource
- CFRLOOK parameter
 - MONITOR command 1-99
- channel programs
 - setting with NJBUFF 1-89
- CHECKPOINT command
 - handling a checkpoint 1-97
 - syntax for 1-97
- CHECKPOINT END EXTENDED QUIESCE
 - command
 - terminating extended quiesce state 1-19
- checkpoint extended quiesce
 - introducing 1-19
- CHECKPOINT SET EXTENDED QUIESCE
 - command
 - beginning extended quiesce 1-19
- CHECKPOINT UNSET EXTENDED QUIESCE
 - command
 - reversing the SET option 1-19
- checkpoints
 - finding quickly 1-11
 - monitoring 1-101

- tracking with \$CHKPINF 1-39
- verifying 1-39
- CHKABORT command
 - ending transactions 2-7
- CHKMSG command
 - ending a transaction 2-7
 - returning checkpoint status 1-98, 2-13
- CHKPOINT dataset
 - enhanced ROLL FORWARD processing 1-60
 - required for recovery 1-9
 - required for ROLL BACK processing 1-9
- CICS Interface
 - updated Transaction Server 2-19
- CICS Transaction Server
 - and CICS Interface 2-19
 - VSE operating system 2-22
- CLEAR GLOBAL statement
 - clearing global found sets 4-11
- CLEAR GLOBALS statement
 - clearing objects from GTBL 4-10
- CLEARGO command 4-9, 4-11
- clients
 - defining a link to 1-73
- CNOS
 - incompatible with Horizon TCP/IP 1-74
- columns
 - matching in NATURAL JOIN feature 1-68
- COMEND parameter
 - specifying in XREF facility 1-56
- common variables
 - defining 1-33
 - programming example 1-33
- compatibility issues
 - accurate display of running users 2-13
 - backpaging count 2-7
 - CCACAT field attribute 2-3
 - CHKABORT command 2-7
 - CHKMSG command 2-7, 2-13
 - FIND statement 2-10
 - IF statement 2-10
 - increased server size 2-5
 - increasing LHEAP 2-5
 - increasing QTBL size 2-5
 - location offsets 2-11
 - mathematics \$functions 2-10
 - MONITOR SL command 2-11
 - more in CCAPRINT file 2-3
 - OS/390 and z/OS installation 1-24
 - pattern matcher 2-9
 - REDEFINE command and INVISIBLE fields 2-7
 - RESTART recovery 2-2
 - ROLL BACK messages 2-12
 - subsystems at initialization 2-3
 - wait types 2-13
- COMSTART parameter
 - specifying in XREF 1-56
- conditional join feature
 - using an ON clause 1-69
- Connect ★
 - free threads 1-24
 - INS204 task parameter 2-18
- correlation name feature
 - in SELECT LIST statement 1-66
- CPQZ
 - posting and unposting automatically 1-19
 - special ECB 1-41, 1-42
- CPQZACTN parameter
 - extended quiesce action 1-76
- CPQZSECS parameter
 - managing an extended quiesce 1-21
 - timing CPQZACTN 1-77
- CPU ID zap
 - installation kit requirements 1-4
- CRAM
 - reinstalling 2-17
 - required for BATCH2 backup 1-22
- CRAMINS job
 - required version 2-17
- CRIOCHNL
 - defined for BATCH2 backup 1-22
- critical file resources
 - monitoring 1-99
 - See also* CFR
- CROSS JOIN feature
 - no join condition specified 1-68
- Cross Reference Selection screen
 - in XREF facility 1-56
- CURRENT_DATE keyword
 - SELECT statement option 1-67
- CURRENT_TIME keyword
 - SELECT statement option 1-67
- CURRENT_TIMESTAMP keyword
 - SELECT statement option 1-67
- CUSTOM parameter
 - using customized parameters 1-78
- CUSTOM=6
 - managing the LOGOUT message 1-78
- CUSTOM=7
 - and MSGCTL command processing 2-8

D

- datasets
 - limits with TIOT 1-31
 - unlimited with XTIOI 1-31
- dataspaces
 - access to IBM real storage 1-26

- definition of 1-25
- DSPOPT parameter 1-79
- putting CCATEMP and CCASERVR in Storage 1-25
- tracking loads with APSYRDD 1-29
- date-timestamps
 - obsolete for ROLL BACK processing 2-12
- deadlocks
 - definition of 1-14
 - programming to avoid 1-14
- decryption keys
 - installation kit requirements 1-4
- DEFINE command
 - TIOT and XTIO options 1-31
- DEFINE LINK command
 - for a server 1-73
 - for the client 1-73
- DEFINE PROCESSGROUP command
 - PORT argument 1-72
 - TCP/IP for Horizon or Horizon/LE 1-72
- Dictionary/204 subsystem
 - INS204 task parameter 2-17
- DISTINCT processing
 - using the ORDERED attribute 2-3
- DKPR statistic
 - comparing with 1-61
- DKPRF statistic
 - comparing with DKPF 1-61
 - tracking fast reads 1-61
- DO WHILE loops
 - continuing with \$STATUSR 1-47
- downloading from CCA Web site
 - information and zaps 1-5
- DSPOPT parameter
 - data- and hiperspace options 1-79
 - using expanded storage 1-29
 - with APSY Precompiled Procedures in Storage 1-29
 - with CCASERVR in Storage 1-29
- dynamic file allocation
 - bypassed, if not necessary 1-12
 - using TIOT and XTIO 1-31

E

- Early Warnings
 - information displayed in CCAPRINT file 2-3
 - on CCA Web site 1-4
- ECB-related functions
 - introducing 1-13
- ECBs
 - assigning a number to 1-14
 - check to see if posted 1-43

- definition of 1-13
- getting associated data string 1-41
- number assigned 1-88
- referencing by number 1-15
- resetting 1-47
- set a data string 1-42
- special ECB, CPQZ 1-41, 1-42
- suspend user until posted 1-49
- using the assigned number 1-14
- wait types 2-13
- ECF
 - \$STATUS return codes 3-11
 - \$STATUSD return codes 3-11
 - images 3-2
 - managing subtasks 3-15
 - overhead loading programs 3-15
 - restrictions and cautions 3-17
 - subtask assignment 3-15
 - User 0 parameters 3-14
 - See also* External Call Facility
- ECISUBS parameter
 - setting for subtask affinity 3-16
- ECMSUBS parameter
 - setting for subtask affinity 3-16
- escape character
 - continued use of 2-9
 - usage in pattern matching 4-6
- Event Control Block *See* ECBs
- example programs
 - a wait for an extended quiesce 1-20
 - defining a common variable 1-33
 - third-party backups 1-19
 - VALUE IN clauses 1-35
- EXCPVR
 - support limited for z/OS 2-19
- Expanded Storage Only (ESO)
 - hiperspaces 1-80
- extended quiesce
 - and ring stream journals 1-19
 - definition of 1-18
 - ending 1-23
 - limit on \$POST calls 1-22
 - managing 1-21
 - managing the duration of 1-18
 - programming a wait for 1-20
 - status messages 1-22
 - understanding 1-18
- Extended Task Input/Output Table. *See* XTIO 1-31
- external backup
 - coordinating 1-18
- External Call Facility
 - definition of 3-2
 - loading a program 3-2
 - See also* ECF

- EXTERNAL CALL statement
 - external module availability 3-16
 - invoking an external module 3-16
 - syntax for 3-5
- EXTERNAL DELETE statement
 - syntax for 3-5
- EXTERNAL LOAD statement
 - reloading a previously-loaded module 3-16
 - syntax for 3-6
- external modules
 - logging users off of 3-4
 - reloading 3-16
- EXTERNAL NAME statement
 - syntax for 3-8
- EXTERNAL PROGRAM statement
 - syntax for 3-6
- external programs
 - acceptable work 3-2
 - loading 3-6
 - logical name 3-2
 - stopping 3-3
 - subtask assignment 3-3
- EXTERNAL START statement
 - syntax for 3-8
- EXTERNAL statements
 - return codes 3-11
- EXTERNAL STOP statement
 - syntax for 3-8

F

- fast read
 - definition of 1-61
 - tracking with DKPRF statistic 1-61
- fast reads
 - improving with MAXOBUF setting 1-61
 - improving with SCHDOPT setting 1-61
- FEO has table
 - UFEOHASH parameter 1-95
- FEO hash table
 - understanding 1-95
- field values
 - NULL values in SQL processing 1-63
 - SELECT DISTINCT processing 1-63
- fields
 - subscripted extraction 1-61
- file pages
 - buffer maximum 1-84
- FIND ALL RECORDS statement
 - with VALUE IN clause 1-34
- FIND ALL VALUES statement
 - accepting VALUE IN clause 1-35
- FIND statement

- performance enhancement 2-10
- FLOAT attribute
 - invalid numeric data 1-53
- floating point numbers
 - locating incorrectly stored records 1-53
 - SQL processing 1-55
 - storing 1-53
- FOR %variable statement 4-7
- FOR EACH OCCURRENCE loops
 - performance enhancement 1-58
 - UFEOHASH parameter 1-95
- FOR EACH RECORD WHERE statement
 - with VALUE IN clause 1-36
- FOR RECORD NUMBER option
 - and remote file nonsupport 1-37
- FOR statement
 - skipping loops without values 4-6
- FORTRAN
 - mathematics \$function runtime libraries 2-10
- free threads for software
 - Connect ★ 1-24
 - Horizon 1-24
 - PQO 1-24
- Freeway/204
 - free seats to CCA products 1-24
 - installation consideration 1-24
- FROM phrase, in index loops 4-7
- FRV fields
 - SQL SELECT DISTINCT processing 1-63

G

- generating compile errors
 - IF statements 2-10
- global found sets and lists
 - clearing 4-11
- global images and screens
 - clearing GTBL 4-9
- global objects
 - tracking GTBL rearrangements for 1-60
- global string variables
 - GTBLPCT parameter setting 1-83
 - tracking GTBL rearrangements for 1-60
- global variable (GTBL) table
 - clearing objects from 4-9
 - REMEMBER statement 4-7
- GTBL
 - calculating space requirements 1-83
 - rearranging and tracking 1-60
- GTBLHASH parameter
 - buckets for global table 1-82
- GTBLPCT parameter
 - percent of GTBL for global string variables 1-83

- GTBLRS statistic
 - tracking GTBL rearrangement for global objects 1-60
- GTBLRU statistic
 - GTBL rearrangements for global string variables 1-60

H

- handling large SQL statements
 - reviewing parameters 1-64
- hash collisions
 - definition of 1-96
- High Level Assembler 2-15
- Hiperspace pages
 - storing APSY precompiled procedures 1-80
- hiperspaces
 - definition of 1-26
 - DSPOPT parameter 1-79
 - Expanded Storage Only (ESO) 1-80
- HLASM 2-15
- Horizon
 - free threads 1-24
 - INS204 task parameter 2-18
- Horizon for TCP/IP
 - converting from Horizon for VTAM 1-74
 - limitations for VSE 2-22
 - process-to-process communications 1-72
 - specifying a port number 1-72
 - VSE limitation 1-74
- Horizon for VTAM
 - converting to Horizon for TCP/IP 1-74
- Horizon or Horizon/LE for TCP/IP
 - CNOS limitations 1-74
 - process-to-process communications 1-72
 - VSE operating system 1-74, 2-22
- Horizon/LE for TCP/IP
 - Connect★ communications-only 1-72
 - limitations for VSE 2-22

I

- IBM
 - APARs required 1-72
 - choosing LE runtime libraries 2-10
 - Maximum Segment Life units 1-75
 - restrictions with NOTE/POINT 2-22
 - SNA LU62 1-74
 - support for access to real storage 1-26
 - TIME-WAIT state 1-74
- IF statements
 - generating compile errors 2-10
- IFAM buffer

- SQLIQBSZ parameter 1-64
- IFAM2 applications
 - 24-bit 2-20
 - 31-bit 2-20
 - under VM 2-20
- improved processing 1-11
- IN GROUP MEMBER clause
 - enhancement 1-37
- In Storage feature
 - APSY precompiled procedures 1-28
 - CCATEMP and CCASERVR 1-25
- index loops
 - processing 4-7
 - processing array elements 4-7
- initialization
 - subsystems available 2-3
- INNER JOIN keywords
 - SQL features 1-68
- INS204
 - changed defaults for tasks 2-17
 - installation changes 1-24
- installation changes
 - for Freeway/204 1-24
- installation kit requirements
 - CPU ID zap 1-4
 - decryption keys 1-4
 - mainframe cartridges 1-4
 - tape map 1-4
- installing SoftSpy
 - required documentation 1-3
- invalid floating point data
 - understanding storage 1-53
- invalid numeric data
 - storing 1-53
- INVISIBLE fields
 - REDEFINE command, rejecting 2-7
- IOS Branch Entry
 - and XTIOU usage 1-32

J

- JCL
 - reusing for RESTART recovery 1-8
- Journal buffers, setting the number of 1-88

K

- KEY attributes
 - in CCACAT file 2-3

L

- LE COBOL program

- creating 2-20
- LENQTBL parameter 1-83
- LHEAP parameter
 - evaluate for large SQL statements 1-64
 - increase in size 2-5
- login delays
 - and BUMP command 4-5
 - and LOGWHO command 4-5
 - automatic security feature 4-4
 - invoking security 4-4
 - STATUS command 4-5
- login failure
 - MONITOR command 4-5
- LOGOUT command
 - changing message number 1-78
- LOGOUT message
 - and CUSTOM=6 1-78
- LOGWHO command
 - unsuccessful logins 4-5
- loops
 - index 4-7

M

- mainframe cartridges
 - installation kit requirements 1-4
- managing
 - checkpoint extended quiesce 1-19
 - third-party backups 1-18
- mathematics \$functions
 - choosing a runtime library 2-10
- MAXBUF parameter
 - file page buffer maximum 1-84
- Maximum Segment Life (MSL)
 - IBM wait unit 1-75
- MAXINCL parameter
 - INCLUDE nesting levels 1-85
- MAXOBUF parameter
 - calculating MINBUF 1-86
 - improving fast page reads 1-61
 - open disk buffers per server 1-85
- message location offset
 - upward incompatibility 2-11
- messages
 - displayed for ROLL BACK processing 1-9
 - paired for more information 2-12
- MINBUF parameter
 - file page buffer minimum 1-86
- Model 204
 - accessing most recent updates 1-4
 - and OS/390 and z/OS operating systems 2-17
 - documentation library CD 1-4
 - field definitions with SQL SELECT DISTINCT

- 1-63
- Model 204 commands
 - ALLOCATE 1-12, 1-31
 - BUMP MODULE 3-4
 - CHECKPOINT 1-97
 - CHKABORT 2-7
 - CHKMSG 1-98, 2-7
 - DEFINE 1-31
 - DEFINE PROCESSGROUP 1-72
 - enhanced
 - 1-97
 - MONITOR 1-100, 1-101
 - MONITOR CFR 1-98
 - MONITOR DATASPACE 1-26
 - MSGCTL 1-104
 - MSGCTL Version 5. processing 2-7
 - obsolete form of ALLOCATE 1-32
 - RENAME FILE 1-104
 - REORGANIZE ORDERED INDEX 1-53
 - VIEW SERVSZ 1-105
- Model 204 image
 - external programs 3-2
- Model 204 parameters
 - APSYPAGE 1-76
 - AUDITOPT 1-76
 - CPQZACTN 1-76, 1-77
 - CUSTOM 1-78
 - DSPOPT 1-79
 - GTBLHASH 1-82
 - GTBLPCT 1-83
 - LHEAP 1-64
 - MAXBUF 1-84
 - MAXINCL 1-85
 - MAXOBUF 1-85
 - MINBUF 1-86
 - MPSYS 1-87
 - MSRES 1-87
 - NBKPG 2-7
 - NECBS 1-88
 - obsolete in Version 5.1 2-15
 - OPSYS 1-89
 - RCVOPT 1-11
 - RESLTHR 1-90
 - SCHODOPT 1-90
 - SECTRLOG 4-2
 - SERVSZ 1-64
 - SLQIQBSZ 1-64
 - SQLBUFSZ 1-64
 - SYSOPT2 1-31, 1-92
 - TEMPPAGE 1-94
 - UFEOHASH 1-95
 - UFEOMISS 1-96
 - VTBL 1-64
- MONITOR command

- CRFLOOK parameter 1-99
 - displaying running users 2-13
 - monitoring an Online 1-100
 - monitoring CFR 1-98
 - monitoring checkpoints 1-101
 - reformatted 2-11
 - unsuccessful logins 4-5
- MONITOR DATASPACE command
 - tracking CCASERVR and CCATEMP in Storage 1-26
- MONITOR SL command
 - reformatted 2-11
- MONITOR VTAM command
 - message location offset 2-11
- MP lock
 - spinning on for improved performance 1-59
- MP/204
 - adjusting MAXOBUF setting 1-86
 - fast logical page reads 1-61
 - processing options 1-87
 - scheduling operation and accounting 1-90
- MPSYS parameter
 - MP/204 processing options 1-87
- MQ/204
 - auditing messages 1-76
- MSGCTL command
 - and CUSTOM=7 setting 2-8
 - setting message output 1-104
 - Version 5.1 changes 2-7
- MSRES parameter
 - Single Record Entries 1-87

N

- NATURAL JOIN feature
 - common columns 1-68
- NBKPG parameter
 - accurately counting backpages 2-7
- NECBS parameter
 - and \$ECBDGET function 1-41
 - and \$ECBDSET function 1-42
 - referencing an ECB 1-15
 - tally of ECBs 1-88
- New features
 - ECF for OS/390 and z/OS 3-2
 - VSE operating system 2-22
- NJBUFF parameter 1-88
- NOAUDIT option
 - for MSGCTL command 1-104
- NonStop/204
 - using third-party backups 1-18
- NOSWAP option
 - caution for \$WAIT 1-50

- NOTE/POINT facility
 - handling CCARF datasets 1-11
 - limitation in VSE 2-22
- NSERVS parameter
 - calculating MINBUF 1-86
- NSUBTKS parameter
 - calculating MINBUF 1-86
- NULL values
 - SELECT DISTINCT processing 1-63
- numeric data
 - detecting invalid 1-48

O

- object replacement modules
 - on CCA Web site 1-4
- Operating system settings 1-89
- OPSYS parameter
 - identifying the operating system 1-89
- ORDERED attribute
 - DISTINCT processing 2-3
 - SQL WHERE processing 1-63
- ORDERED CHARACTER attribute
 - in CCACAT file 2-3
- Ordered Index
 - storing ORDERED NUMERIC attribute values 1-53
- ORDERED NUMERIC attributes
 - in CCACAT file 2-3
 - invalid numeric data 1-53
 - storing invalid values 1-53
- OS/390 and z/OS operating systems
 - and Model 204 2-17
 - ECF subtasks 3-3
 - installation changes 1-24
- OUTER JOIN features
 - preserving unmatched rows 1-69
 - restrictions 1-70

P

- page reads
 - fast logical 1-61
- Parallel Query Option (PQO)
 - free thread 1-24
 - INS204 task parameter 2-18
- parameters
 - COMEND in XREF facility 1-56
 - COMSTART in XREF 1-56
 - reviewing for large SQL statements 1-64
- password table
 - login privileges 4-5
 - login security 4-5

- pattern characters
 - usage as a character 2-9
- pattern matching
 - \$CHKPAT function 2-9
 - changes to 2-9
 - verifying syntax 4-12
- PDL usage
 - increases 2-5
- Peer Protocol Reference
 - conforming to 1-74
- performance enhancements
 - CCARF dataset scanning 1-11
 - dynamic file allocation 1-12
 - FIND statement 2-10
 - finding checkpoints quickly 1-11
 - FOR EACH OCCURRENCE loops 1-58
 - reduced Resource Enqueue Table entries 1-58
 - RESTART recovery 1-11
 - ROLL BACK processing 1-60
 - ROLL FORWARD processing 1-60
 - spinning on an MP lock 1-59
 - SQL processing 1-63
- performance improvements
 - subscripted field extraction 1-61
- PORT parameter
 - DEFINE PROCESSGROUP command 1-72
- POSITION statement
 - FOR processing 4-8
- posting
 - CPQZ 1-19
 - QZSIG 1-19
- posting an ECB
 - releasing waiting threads 1-15
 - scope of 1-14
- privilege byte, values for 4-5
- processing Index loops
 - evaluating expressions 4-7
- process-to-process communications
 - Horizon and Horizon/LE for TCP/IP 1-72

Q

- QTBL size
 - increase in V5.1 2-5
- QZSIG
 - posting and unposting 1-19

R

- RCVOPT parameter
 - setting for RESTART recovery 1-11
- recovery
 - automated 1-8
 - forcing a successful rerun 1-9
 - statistics 4-5
 - entry types 4-5
 - tracking updates 1-10
 - REDEFINE command
 - and INVISIBLE field behavior 2-7
 - REFERENCES clause
 - in syntax 1-64
 - reinstalling
 - CRAM 2-17
 - REMEMBER statement 4-7
 - remote files
 - FOR RECORD NUMBER option nonsupport 1-37
 - Remote servers
 - defining 1-72
 - RENAME FILE command
 - avoiding DUMP or RESTORE commands 1-105
 - renaming a file 1-104
 - REORGANIZE ORDERED INDEX command
 - locating floating point values 1-53
 - requirements
 - calculating GTBL space 1-83
 - for BATCH2 backup 1-22
 - storage for APSY precompiled procedures
 - APSY precompiled procedures
 - storage requirements 1-28
 - reserved words
 - for SQL 1-71
 - RESLTHR parameter
 - APSY resident load threshold 1-90
 - Resource Enqueue Table
 - reduced entries performance enhancement 1-58
 - RESTART recovery
 - bypassing dynamic file allocation 1-12
 - compatibility issue 2-2
 - performance improvements 1-11
 - rerunning after successful recovery 1-9
 - using TIOT or XTOT 1-31
 - RESULT macro
 - limitations 2-10
 - ring stream journals
 - and extended quiesce 1-19
 - ROLL BACK processing
 - CHKPOINT dataset required 1-9
 - forcing 1-10
 - messages displayed 1-9
 - obsolete date-timestamp 2-12
 - running independently 1-9
 - speed enhanced 1-60
 - ROLL FORWARD processing
 - CCARF dataset 1-10

- enhanced CHKPOINT records 1-60
- estimating time to complete 1-10
- operational changes 1-10
- run separately 1-10
- RSQL engine
 - INS204 task parameter 2-18
 - Version 5.1 enhancements 1-66
- RUNG queue
 - displaying 1-100
- running users
 - displaying accurate count of 2-13

S

- sample programs
 - submitting a backup 1-20
 - thread synchronization 1-15
- SCHDOPT parameter
 - and since-last statistics 1-91
 - calculating since-last statistics 1-91
 - improving fast page reads 1-61
- scheduler queues
 - displaying running users 2-13
- SCHODOPT parameter
 - MP/204 scheduling 1-90
- secondary recovery
 - automated 1-8
- SECTRLOG parameter
 - security trusted logins 4-2
 - unavailable to TPROCESS applications 4-2
- Security
 - login delays 4-4
- security
 - login delays 4-4
 - login password table 4-5
 - login privilege bytes 4-5
 - monitoring login 4-5
- SELECT DISTINCT statement
 - Model 204 field definitions affected 1-63
 - processing Model 204 field values 1-63
- SELECT LIST statement
 - assigning alias names 1-66
 - correlation names in 1-66
 - using an alias in 1-66
- SELECT statement options
 - CURRENT_DATE 1-67
 - CURRENT_TIME 1-67
 - CURRENT_TIMESTAMP 1-67
 - ON clause for conditional join 1-69
 - SQLERROR 1-68
 - SQLSTATE 1-68
 - SQLVERSION 1-67
 - USER 1-67
- sequential data files
 - limitations in VSE 2-22
- server sizes
 - increase in V4R2.1 2-5
- servers
 - defining a link to 1-73
- SERVSZ parameter
 - evaluate for large SQL statements 1-64
- setting message output
 - MSGCTL command 1-104
- signaling threads
 - facilities used 1-13
- Since-last statistics
 - user 1-110
- since-last statistics
 - resetting SCHDOPT 1-91
- size increase requirements
 - LHEAP 2-5
 - QTBL 2-5
 - server 2-5
 - VTBL 2-5
- small page area
 - tracking pages used 1-93
- SoftSpy
 - documentation 1-3
- Software required
 - CRAMINS job 2-17
- SQL join features
 - INNER JOIN 1-68
 - NATURAL JOIN 1-68
 - ON clause 1-69
 - OUTER JOIN 1-69
 - OUTER JOIN restrictions 1-70
 - USING clause 1-69
- SQL processing
 - floating point numbers 1-55
 - improvements and changes 1-63
- SQL SELECT DISTINCT statements
 - FRV fields 1-63
- SQL statements
 - enhancements 1-66
 - larger than 32K bytes 1-64
 - larger the 32k bytes 1-64
- SQL support subsystems
 - INS204 task parameter 2-17
- SQL WHERE processing
 - Model 204 ORDERED fields 1-63
- SQLBUFSZ parameter
 - handling large statements 1-64
- SQLERROR keyword
 - SELECT statement option 1-68
- SQLIQBSZ parameter
 - IFAM buffer 1-64
- SQLSTATE keyword

- SELECT statement option 1-68
- SQLVERSION keyword
- SELECT statement option 1-67
- Statistics
 - user 1-110
 - since-last 1-110
 - statistics
 - APSY load 1-29
 - fast page reads 1-61
 - for GTBL hashing 1-60
 - new in Version 5.1 1-107
 - obsolete in Version 5.1 2-15
 - since-last and SCHDOPT setting 1-91
- STATUS command
 - and unsuccessful logins 4-5
- status messages
 - for extended quiesce 1-22
- STSSYM member
 - reassembling for new statistics 1-107
- STSYS member
 - reassembling for new statistics 1-107
- STUSER member
 - reassembling for new statistics 1-107
- STUSYM COPY member
 - reassembling for new statistics 1-107
- subscripted fields
 - improved value extraction 1-61
- subsystems
 - available at initialization 2-3
- subtask affinity
 - definition of 3-15
 - setting ECISUBS parameter 3-16
 - setting ECMSUBS parameter 3-16
- subtasks
 - assignment in ECF 3-15
- synchronizing threads
 - facilities used 1-13
- SYSOPT2 parameter
 - determining XTLOT or TIOT 1-31
 - system options 1-92
- System programmers
 - consideration for CCATEMP in Storage 1-27
 - considerations for CCASERVR in Storage 1-27

T

- tape map
 - installation kit requirements 1-4
- Task Input/Output Table. *See* TIOT 1-31
- TCP/IP
 - and Horizon or Horizon/LE 1-72
- TCPIP
 - INS204 task parameter 2-18

- TEMPCUR parameter
 - CCATEMP high-water mark 1-92
- TEMPCURE parameter
 - expansion area pages 1-92
- TEMPCURS parameter
 - pages in small page area 1-93
- TEMPHIE parameter
 - small page area highest count 1-93
- TEMPHIS parameter
 - tracking small page area 1-93
- TEMPMAX parameter
 - CCATEMP page high-water mark 1-93
- TEMPPAGE parameter
 - CCATEMP pages in storage 1-94
 - compared to APSYPAGE 1-28
- TEMPSIZE parameter
 - highest CCATEMP page count 1-94
- third-party backup
 - programming 1-19
- third-party backups
 - managing 1-18
 - NonStop/204 1-18
 - sample procedure 1-20
- thread synchronization
 - programming 1-15
- threads
 - implementing signaling 1-13
 - implementing synchronization 1-13
 - programming synchronization 1-14
 - released from waiting 1-15
 - signaling and synchronization 1-13
 - User Language programs synchronizing 1-15
 - waiting for an ECB 1-45
- TIME-WAIT state
 - to flush old packets 1-74
- tiny load
 - definition of 1-30
- tiny loads
 - tracking with APSYLDT 1-30
- TIOT
 - dataset limits 1-31
 - option in ALLOCATE command 1-31
 - reallocation for RESTART recovery 1-31
 - with dynamically allocated files 1-31
- TO phrase, in index loops 4-7
- TPROCESS
 - nonsupport for SECTRLOG parameter 4-2
- TPROCESS applications
 - SECTRLOG parameter not supported 4-2
- tracking users
 - AUDIT204 reports 2-22

U

- UFEOHASH parameter
 - FEO hash table 1-95
 - optimize FEO loops 1-95
- UFEOMISS parameter
 - and hash collisions 1-96
 - missed optimizations for FEO 1-96
- unposting
 - CPQZ 1-19
 - QZSIG 1-19
- unposting an ECB 1-15
 - timing of 1-14
 - while waiting 1-15
- updates
 - tracking during recovery 1-10
- usage
 - PDL increases 2-5
- USER keyword
 - SELECT statement option 1-67
- User Language procedures
 - support for Block Comments 1-56
- User Language programming
 - thread synchronization 1-15
- User Language statements
 - FIND ALL RECORDS 1-34
 - FIND ALL VALUES 1-35
 - FIND performance improvement 2-10
 - FOR EACH RECORD WHERE 1-36
- user-written \$functions
 - changes required in Version 5.1 2-10
- USING clause feature
 - matching column names 1-69

V

- VALUE IN clause
 - in FIND ALL VALUES statements 1-35
 - increased use of PDL 2-5
 - value set processing 1-33
 - with FIND ALL RECORDS statement 1-34
 - with FIND ALL VALUES statement 1-35
 - with FOR EACH RECORD WHERE statement 1-36
- VALUE IN clause processing
 - rewriting applications with nested loops 1-34
- VALUE IN clauses
 - program example 1-35
- variables
 - defining one for common use 1-33
- Version 5.1
 - new statistics 1-107
- VIEW SERVSZ command

- display server size 1-105
- VM operating system
 - 31-bit BATCH2 processing 2-20
 - IFAM2 processing 2-20
 - new features 2-20
- VSE operating system
 - CICS Transaction Server 2-22
 - creating a COBOL II program 2-20
 - Horizon for TCP/IP 2-22
 - Horizon for TCP/IP limitation 1-74
 - limitation of Horizon/LE for TCP/IP 2-22
 - limitation with NOTE/POINT facility 2-22
 - new features 2-22
 - not supporting TCP/IP for Horizon or Horizon/LE 1-74, 2-22
- VTAM
 - automatic installation 2-19
 - unencrypted 1-24
- VTBL
 - increased size 2-5
- VTBL parameter
 - evaluate for large SQL statements 1-64

W

- wait types
 - for ECBs 2-13
- waiting on an ECB
 - scope of 1-14
- wildcard
 - asterisk (*) 1-66
- workflow
 - suggested for SQL processing 1-75

X

- XDM
 - implementing 4-4
- XREF facility
 - Cross Reference Selection screen 1-56
 - support for Block Comments in procedures 1-56
- XTIOT
 - and IOS Branch Entry 1-32
 - datasets unlimited 1-31
 - option in ALLOCATE command 1-31
 - reallocation for RESTART recovery 1-31
 - with dynamically allocated files 1-31

Z

- z/OS
 - support limited for EXCPVR 2-19
- z/OS operating system

introducing 2-17
zaps
downloading from CCA Web site 1-5

