

Rocket Model 204 Release Notes

New Features

Version 7 Release 4.0

June 2012
204-0704-NF-02



Notices

Edition

Publication date: June 2012

Book number: 204-0704-NF-02

Product version: Rocket Model 204 Release Notes Version 7 Release 4.0

Copyright

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

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

Trademarks

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

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

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

Note

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

Contact information

Website: www.rocketsoftware.com

Rocket Software, Inc. Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA
Tel: +1 781 577 4321
Fax: +1 617 630 7100

Contacting Global Technical Support

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

Email: m204support@rocketsoftware.com

Telephone:

North America	+1 800 755 4222
United Kingdom/Europe	+44 (0) 20 8867 6153

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

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

www.rocketsoftware.com/support

Contents

Notices	2
Contacting Global Technical Support	3
Chapter 1: Features and enhancements	7
Overview	7
New in this release	7
Installation considerations	12
Accessing the Rocket Model 204 Documentation	13
Support for IPv6 addressing	14
\$SNDMAIL function	14
\$SOCKET function	15
Message changes related to IPv6 addressing	15
New MONITOR LINK format	15
New MONITOR PROCESSGROUP format	16
IPv4-mapped IPv6 addresses	17
\$SOCKETX	17
DEFINE LINK: Horizon for TCP/IP	19
DEFINE PROCESSGROUP: Horizon for TCP/IP	19
Compatibility issues	20
User Language enhancements	20
\$SNDMAIL function	20
\$SOCKETX function	20
\$DEPAD, \$DEPADR, and \$DEPADL functions	21
External Call Facility (ECF)	21
Notification of loss of pattern optimization	21
Expressions in User Language statements	22
Expressions as field values within update statements	24
FILE RECORDS statement	25
FIND Statement EQ VALUE clause	26
FIND ALL VALUES	27
FIND statement – POINT\$ condition	27
FIND statement — SFL\$ and SFGE\$ conditions	28
FOR EACH RECORD IN ORDER BY	28
FOR EACH RECORD WHERE	29
FOR EACH VALUE	29
FOR RECORD NUMBER	30
AUDIT ALL INFORMATION statement	31
STORE RECORD block	31
ANDIF and ORIF short-circuit operators	34
Operating system and peripherals support	35
XDM improved ALET (Access List Entry Token) capacity	35
Extended Address Volumes (EAV)	36
Performance enhancements	36
Model 204 zIIP support	36
zIIP support in Model 204	36
Model 204 workload eligible for zIIP offload	37
Model 204 zIIP infrastructure created during initialization	37
Model 204 parameters for zIIP support	37
Model 204 zIIP-related statistics and monitoring	38
zIIP utilization	39
zIIP-compatible functions	39
New and enhanced zIIP statistics	39
1 MB virtual pages	40
FTBL and server storage above the bar	41
EBM and procedure pages allowed above the bar	43

Coded and binary values no longer converted to string during SORT	43
Chapter 2: Additional features	44
Utilities enhancements	44
UTILJ enhancements	44
Files	44
Date/time stamp (DTS) file update under IFAM	44
Creating and using in-memory Model 204 files	44
Recovery, Reliability, and CCAJRNL	47
IGNORE option for RESTART command	47
Changes to journal record layouts	47
Increased journal block header information	47
Expanded journal entries	48
Model 204 SQL Catalog enhancements	48
LOB fields in SQL statements	48
SQL Catalog (CCACAT file) implementation	50
MQ/204	51
MQ/204 DEFINE QUEUE statement enhancement	51
Other features	51
Subroutine processing	51
New demonstration database files	51
Chapter 3: Upward compatibility issues	52
Increased SERVSZ requirement	52
CCAJRNL and CHPNT data sets	52
ANDIF and ORIF reserved words	52
Long DDNAME in OPEN or CREATE commands	53
BLDREUSE command with NEW option	53
\$DSN and \$DSNNUM functions	53
FLOD multistep processing	53
Horizon IPv6 addressing and symbolic addresses	54
STORE RECORD limitation	54
Chapter 4: Security enhancements	55
Mixed-case login password support	55
Example of lowercase login and password	55
Chapter 5: New and enhanced parameters and commands	57
Parameters	57
ASTRPPG	57
CUSTOM	57
DSPOPT	58
JOBSTEP	59
OPSYS	59
SCHDOFL	60
SCHDOFLZ	61
SCHDOPT	62
SCHDOPT1	63
SERVNSA	63
SERVNSSZ	64
SMTPCADD	64
SMTPSADD	64
STORCURG	65
SQLBUFSZ	65
STORMAXG	66
STORMXZP	66
STORZPAG	66
TCPOPTS	66
TEMPBOP	67
ZPAGEOPT	68
ZQMAX	68

Commands.....	68
ALLOCATE	68
CLOSE LINK.....	69
COMPACTB.....	69
DEFINE DATASET	70
LOGCTL.....	70
MONITOR GSTORAGE	71
MONITOR TASKS.....	72
OPEN LINK.....	74
RESTART.....	74
USE PROC.....	74
Chapter 6: Horizon IPv6 addressing support	76
Horizon and sockets	76
DEFINE LINK command for TCP/IP Horizon	77
DEFINE PROCESSGROUP command for TCP/IP Horizon.....	77
Chapter 7: Connect* Suite enhancements	78
Connect* for JDBC.....	78
Connect* for .NET Framework	79
IPv6 address support for Connect*	80
Chapter 8: New and updated messages	82
Messages	82
Index.....	101

Chapter 1: Features and enhancements

Topics covered in this chapter

[Overview](#)
[Installation considerations](#)
[Accessing the Rocket Model 204 Documentation](#)
[Support for IPv6 addressing](#)
[User Language enhancements](#)
[Operating system and peripherals support](#)
[Performance enhancements](#)

Overview

This chapter contains installation and features information for the Rocket Model 204 version 7.4.0 release.

Before beginning your installation, please read through this information about product installation and changes.

New in this release

This topic describes the major new features and enhancements for Model 204 version 7.4.0.

Support for IPv6 addressing

Feature	Operating System	Description
IPv6 addressing	z/OS	M204 now supports Internet Protocol version 6 (IPv6) addressing. See Support for IPv6 addressing, on page 14

User Language

Feature	Operating System	Description
Functions support for IPv6 addressing	z/OS	User Language functions supporting IPv6 addressing: <ul style="list-style-type: none">▪ \$SNDMAIL function can handle IPv6 addressing▪ \$SOCKET function supports IPv6 network addresses▪ \$SOCKETX GETNFO call can return IPv4 and/or IPv6 information See Support for IPv6 addressing, on page 14
\$SNDMAIL in BATCH204	z/OS, z/VSE	\$SNDMAIL is now supported in BATCH204 if relinked with certain objects. See \$SNDMAIL function, on page 20
\$DEPAD, \$DEPADR, \$DEPADL functions	z/VM, z/OS, z/VSE	The new functions \$DEPAD, \$DEPADR, and \$DEPADL remove characters from a string. See \$DEPAD, \$DEPADR, and \$DEPADL functions, on page 21

ECF statements	z/VM, z/OS, z/VSE	External Call Facility (ECF) statements can now pass up to 40 parameters. See External Call Facility (ECF), on page 21
Optimizing IS LIKE pattern searches	z/VM, z/OS, z/VSE	When pattern optimization cannot be performed with IS LIKE, an informational message is issued to the audit trail. See Notification of loss of pattern optimization, on page 21
Expressions as field values	z/VM, z/OS, z/VSE	You can use expressions to supply values in many types of User Language statements. See Expressions in User Language statements, on page 22
AUDIT ALL INFORMATION statement	z/VM, z/OS, z/VSE	Model 204 writes the output of entire records to the audit trail. See AUDIT ALL INFORMATION statement, on page 31
STORE RECORD block THEN CONTINUE statement	z/VM, z/OS, z/VSE	For convenience and efficiency you can use a THEN CONTINUE statement within a STORE RECORD block in place of an FRN \$CURREC block that immediately follows the END STORE statement. See STORE RECORD block, on page 31
Short-circuit logical operations (ANDIF and ORIF)	z/VM, z/OS, z/VSE	Compiled code can bypass the evaluation of a second expression depending on the result of the previous expression(s). See Short-circuit operators, on page 34

Operating system and peripherals support

Feature	Operating System	Description
XDM improved ALET capacity	z/VM, z/OS, z/VSE	The XDM (Cross-memory Data Mover) has been modified to allow the full use of ALETs up to the IBM limit of 512. See XDM improved ALET (Access List Entry Token) capacity, on page 35
EAV support	z/OS	Support for allocation and use of files on Extended Address Volumes (EAV) above the 64K cylinder boundary. See Extended Address Volumes (EAV), on page 36

Performance

Feature	Operating System	Description
zIIP support	z/OS	Model 204 zIIP support enables you to offload Model 204 work from regular processors to zIIP processors. See zIIP support in Model 204, on page 36
1MB virtual pages	z/OS	1MB pages are now supported with storage above the bar. See 1 MB virtual pages, on page 40

FTBL and above-the-bar storage	z/VM, z/OS	FTBL can now reside in 64-bit storage (above the bar). See FTBL and server storage above the bar, on page 41
EBM pages and above-the-bar storage	z/VM, z/OS	Existence Bit Map (EBM) pages and procedure pages are now eligible to be placed in above-the-bar storage. See EBM and procedure pages allowed above the bar, on page 43

Utilities

Feature	Operating System	Description
UTILJ FILENAME=	z/VM, z/OS	You can now specify the argument FILENAME= in the parameter list for UTILJ if a CCAOUT DD card is present. See UTILJ enhancements, on page 44

Parameters and Commands

For a listing of all new and enhanced parameters and commands in this release, see [New and enhanced parameters and commands, on page 57](#).

Feature	Operating System	Description
IPv6 addressing	z/OS	New parameter to support IPv6 addressing: TCPOPTS. Enhanced parameters: SMTPCADD, SMTPSADD. Enhanced commands: MONITOR LINK, MONITOR PROCESSGROUP; TCP/IP Horizon commands DEFINE LINK and DEFINE PROCESSGROUP. See Support for IPv6 addressing, on page 14
zIIP processing	z/OS	New parameters to support zIIP processing: AMPSUBZ, NMPSUBZ, SCHDOFLZ, SCHDOPT1, ZQMAX. Enhanced parameter: SCHDOPT. Enhanced command: MONITOR TASKS. See zIIP support in Model 204, on page 36
1 MB virtual pages	z/OS	New parameters to support 1MB virtual pages: STORZPAG, STORMXZP, ZPAGEOPT. Enhanced parameter: OPSYS. New command: MONITOR GSTORAGE. See 1 MB virtual pages, on page 40
FTBL above-the-bar storage	z/VM, z/OS	New parameters to support FTBL ATB storage: SERVNSSZ, SERVNSA, STORCURG, STORMAXG. See FTBL and server storage above the bar, on page 41
Extended Address Volumes (EAV)	z/OS	Enhanced parameter to support EAV: OPSYS. Enhanced command: ALLOCATE. See Extended Address Volumes (EAV), on page 36
In-memory Model 204 files	z/VM, z/OS	Enhanced commands to support in-memory Model 204 files: DEFINE DATASET, ALLOCATE. See Creating and using in-memory Model 204 files, on page 44
Support for Large Objects (CLOB, BLOB)	z/VM, z/OS, z/VSE	Enhanced parameter: SQLBUFSZ. See SQLBUFSZ, on page 65

Back out pages	z/VM, z/OS, z/VSE	The TEMPBOP parameter sets the number of CCATEMP pages reserved for back out pages for subtransactions that have already started. See TEMPBOP, on page 67
Combining extension records	z/VM, z/OS, z/VSE	The new command COMPACTB combines as many extension records as possible into one extension record for the current or specified file. See COMPACTB, on page 69
Formatting USE data set output saved to a temporary user procedure	z/VM, z/OS, z/VSE	Enhanced command: USE PROC. See USE PROC, on page 74
JCL PROC identification	z/VM, z/OS, z/VSE	Newly viewable parameter: JOBSTEP. See JOBSTEP, on page 59

Files

Feature	Operating System	Description
Updating DTS files under IFAM	z/VM, z/OS, z/VSE	IFAM applications can now execute all update functions on a DTS file. See Date/time stamp (DTS) file update under IFAM, on page 44
In-memory Model 204 files	z/VM, z/OS	You can create temporary in-memory Model 204 files, which incur less CPU overhead and none of the I/O associated with permanent Model 204 files. See Creating and using in-memory Model 204 files, on page 44

Recovery, Reliability, and CCAJRN

Feature	Operating System	Description
IGNORE option for RESTART command	z/VM, z/OS, z/VSE	The RESTART command IGNORE option enables you to remove a problem file from restart. See IGNORE option for RESTART command, on page 47
Increased journal block header information	z/VM, z/OS, z/VSE	Various journal analysis utilities require additional journal information at sites that embrace GDG streams and the SWITCH STREAM command as a means to keeping their Onlines up longer than a day. See Increased journal block header information, on page 47
Field journal entries enlarged	z/VM, z/OS, z/VSE	Journal entries for fields increased by two bytes. See Expanded journal entries, on page 48

Model 204 SQL

Feature	Operating System	Description
Support for Large Objects (CLOB, BLOB)	z/VM, z/OS, z/VSE	<p>The Model 204 SQL engine now supports Character Large Objects (CLOBs) and Binary Large Objects (BLOBs). See Model 204 SQL Catalog enhancements, on page 48</p> <p>Model 204 version 7.4.0 is compatible with all versions of Connect*. However, versions of Connect* prior to 7.4.0 do not support CLOB/BLOB data types. See Model 204 compatibility with Connect*.</p> <p>Connect* version 7.4.0 is compatible with all previous supported releases of Model 204 with no restrictions.</p>

Security

Feature	Operating System	Description
Mixed-case login password support	z/VM, z/OS, z/VSE	<p>Mixed-case passwords are supported for improved login security. Supporting mixed-case passwords: CUSTOM parameter and LOGCTL command.</p> <p>See Mixed-case login password support, on page 55</p>

MQ/204

Feature	Operating System	Description
DEFINE QUEUE statement enhancement	z/VM, z/OS, z/VSE	<p>When a queue is known to the default queue manager, as specified by the MQDEQMAN keyword, you can omit an explicit DEFINE QUEUE command.</p> <p>See MQ/204 DEFINE QUEUE statement enhancement, on page 51</p>

Horizon

Feature	Operating System	Description
IPv6 network addresses	z/OS	<p>Horizon supports IPv6 network addresses for TCP/IP.</p> <p>See Support for IPv6 addressing, on page 14</p>
CLOSE LINK... ABANDON command	z/VM, z/OS, z/VSE	<p>You can disable an opened link that will not close because of outstanding VTAM requests.</p> <p>See CLOSE LINK, on page 69</p>
OPEN LINK command	z/VM, z/OS, z/VSE	<p>Using the new REPLACE keyword in an OPEN LINK command, you can sometimes replace an abandoned link and take over its associated REMOTE, SESSIONGROUP and PROCESSGROUP definitions.</p> <p>See OPEN LINK, on page 74</p>

Connect* Suite

Feature	Operating System	Description
Connect* for JDBC enhancement: BLOB and CLOB support	z/VM, z/OS, z/VSE	Connect* for JDBC now provides support for Model 204 BLOB and CLOB data types in SQL update and retrieval statements. See Connect* for JDBC, on page 78
Connect* for .NET Framework enhancements: Visual Studio Integration; BLOB and CLOB support	z/VM, z/OS, z/VSE	<ul style="list-style-type: none"> Connect* for .NET Framework now supports integration of the Model 204 .NET Framework data provider with Microsoft Visual Studio™ 2005. Later versions are not yet supported. Connect* for .Net Framework now provides support for Model 204 BLOB and CLOB data types in SQL update and retrieval statements. See Connect* for .NET Framework, on page 79
IPv6 address support	z/OS	All Connect* clients now support Internet Protocol Version 6 (IPv6) 128-bit addresses as well as IPv4 32-bit addresses. See IPv6 address support for Connect*, on page 80

Installation considerations

Software delivery

Software delivery is now electronic. Installation from tape is no longer supported.

See the *Rocket M204 Installation Guide* for instructions about downloading the installation files from the Rocket M204 FTP site.

Operating system requirements

Model 204 version 7.4.0 requires the following operating system support:

- For IBM z/OS: Version 1 Release 7 is sufficient for all new functionality except for the following features:
 - Large (1 MB) page support requires Version 1 Release 9.
 - Extended address volumes (EAV) requires Version 1 Release 12.
- For IBM z/VM: Version 5 Release 4.0 or later
- For IBM z/VSE:
 - Version 4 Release 3 or
 - Version 4 Release 2.1 or
 - Version 4 Release 2.0, with these program temporary fixes (PTFs) installed:
 - UD53436
 - UD53437
 - UD53438
 - UD53439

Model 204 version 7.4.0 for IBM z/VSE version 5.1 will be available in Summer 2012.

Hardware requirements

Model 204 version 7.4.0 requires the IBM z/890 or above processor, except for the following feature:

- The large (1 MB) page support feature requires the IBM z10 or above processor.

Model 204 compatibility with operating systems

For the most current listing of IBM mainframe computers that are compatible with Model 204 version 7.4.0, go to www.rocketsoftware.com/support and click the **Rocket M204** link.

Model 204 compatibility with Connect*

Model 204 version 7.4.0 has added SQL support for CLOB and BLOB SQL data types. All supported versions of Connect* are compatible with Model 204 version 7.4.0.

However, CLOB/BLOB support requires Connect* version 7.4.0. Use of Connect* versions earlier than version 7.4.0 to SELECT, UPDATE, or INSERT CLOB/BLOB data will fail and might produce unexpected application behavior and/or return an error.

Connect* version 7.4.0 is compatible with versions of Model 204 that are earlier than version 7.4.0.

MP/204 licensing validation provided in CPUID authorization zap

Customers who are licensed for MP/204 no longer require an MP/204 installation decryption key. MP/204 licensing validation is now provided in the CPUID Authorization zap.

Replacement of EVCP with RSQL results in larger module sizes

The EVCP object is no longer supported. Any modules that previously linked in EVCP will now link in RSQL. This replacement will significantly increase the size of the module. You might require increased region, virtual machine, or partition size when running or executing Model 204 jobs.

Accessing the Rocket Model 204 Documentation

As of this release, the Rocket Model 204 documentation is no longer provided on CD. The documentation is available for download from the Rocket M204 website. This enables you to access the latest updated documentation whenever you need it.

To access the Rocket Model 204 documentation:

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

Support for IPv6 addressing

M204 now supports Internet Protocol version 6 (IPv6) addressing. This feature applies to z/OS systems only.

Introduction

Before Version 7 Release 4, all Model 204 communications using TCP/IP used the IPv4 addressing structure, which uses 32-bit network addresses. Version 7 Release 4 adds support for IPv6 addresses, which are 128 bits long.

IPv6 Addresses

Where a dotted-decimal (IPv4) address can be entered, you can now enter an alternative format (IPv6) of 8 16-bit sections separated by colons, such as:

```
AB10:F100:1020:0000:C100:6432:1000:5000
```

The colons indicate that this is a 128-bit address. Both formats are supported.

When an interface (link) is opened, it is by default able to support both IPv6 and IPv4 sockets. A SOCKET call will determine whether a particular socket is AF-INET family (IPv4) or AF-INET6 family (IPv6).

A client can check in advance whether a server is IPv6 or IPv4 by using the getaddrinfo() call.

A server can accept connections from both IPv6 and IPv4. If an IPv6 server gets a connection from an IPv4 client, the IPv6 socket address structure holds the IPv4 client address as an IPv4-mapped IPv6 address.

Special Values

As with IPv4, zero is a special value, meaning ANY. For IPv6, there is a special loopback address ::1, but no special coding is required for this, just as no special code was required for the IPv4 loopback address 127.0.0.1.

A client can use the loopback address of 127.0.0.1 or ::1 for the link's local IP address.

\$SNDMAIL function

The \$SNDMAIL function can handle IPv6 addressing.

Parameters: SMTPCADD, SMTPSADD.

These parameters set the network addresses for the SMTP client and server. The parameters can be passed in 3 formats:

- Symbolic name, such as MYMVS1, or SMTP.MYDOMAIN.COM.
- Dotted decimal address, such as 182.148.55.206.
- IPV6 address, with up to 8 halfwords separated by colons, such as C0:12:46:F0:00:00:00:01, which can also be written as C0:12:46:F0::01. In other words, the double colon shows that there are missing halfwords of binary zeros.

\$SNDMAIL EMAILD structure: The SMTPCADD and SMTPSADD parameters can be overridden on individual \$SNDMAIL function calls by passing an EMAILD structure as the final parameter. This structure is now as follows:

TITLE 'EMAILD - IMAGE WITH SMTP PARAMETERS'			
EMAILD	DS	0F	EMAILD MAPS IMAGE
EMSERVPT	DS	F	SERVER PORT
EMCLNTPT	DS	F	CLIENT PORT
EMSERV	DS	CL255	SERVER NAME
EMCLIENT	DS	CL255	CLIENT NAME
EMSERVDD	DS	CL15	SERVER DOTTED DECIMAL ADDRESS
EMCLNTDD	DS	CL15	CLIENT DOTTED DECIMAL ADDRESS
EMSERVD6	DS	CL45	SERVER IPV6 PRESENTATION FORMAT
EMCLNTD6	DS	CL45	CLIENT IPV6 PRESENTATION FORMAT
EMAILDL	EQU	*-EMAILD	LENGTH OF BLOCK PASSED IN IMAGE

Note the addition of the two new EMSERVD6 and EMCLNTD6 fields for the colon-separated network addresses. These two fields are optional, to provide compatibility with earlier versions.

The following error message has been enhanced to give more detail about \$SNDMAIL errors, both parameter errors and errors resulting from socket calls.

M204.2799: \$SNDMAIL ERROR *errortext*

For details, see message **2799**.

\$SOCKET function

The \$SOCKET function supports IPv6 network addresses.

You can increase the address parameter on the INIT, CONNECT, and ACCEPT calls from 15 to 45 bytes if you are using IPv6 addresses. Specify IPv4 addresses in the dotted decimal format. IPv6 addresses are in the colon-separated format.

The domain parameter on the SOCKET call should be set to 2 for IPv4 addressing or set to 19 for IPv6 (AF_INET6) addressing.

Message changes related to IPv6 addressing

The messages described below have been added or enhanced to support the IPv6 addressing feature.

The new message 2907 has been added:

M204.2907: LINK %C REJECTED ACCEPT. SOCKADDR IS %X

For details, see message **2907**.

The error message

M204.2799: \$SNDMAIL ERROR *errortext*

has been enhanced to give more detail about errors, both parameter errors and errors resulting from socket calls.

For details, see message **2799**.

New MONITOR LINK format

The MONITOR LINK command has been enhanced to accommodate IPv6 addressing.

The new format of MONITOR LINK is:

MONITOR LINK TCPSQL

LOCAL I.D.

::

MAXCON	CONNEC	CONVS	FLGS	TRAN/PROTO	PORT#
49	1	1	A	TCSE/IP	6688

USER	PROCESS	SENDS	RCVS	FLGS	PROCESSGRP	REMOTE I.D.
13	CCARCL	3	4	BI	CTCP10	
	REMOTE ID: ::FFFF:10.204.204.116					

Note

The Local ID above has been placed on a separate line because, if IPv6, it can be up to 45 bytes in length.

On the session detail lines, the remote ID is kept on the one line unless longer than 15 bytes, in which case it is put on a separate line.

In the above example the session partner address for user 13 is an IPv4 application, but it is held as an IPv4-mapped IPv6 address so that it can communicate with the IPv6 host.

New MONITOR PROCESSGROUP format

The MONITOR PROCESSGROUP command has been enhanced to accommodate IPv6 addressing.

MONITOR PROCESSGROUP can now be used with IPv6 network addresses, as shown in the following example:

MONITOR PROCESSGROUP ANY10

REMOTE I.D.

10.0.0.0

LINKNAME	INCNLI	OTCNLI	CONNEC	INCONVS	OTCONVS	FLGS
TCPSQL	4	2	1	1	0	A*

USER	PROCESS	SENDS	RCVS	FLGS	REMOTE ADDRESS	PORT#
41	CCARCL	6	7	BI		16163
	REMOTE ID: ::FFFF:10.204.204.116					

MONITOR PROCESSGROUP ANY10

REMOTE I.D.

10.0.0.0

LINKNAME	INCNLI	OTCNLI	CONNEC	INCONVS	OTCONVS	FLGS
TCPSQL	4	2	1	1	0	A*

USER	PROCESS	SENDS	RCVS	FLGS	REMOTE ADDRESS	PORT#
9	CCARSQ	5	5	BI	10.204.204.116	16136

> (with LOCALID=ANY on DEFINE LINK)

Note

The Remote ID is placed on a separate line because, if it is an IPv6 address, its length can be up to 45 characters. Its place on the detail line has been taken by INCNLI and OTCNLI.

INCNLI is the current inbound conversation limit, as set on the INLIMIT parameter of DEFINE PROCESSGROUP. OTCNLI is the current outbound conversation limit, set on the OUTLIMIT parameter of DEFINE PROCESSGROUP. Both values might have been altered from the original INLIMIT/OUTLIMIT values by MODIFY PROCESSGROUP commands.

IPv4-mapped IPv6 addresses

On a TCP/IP system that has been enabled for IPv6, it is likely that much communication will be between the local IPv6-enabled host and IPv4 partner nodes. If an M204 Online wants to communicate with both IPv4 and IPv6 partners, it does this by using IPv6-family addressing, using IPv4-mapped IPv6 addresses.

For example, CCAMVS1 has an IPv4 address of 192.168.55.206, but an IPv6 address of 2001:21:21:55::206 (a compressed way of notating 2001:0021:0021:0055:0000:0000:0000:0206).

If we use GETADDRINFO to look up CCAMVS1 on a IPv6 system, GETADDRINFO will return the IPv6 address 2001:21:21:55::206 but it is possible also to request the IPv4-mapped IPv6 address ::FFFF:192.168.55.206, which is a compressed way of notating 0000:0000:0000:0000:0000:FFFF:C0A8:37CE, where the IPv4 address is in the last 4 bytes of the 16-byte IPv6 address.

The following scenarios describe connections between the IPv6-enabled system and IPv6 and IPv4 machines.

Scenario 1: M204 server on IPv6-enabled system needs to be available for IPv6 and IPv4 clients

Specify LOCALID=ANY (or LOCALID=:) on the DEFINE LINK. When the Online does a BIND to this address, it will allow connections from both IPv4 and IPv6 clients. There should be a processgroup for incoming IPv4 connections and a separate processgroup for IPv6 connections, both of which can be masked:

```
DEFINE PROCESSGROUP IPV4GP ... REMOTEID=10.24.24.192
MASK=255.255.255.0
DEFINE PROCESSGROUP IPV6GP ... REMOTEID=2001:0021:0024:0C12::2FCE
MASK=FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:0000
```

Scenario 2: M204 client on IPv6-enabled system needs to be connected to IPv6 and IPv4 servers

Specify LOCALID=*symbolic name* or LOCALID=*ipv4-mapped ipv6 address* on the DEFINE LINK.

If you know that all servers you will connect to are IPv6-enabled, you can also use LOCALID=*ipv6 address*. The processgroups should specify the IPv4 address or IPv6 address of the server that the client will connect to:

```
DEFINE PROCESSGROUP IPV4GP ... REMOTEID=10.24.24.192
DEFINE PROCESSGROUP IPV6GP ... REMOTEID=2001:0021:0024:0C12::2FCE
```

\$SOCKETX

This new socket call invokes the GETADDRINFO service using the SOCKCALL GETINFO call. It is currently available for z/OS systems only.

```
$SOCKETX('GETINFO', name, output structure, flag fullword, family)
```

The flag and family fullwords, described in the *Application Programming Interface Reference* manual, have the following values:

```
* FLAG WORD= (FULLWORD) FLAG WORD FOR GETADDRINFO CALL
*   VALID VALUES (ADDITIVE) ARE
* AI_PASSIVE      :    1 - get value suitable for listening
* AI_CANONNAMEOK  :    2
* AI_NUMERICHOST  :    4
* AI_NUMERICSERV  :    8
* AI_V4MAPPED     :   16 - include V4-mapped IPV6 addresses
* AI_ALL          :   32 - get all possible addresses
* AI_ADDRCONFIG   :   64
family fullword: fixed value or literal
*   USED TO LIMIT THE ADDRESSES RETURNED
*   VALID VALUES ARE
*   AI_UNSPEC      :    0 - get both IPV4 and IPV6
*   AI_INET        :    2 - get IPV4 only
*   AI_INET6       :   19 - get IPV6 only
```

The default for both the flags and family fullwords is zero.

The output structure, which can be either variable or image, is

```
ADDRESS1      STRING LEN 45  first address
FAMILY1       STRING LEN 1   4, 6, or space
ADDRESS2      STRING LEN 45  second address
FAMILY2       STRING LEN 1   4, 6, or space
```

With a return code zero result, it is possible that ADDRESS2 and FAMILY2 are both set to spaces if there is only one address returned that fits the supplied criteria.

The following sample procedure illustrates the GETINFO call:

```
BEGIN
DECLARE %PATH1  STRING LEN 80
DECLARE %FAMILY FIXED
DECLARE %FLAGS  FIXED
DECLARE %I      FIXED
%PATH1 = $READ('SYMBOLIC NAME?')
%PATH1 = $UNBLANK(%PATH1)
%FAMILY = $READ('FAMILY - 0 (ANY) 2 (IPV4) OR 19 (IPV6)')
%FLAGS  = $READ('FLAGS - 1 (PASSIVE) 16 (V4MAPPED) 32 (ALL)')
  IMAGE GHOUTPUT
    ARRAY OCCURS 2
      ADDRESS IS STRING LEN 45
      FAMILY  IS STRING LEN 1
    END ARRAY
  END IMAGE
PREPARE IMAGE GHOUTPUT
  DECLARE %RC      FIXED
  %RC = $SOCKETX('GETINFO',%PATH1,'GHOUTPUT',%FLAGS,%FAMILY)
  FOR %I FROM 1 TO 2
    PRINT 'ADDRESS FROM GETINFO ' AND %GHOUTPUT:ADDRESS(%I)
    PRINT 'FAMILY FROM GETINFO ' AND %GHOUTPUT:FAMILY(%I)
  END FOR
  PRINT 'RC ' AND %RC
END
```

which will give output like this:

```

$$SYMBOLIC NAME?
MYHOST
$$FAMILY - 0 (ANY) 2 (IPv4) OR 19 (IPv6)
0
$$FLAGS - 1 (PASSIVE) 16 (V4MAPPED) 32 (ALL)
48
ADDRESS FROM GETINFO 2001:23:20:55::206
FAMILY FROM GETINFO 6
ADDRESS FROM GETINFO ::FFFF:191.28.55.206
FAMILY FROM GETINFO 6
RC 0

```

Note

In the previous example two addresses are returned, because the ALL flag (32) was included, and the second address returned is a v4-mapped IPv6 address.

DEFINE LINK: Horizon for TCP/IP

The DEFINE LINK command for TCP/IP Horizon connections now allows IPv6 network addresses. The LOCALID parameter can, in addition to the supported symbolic and dotted decimal formats, be specified in colon-separated format as described in [\\$SNDMAIL function, on page 14](#).

The new optional LOOKUP parameter is valid for TCP/IP links only. LOOKUP is used when the GETADDRINFO service looks up a supplied symbolic address. Possible values for LOOKUP are:

- IPV6ONLY, only IPv6 and IPv4-mapped IPv6 addresses are returned.
- IPV4ONLY, only IPv4 addresses are returned.
- ANY (default), all matching addresses are returned.

Where more than one address is returned, Horizon uses an IPv4-mapped IPv6 address if it is available.

DEFINE PROCESSGROUP: Horizon for TCP/IP

The REMOTEID and MASK parameters for the DEFINE PROCESSGROUP command can now be specified in a colon-separated IPv6 address format.

You can also specify symbolic remote addresses on the DEFINE PROCESSGROUP command, using REMOTEID=*symbolicaddress*.

If you specify REMOTEID=*symbolicaddress*, you must also enter the new LOOKUP parameter: LOOKUP=IPV6ONLY|IPV4ONLY|ANY.

The three possible values of the LOOKUP parameter affect GETADDRINFO as follows:

- IPV6ONLY– GETADDRINFO will find only IPv6 addresses
- IPV4ONLY– GETADDRINFO will find only IPv4 addresses
- ANY– GETADDRINFO will return any possible address

Do not use the MASK parameter if you are specifying a symbolic address for REMOTEID. MASK is for use only when REMOTEID indicates a range of addresses (with the least significant bits zero where a range of addresses is possible).

Validation of the mask will fail if zero bits in the mask correspond with nonzero bits in the binary remote ID. This is true for IPv6, IPv4, and v4-mapped IPv6 addresses.

Compatibility issues

Compatibility issues with IPv6 addressing arise only if you use symbolic addresses and have those addresses enabled for IPv6.

For example, suppose you have an Online where the DEFINE LINK commands have LOCALID=MYHOST. If MYHOST is enabled for IPv6, the GETADDRINFO service will return an IPv6 address, unless LOOKUP=IPV4ONLY is also specified. A socket will be created with family AF_INET6.

As a server, MYHOST will be able to communicate with both IPv6 and IPv4 clients. IPv4 client addresses will be translated to IPv4-mapped IPv6 addresses.

For example, an IPv4 client called MYCLIENT may have a dotted decimal address 192.206.54.34, which is binary x 'C0CE3622' and can also be expressed as the IPv4-mapped IPv6 address:

```
0000:0000:0000:0000:0000:FFFF:C0CE:3622
```

or

```
:::FFFF:C0CE:3622
```

where the last fullword of the 128-bit IPv6 address contains the 32-bit IPv4 address. The processgroup in this case can specify the REMOTEID and MASK parameters in the IPv4 format.

However, if MYHOST is the client, it cannot connect correctly to an IPv4 client unless LOOKUP=IPV4ONLY is also specified to force GETADDRINFO to return an IPv4 address so that an IPv4 socket will be used for the connection to MYCLIENT.

User Language enhancements

The following topics describe new and enhanced User Language functions and statements in Version 7 Release 4.0.

\$SNDMAIL function

\$SNDMAIL is now supported in BATCH204 if you relink BATCH204 with the following objects after you install Model 204 7.4:

- TC62 (or TC62DOS on z/VSE systems)
- TCIU
- TCSE
- LU62
- FUNR (or FUNRDOS on z/VSE systems)

For more information, see "Enabling \$SNDMAIL" in the *Rocket Model 204 User Language Manual*.

\$SOCKETX function

The new \$SOCKETX GETINFO socket call invokes the GETADDRINFO service to translate a network symbolic address to an IPV4 or IPV6 address. \$SOCKETX GETINFO is available on z/OS systems.

For details, see [\\$SOCKETX](#), on page 17.

\$DEPAD, \$DEPADR, and \$DEPADL functions

The new functions \$DEPAD, \$DEPADR, and \$DEPADL remove characters from a string.

\$DEPAD

Format: \$DEPAD(%string,%remove)

Removes the %remove characters from both sides of %string.

Examples:

```
if %STRING = '___ABC___' then PRINT $DEPAD(%STRING,'_') ---> ABC
if %STRING = 'XYZABCXYZ' then PRINT $DEPAD(%STRING,'XYZ') ---> ABC
IF %STRING = '+A+B+C+' THEN PRINT $DEPAD(%STRING, '+') ---> A+B+C
```

\$DEPADR

Format: \$DEPADR(%string,%remove)

Removes the %remove characters from the right of %string.

Example:

```
if %STRING = '+A+B+C+' then PRINT $DEPADR(%STRING, '+') ---> +A+B+C
```

\$DEPADL

Format: \$DEPADL(%string,%remove)

Removes the %remove characters from the left of %string.

Example:

```
if %STRING = '+A+B+C+' then PRINT $DEPADL(%STRING, '+') ---> A+B+C+
```

External Call Facility (ECF)

EXTERNAL CALL statements can now pass more parameters.

The maximum number of parameters that can be passed in an EXTERNAL CALL statement has been increased from 20 to 40.

The maximum value setting for ECPSIZE is increased from 655340 to 1310680 to accommodate the extra parameters.

Notification of loss of pattern optimization

When pattern optimization cannot be performed with IS LIKE, an informational message is issued to the audit trail.

When a User Language statement specifies the IS LIKE pattern clause, an informational message is issued to the audit trail when pattern optimization cannot be performed. The message notifies you that the entire Ordered Index will be searched:

```
M204.2864: FULL SEARCH OF ORDERED INDEX PERFORMED
```

Pattern optimization cannot be performed if the pattern and any of its members begin with:

- set characters (parentheses)
- range characters (hyphens)
- hexadecimal value 'FF'

- a wild card character (*)
- a placeholder character (+)
- a numeric digit character (#)
- an alphabetic pattern character (@)

Because message M204.2864 is issued at pattern evaluation time and will be issued each time the pattern is evaluated for each segment of the file, you might want to remove the overhead of issuing the message entirely. Issue the following command to minimize the amount of CPU used:

```
MSGCTL M204.2864 NOACTION
```

Expressions in User Language statements

In this release, you can now use expressions for value retrieval in the following User Language statements, as described in this section:

- Update statements
- FILE RECORDS
- FIND
- FIND ALL VALUES
- FIND – POINT\$ condition
- FIND – SFL\$ and SFGE\$ conditions
- FOR EACH RECORD IN ORDER BY
- FOR EACH RECORD WHERE
- FOR EACH VALUE
- FOR RECORD NUMBER

Expression types supported in statements

The following table provides examples of the types of expressions that you can use to supply retrieval values for statements.

Expression type	Example
Function call	IN ?&FILE FD ORDER.DATE EQ VALUE(\$DATE(1, ''))
String concatenation	IN ?&FILE FD BIRTH.DATE EQ VALUE(%YYYY WITH %MM WITH %DD)
Arithmetic operation	IN ?&FILE FD SALES.VALUE EQ VALUE(%ITEM.VALUE * %ITEM.SALES)
User Language construct	IN ?&FILE FD NUM.ORDERS EQ VALUE(COUNT IN FD.ORDERS)
Boolean expression	IN ?&FILE FD ORDER.STATUS EQ VALUE(%STATUS IS LIKE 'A*')

The following example shows these different types of expressions used as values.

Example

```
B
%MM IS STRING LEN 2
%DD IS STRING LEN 2
%BASE.PREMIUM IS FLOAT
%PCT.INCREASE IS FLOAT
*
```

```

* function call expression
*
PRINT 'FD1'
FD1:
IN CLIENTS FPC INCIDENT DATE EQ -
  VALUE($DATECHG('YYYYMMDD',$DATE(1,''),-3169))
  END FIND
FR FD1
  PAI
  PRINT
END FOR
*
* string concatenation expression
*
PRINT 'FD2'
%MM = '09'
%DD = '17'
FD2:
IN CLIENTS FPC ANNIV DATE EQ VALUE(%MM WITH %DD)
  END FIND
FR FD2
  PAI
  PRINT
END FOR
*
* arithmetic operation expression
*
PRINT 'FD3'
%BASE.PREMIUM = 1280
%PCT.INCREASE = 1.25
FD3:
IN CLIENTS FPC TOTAL PREMIUM EQ VALUE(%BASE.PREMIUM * %PCT.INCREASE)
  END FIND
FR FD3
  PAI
FE01:
  FE0 VIN
*
* UL syntax expression
*
  PRINT 'FD4'
FD4:
  IN VEHICLES FPC VIN EQ VALUE(VALUE IN FE01)
  END FIND
  FR FD4
  PAI
  PRINT
  END FOR
END FOR
*
* boolean expression
*
PRINT 'FD5'
FD5:
IN CLIENTS FPC INCIDENT EQ VALUE('T' WITH (%DD IS LIKE '1*'))
  END FIND
FR FD5
  PAI
  PRINT
END FOR
END

```

Expressions as field values within update statements

You can enter an expression as the value of a *fieldname=value* pair in update statements, without assigning the expression result to a %variable or image item.

In previous releases of Model 204, expressions were coded on statements that assigned the expression result to a %variable or image item. Then the %variable or image item could be used in the update statements. This assignment allocates extra, sometimes static, space in NTBL, STBL, and VTBL. With the new support for expressions in update statements, this coding is no longer required.

The following syntax illustrates using an expression on the right side of the equal sign (=) in the following update statements:

```
STORE RECORD fieldname=(expression)
END_STORE

ADD fieldname=(expression)

CHANGE fieldname TO (expression)

CHANGE fieldname = (expression1)TO (expression2)

DELETE fieldname = (expression)

INSERT fieldname = (expression)
```

where:

- *fieldname* specifies the name of the field
- (*expression*), (*expression1*), and (*expression2*) specify the resolved value at the time of evaluation.
- (*expression*), (*expression1*), and (*expression2*) are any expressions of the type listed in [Expressions in User Language statements, on page 22](#).

Note

The (*expression*) on the right side of the equal sign (=) must be enclosed within parentheses. When the field assignment processor sees the open parenthesis, it calls the expression compiler. Without parentheses the field assignment processor treats the value as a literal string.

Example 1

You can issue the following ADD statement and *This**that* is loaded into the field:

```
ADD TEST = ('This' WITH 'that')
```

Example 2


```
B
%REC IS STRING LEN 3
%CT IS FLOAT
%VAL1 IS FLOAT
%VAL2 IS FLOAT
%REC = 'REC'
FOR %CT FROM 1 TO 10
  IN EXPRESS STORE RECORD
    ORD1 = (%REC WITH %CT)
    ORD2 = (%CT * 2)
    ORD4 = (%CT * 4)
  END STORE
  COMMIT
FD1:
  IN EXPRESS FD ORD1 EQ VALUE(%REC WITH %CT)
    END FIND
  FR FD1
    CHANGE ORD2 TO (%CT * 2.1)
    ADD ORD3 = (%CT * 3)
    CHANGE ORD4 = (%CT * 4 ) TO (%CT * 4.1)
    DELETE ORD3 = (%CT * 3)
    INSERT ORD4 = (%CT * 5)
  END FOR
END FOR
END
```

Note

The use of expressions as field values is not supported in:

- Statements inside a PQO UPDATE bracket
 - UPDATE RECORD ... END UPDATE block statements
-

STORE RECORD with a hashed or sorted key

STORE RECORD (*expression*) is also supported for hashed and sorted organization files in order to specify the hash or sort key as an expression.

Example

```
IN TEST1 STORE RECORD (expression)
...
END STORE
```

Where

expression is the hash or sort key.

FILE RECORDS statement

You can use an expression to provide the value in a FILE RECORDS statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FILE RECORDS IN *label* UNDER *fieldname=values* statements

Syntax

The expression is enclosed in parentheses.

```
FILE RECORDS IN label UNDER fieldname = (expression)
```

Where

expression is any expression of the type listed in [Expressions in User Language statements, on page 22](#).

Example

```
B
%REC IS STRING LEN 3
%CT IS FLOAT
%VAL1 IS FLOAT
%VAL2 IS FLOAT
%REC = 'REC'
FOR %CT FROM 1 TO 10
  IN EXPRESS STORE RECORD
    ORD1 = (%REC WITH %CT)
    ORD2 = (%CT * 2)
    ORD4 = (%CT * 4)
  END STORE
  COMMIT
FD1:
  IN EXPRESS FD ORD1 EQ VALUE(%REC WITH %CT)
  END FIND
  FR FD1
    CHANGE ORD2 TO (%CT * 2.1)
    ADD ORD3 = (%CT * 3)
    CHANGE ORD4 = (%CT * 4 ) TO (%CT * 4.1)
    DELETE ORD3 = (%CT * 3)
    INSERT ORD4 = (%CT * 5)
  END FOR
  FILE RECORDS IN FD1 UNDER INVORD5 = (%REC WITH %CT)
END FOR
PRINT 'FRV1'
FRV1:
IN EXPRESS FRV INVORD5
FD2:
  IN EXPRESS FD INVORD5 = VALUE IN FRV1
  END FIND
CT2:
  CT FD2
  PRINT VALUE IN FRV1 AND COUNT IN CT2
END FOR
END
```

FIND Statement EQ VALUE clause

You can search for expressions using the EQ VALUE (*expression*) clause in a FIND statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FIND statements using the EQ VALUE clause

Syntax

The expression is enclosed in parentheses after the EQ VALUE clause.

```
FIELDNAME EQ VALUE(expression)
```

Where

expression is any expression of the types listed in [Expressions in User Language statements, on page 22](#).

Example

```
FD: IN FILE PEOPLE FD
    LAST EQ VALUE($READ('LAST NAME?')) )
```

END FIND

FIND ALL VALUES

You can search for expressions in a FIND ALL VALUES statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FIND ALL VALUES statements

Syntax

The expressions are enclosed in parentheses.

FIND ALL VALUES OF *fieldname* FROM (*expression1*) TO (*expression2*)

Where

expression1 and *expression2* are any expressions of the types listed in [Expressions in User Language statements, on page 22](#).

Example

```
B
%MM IS STRING LEN 2
%DD IS STRING LEN 2
%MM = '09'
%DD1 = '05'
%DD2 = '30'
PRINT 'FR1'
FDV1:
IN CLIENTS FIND ALL VALUES OF ANNIV DATE -
  FROM (%MM WITH %DD1) TO (%MM WITH %DD2)
FD1:
IN CLIENTS FPC ANNIV DATE = VALUE IN FDV1
  END FIND
FR FD1
  PAI
  PRINT
END FOR
END
```

FIND statement – POINT\$ condition

You can use an expression to provide the value in the POINT\$ condition of a FIND statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in the POINT\$ condition of FIND statements

Syntax

The expression is enclosed in parentheses.

POINT\$ VALUE(*expression*)

Where

expression is any expressions of a type listed in [Expressions in User Language statements, on page 22](#).

Example

This example finds the first MA record on file, and then counts the number of records from this point in the file:

```
FD1:
```

```
IN ICSCUST FPC STATE = MA
    END FIND
FOR 1 RECORD IN FD1
    PRINT 'CURREC = ' WITH $CURREC
    PRINT '>= POINT$'
FD2:
    IN ICSCUST FPC POINT$ VALUE($CURREC)
    END FIND
END FOR
```

FIND statement — SFL\$ and SFGE\$ conditions

You can use an expression to provide the value in the SFL\$ and SFGE\$ conditions of a FIND statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in the SFL\$ and SFGE\$ conditions of FIND statements

Syntax

The expression is enclosed in parentheses.

SFL\$ VALUE(*expression*)

SFGE\$ VALUE(*expression*)

Where

expression is any expression of a type listed in [Expressions in User Language statements, on page 22](#).

Example

```
FOR %CT FROM 1 TO 10
    IN EXPRESS STORE RECORD ($EDITN(%CT, '999'))
        ORD1 = (%REC WITH %CT)
        ORD2 = (%CT * 2)
        ORD4 = (%CT * 4)
    END STORE
END FOR
COMMIT
FOR %CT FROM 1 TO 5
    PRINT 'LESS THAN ' WITH %CT
FD1:
    IN EXPRESS FPC SFL$ VALUE($EDITN(%CT, '999'))
    END FIND
    PRINT 'GREATER THAN OR EQUAL TO ' WITH %CT
FD2:
    IN EXPRESS FPC SFGE$ VALUE($EDITN(%CT, '999'))
    END FIND
END FOR
```

FOR EACH RECORD IN ORDER BY

You can use expressions to provide the values in the FROM ... TO clause of a FOR EACH RECORD IN ORDER BY *fieldname* statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FOR EACH RECORD IN ORDER BY *fieldname* statements using the FROM (*expression1*) TO (*expression2*) clause

Syntax

The expressions are enclosed in parentheses.

FOR EACH RECORD IN ORDER BY *fieldname* FROM (*expression1*) TO (*expression2*)

Where

expression1 and *expression2* are any expressions of the types listed in [Expressions in User Language statements, on page 22](#).

Example

```
B
%MM IS STRING LEN 2
%DD IS STRING LEN 2
%MM = '09'
%DD1 = '05'
%DD2 = '30'
PRINT 'FR1'
FR1:
IN CLIENTS FOR EACH RECORD IN ORDER BY ANNIV DATE -
  FROM (%MM WITH %DD1) TO (%MM WITH %DD2)
  PAI
  PRINT
END FOR
END
```

FOR EACH RECORD WHERE

You can search for expressions using the EQ VALUE (*expression*) clause in a FOR EACH RECORD WHERE statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FOR EACH RECORD WHERE statements using the EQ VALUE clause

Syntax

The expression is enclosed in parentheses after the EQ VALUE clause.

FOR EACH RECORD WHERE *fieldname*EQ VALUE (*expression*)

Where

expression is any expression of the types listed in [Expressions in User Language statements, on page 22](#).

Example

```
B
%MM IS STRING LEN 2
%DD IS STRING LEN 2
%MM = '09'
%DD = '17'
PRINT 'FR1'
FR1:
IN CLIENTS FOR EACH RECORD WHERE ANNIV DATE EQ VALUE(%MM WITH %DD)
  PAI
  PRINT
END FOR
END
```

FOR EACH VALUE

You can use expressions to provide the values in a FOR EACH VALUE (FRV) statement.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FOR EACH VALUE statements

Syntax

The expressions are enclosed in parentheses.

FOR EACH VALUE OF *fieldname* FROM (*expression1*) TO (*expression2*)

Where

expression1 and *expression2* are any expressions of the types listed in [Expressions in User Language statements, on page 22](#).

Example

```
B
%MM IS STRING LEN 2
%DD IS STRING LEN 2
%MM = '09'
%DD1 = '05'
%DD2 = '30'
PRINT 'FR1'
FRV1:
IN CLIENTS FOR EACH VALUE OF ANNIV DATE -
  FROM (%MM WITH %DD1) TO (%MM WITH %DD2)
  PRINT VALUE IN FRV1
END FOR
END
```

FOR RECORD NUMBER

The FOR RECORD NUMBER statement (abbreviation: FRN) accepts an expression to supply its value.

Status

Updated in version 7.4.0.

Purpose

Provides support for the general use of expressions in FOR RECORD NUMBER statements

Syntax

FRN *expression*

A multi-part expression must be enclosed in parentheses; for example:

FRN (%A + %B)

Where

expression is any expression of a type listed in [Expressions in User Language statements, on page 22](#).

Example

```
IN FILE CLAIMS STORE RECORD
...
END STORE
IN FILE CLAIMS FRN $CURREC
...
END FOR
```

AUDIT ALL INFORMATION statement

You can write the output of entire records to the audit trail, CCAUDIT, with AUDIT ALL INFORMATION. The output is identical to the output from PRINT ALL INFORMATION (PAI) but each field=value pair is sent to CCAUDIT as a series of US lines.

Status

New in version 7.4.0.

Purpose

The AUDIT ALL INFORMATION (or AAI) statement writes the output of entire records to CCAUDIT.

This statement occurs in a FOR EACH RECORD or FOR RECORD NUMBER loop.

Syntax

```
AUDIT ALL INFORMATION
```

```
or  
AAI
```

STORE RECORD block

The STORE RECORD block supports a THEN CONTINUE statement, allowing for the conditional building of a Model 204 record.

Status

Updated in version 7.4.0.

Purpose

To store data in a Model 204 file.

Syntax

```
STORE RECORD  
  fieldname1=[ value1 | (expression1) ]  
  [fieldname2=(expression)]  
  ...  
[THEN CONTINUE  
  statement  
  statement  
  ...]  
END STORE
```

The statements following the THEN CONTINUE statement of the STORE RECORD block operate as if they were coded within a FRN \$CURREC block, which immediately follows the END STORE statement. This construct is easier for coding because you do not need to again specify the file specification of the STORE RECORD statement. It is also more efficient because an actual FRN statement is not necessary.

Example

The following basic syntax example illustrates that you can use any intervening statements after THEN CONTINUE and before END STORE.

```
%COLOR = 'BLUE'  
STORE RECORD  
  MODEL = %MODEL  
  THEN CONTINUE  
  FR WHERE RECTYPE = 'TABLE' AND COLOR = %COLOR  
    %CODE = COLOR_CODE  
  END FOR  
  ADD COLOR_CODE=%CODE  
  PAI  
END STORE
```

Support for nested STOREs

The THEN CONTINUE block allows for the coding of a nested STORE..END STORE block within the body of the outer STORE, so that related records may be built together. The nested STOREs can refer to a different file context, without compromising the file context of the outer STORE. The following example stores an order header record along with an order line record.

```
IN ORDHDR STORE RECORD
  ORDER_NUMBER = 1000568
  CUSTOMER_NUMBER = 111456
THEN CONTINUE
  IN ORDLIN STORE RECORD
    ORDER_NUMBER = 1000568
    ITEM_ID = F004
    ITEM_QTY = 3
  END STORE
  ADD ORDER_STATUS = A
END STORE
...

```

The results of this would be the following record stored in the ORDHDR file:

```
ORDER_NUMBER = 1000568
CUSTOMER_NUMBER = 111456
ORDER_STATUS = A

```

and the following record stored in the ORDLIN file:

```
ORDER_NUMBER = 1000568
ITEM_ID = F004
ITEM_QTY = 3

```

Support for multiply occurring fields

In the following example a For Each Occurrence loop is driven, based on occurrences of the field SALES_MM previously stored, to store occurrences of MONTHLY_TOTAL:

```
B
%MONTHLY_SALES IS FLOAT ARRAY (3)
%MONTHLY_SALES(1) = 10
%MONTHLY_SALES(2) = 15
%MONTHLY_SALES(3) = 35
IN SALES STORE RECORD
  RECTYPE = TOT_SALES
  SALES_MM = '01'
  SALES_MM = '02'
  SALES_MM = '03'
THEN CONTINUE
FEO_SALES:
  FEO_SALES_MM
  ADD MONTHLY_TOTAL = %MONTHLY_SALES(OCCURRENCE IN FEO_SALES)
END FOR
END STORE
END

```

The resultant record in the SALES file is:


```

RECTYPE = TOT_SALES
SALES_MM = 01
SALES_MM = 02
SALES_MM = 03
MONTHLY_TOTAL = 10
MONTHLY_TOTAL = 15
MONTHLY_TOTAL = 35

```

Support for COMMIT and BACKOUT

The COMMIT and BACKOUT statements can be used following a THEN CONTINUE statement to save parts of a record as it is built, and to back out all of parts of a record conditionally. In the following example:

```

.....
%CUSTNO = '100639'
IN ORDERS STORE RECORD
  RECTYPE = ORDER
  ORDER_NUMBER = 1000234
  CUSTOMER_NUMBER = %CUSTNO
THEN CONTINUE
  COMMIT /* save the order header */
FIND_CUST:
  IN CLIENTS FD RECTYPE = POLICYHOLDER
    POLICY NO = %CUSTNO
  END FIND
  FOR 1 RECORD IN FIND_CUST
    %ADDRESS = ADDRESS
    %CITY    = CITY
  END FOR
  ADD ADDRESS = %ADDRESS
  ADD CITY    = %CITY
  COMMIT /* Save the customer address */
  ADD DELIV_DATE = ($DATECHG('YYYYMMDD',$DATE(1,''),10))
  IF %ORDER_DELAYED = 'Y' THEN
    BACKOUT /* Back out deliv date if delay detected */
  END IF
END STORE
.....

```

if %ORDER_DELAYED is not 'Y' then the record is stored as follows:

```

RECTYPE = ORDER
ORDER_NUMBER = 1000234
CUSTOMER_NUMBER = 100639
ADDRESS = 0880 HANCOCK STREET
CITY = LANCASTER
DELIV_DATE = 20111228

```

otherwise the DELIV_DATE f-v pair is backed out.

Known restrictions or limitations

- Be cautious of using the JUMP TO statement following THEN CONTINUE to jump to a label outside the STORE..END STORE block, as this might lead to the storing of a partial record.
- It is possible to call a subroutine after the THEN CONTINUE statement, as you might in a FOR RECORD NUMBER loop. Additional update statements to the current record are allowed in the subroutine but only in a FRN \$CURREC loop. Otherwise, record context is not established and any additional updating statements within the subroutine would be rejected with the following compilation error:

```
M204.0229: INVALID STATEMENT
```

- A DELETE RECORD statement following THEN CONTINUE, but before END STORE, will cause the current record context to be lost. Any further update statements will cause the request to be canceled with one of the following messages:

M204.1233: DFAV, BAD RECORD NUMBER *n* FOR FILE *filename*

M204.1266: NONEXISTENT RECORD REFERENCED - *n* IN FILE DSNLIST

Note

When using THEN CONTINUE, keep in mind standard considerations for coding any update unit. Be aware that creating longer update units has implications for resource sharing, checkpoints, and recovery requirements.

ANDIF and ORIF short-circuit operators

The operators ANDIF and ORIF provide short-circuit logical operations.

The purpose of short-circuit operators is to enable the compiled code to bypass the evaluation of a second expression, depending on the result of the previous expression. If the result of the first expression can determine the final result of the operation, there is no need to continue evaluating subsequent expressions because they will not change the outcome.

Consider using short-circuit operators if your code can skip evaluation of other expressions, especially if they are complex expressions. Any advantages of using short-circuit operators instead of nested IFs will depend on your code. In most cases, standard nested IF processing will provide better performance.

Using the ANDIF operator

Consider the following code that must evaluate each EQ expression before taking action:

```
IF LASTNAME EQ 'SMITH' AND -  
  FIRSTNAME EQ 'SAM' AND -  
  BIRTH.YEAR EQ 1963 THEN  
  * do something  
END IF
```

You could rewrite the previous code to prevent evaluation of some conditions if the higher level conditions are false. For example, if LASTNAME is not equal to 'SMITH', the subsequent conditions on FIRSTNAME and BIRTH.YEAR are not evaluated:

```
IF LASTNAME EQ 'SMITH' THEN  
  IF FIRSTNAME EQ 'SAM' THEN  
    IF BIRTH.YEAR EQ 1963 THEN  
      * do something  
    END IF  
  END IF  
END IF
```

You could also accomplish the same thing without nested IF statements by using ANDIF. For example, if LASTNAME is not equal to 'SMITH', the subsequent conditions are not evaluated:

```
IF LASTNAME EQ 'SMITH' ANDIF -  
  FIRSTNAME EQ 'SAM' ANDIF -  
  BIRTH.YEAR EQ 1963 THEN  
  * do something  
END IF
```

Comparing ANDIF evaluation sequence

Comparing ANDIF evaluation sequence illustrates how sequential expressions are evaluated when using the ANDIF operator and the result of the operations:

Expression 1	Expression 2	Expression 3	Result
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	(Not evaluated)	FALSE
FALSE	(Not evaluated)	(Not evaluated)	FALSE

Using the ORIF operator

Similarly, with the ORIF operator:

```
IF LASTNAME EQ 'SMITH' ORIF -
  LASTNAME EQ 'JONES' ORIF -
    FIRSTNAME EQ 'FRED' THEN
  * do something
END IF
```

Comparing ORIF evaluation sequence

Comparing ORIF evaluation sequence illustrates how subsequent expressions are evaluated using the ORIF operator and the result of the operations:

Expression 1	Expression 2	Expression 3	Result
TRUE	(Not evaluated)	(Not evaluated)	TRUE
FALSE	TRUE	(Not evaluated)	TRUE
FALSE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE

Note

You can intersperse ANDIF evaluations and ORIF evaluations with AND evaluations and OR evaluations.

Operating system and peripherals support

The following topics describe operating system or peripherals enhancements in Model 204 version 7, release 4.

XDM improved ALET (Access List Entry Token) capacity

The XDM (Cross-memory Data Mover) has been modified to allow the full use of ALETs up to the IBM limit of 512.

This allows you to support more concurrent connections between multiple ONLINEs and multiple XDM clients connecting to those ONLINEs (CICS tasks and batch IFAM tasks).

For example: In previous releases, a single CICS region that communicated to 3 ONLINE regions would need 3 ALETs, one for each ONLINE. In this release, each CICS region may talk to multiple ONLINEs using one ALET. So for 3 CICS regions to communicate to 3 ONLINE regions, a total of 6 ALETs is needed – one ALET for each CICS region and one for each Model 204 ONLINE.

Extended Address Volumes (EAV)

Model 204 v7.4 supports database and sequential files on EAV above the 64K cylinder boundary. This feature requires IBM z/OS version 1.12 or later.

Extended Address Volumes (EAV) are volumes with more than 64K cylinders.

Users of z/OS version 1.12 and later can allocate and use Model 204 database files and checkpoint, CCAJRN, CCAJLOG, and USE data sets on EAV volumes above the 64K cylinder boundary. Both IOS Branch Entry (XMEMOPT=2) and EXCP I/O drivers can handle database files on EAV.

To allocate a data set on an EAV volume, use the new EATTR parameter of the ALLOCATE command. See [ALLOCATE](#), on page 68.

The OPSYS parameter value X'000040' specifies that EAV volumes are supported. See [OPSYS](#), on page 59.

Note

Server data sets are not supported when allocated above the 64K cylinder boundary. An attempt to open any server data set allocated above 64K cylinders results in issuing error message **2917**, which identifies the server data set name, and run termination.

Performance enhancements

Performance enhancements in this release include 1 MB virtual pages and related statistics as well as improved SORT processing.

Model 204 zIIP support

On a z/OS system, Model 204 zIIP support enables you to offload Model 204 work from regular processors to zIIP processors.

zIIP support in Model 204

Model 204 can run a portion of workload on IBM System z Integrated Information Processor (zIIP) engines. The only unit of work that z/OS schedules to run on a zIIP engine is the enclave Service Request Block (SRB). An enclave is a collection of resources, such as address spaces, TCBs, and SRBs, that are controlled by Workload Manager (WLM) as an entity. Therefore, an enclave SRB is a part of an enclave.

Model 204 creates an enclave SRB with the same dispatching priorities as the maintask. Model 204 uses enclave SRB, in this case zIIP, in the same way as an MP subtask. The term zIIP subtask is the equivalent of enclave SRB.

Note

MP subtasks and zIIP subtasks may be used together. Therefore, AMPSUBS, NMPSUBS, AMPSUBZ, and NMPSUBZ may all be greater than zero. However, to avoid unnecessary overhead:

- Do not set AMPSUBS greater than the number of central processors actually available.
- Do not set AMPSUBZ greater than the number of zIIP processors actually available.

When using zIIP processor offloading, IOS Branch Entry must be enabled (XMEMOPT=2). If you are also using XDM-CRAM, set XMEMOPT=82. The cross-memory environment will be initialized as part of XDM and IOS initialization.

Model 204 workload eligible for zIIP offload

Components of Model 204 that are eligible for zIIP processing include:

- FIND statement for files and groups
- SORT statement, when there are more than 64 records
- APSY load
- Server swapping
- Screen definition statements
- Menu definition statements
- FOR and REPEAT loops, if they do not:
 - perform updates
 - issue READ SCREEN statements
 - issue MQ calls (for example, MQGET)
 - issue calls to mathematical functions (for example, \$ARC)
 - read or write external, sequential files
 - use non-Model 204 supplied \$functions or third-party software

Model 204 zIIP infrastructure created during initialization

The NMPSUBZ parameter initiates a zIIP infrastructure.

If NMPSUBZ is greater than zero, then that number of zIIP subtasks is created at initialization. Each zIIP subtask is an enclave SRB.

AMPSUBZ specifies the number of active zIIP subtasks and can be reset during the run.

Model 204 parameters for zIIP support

XMEMOPT

When you are using zIIP processor offloading, IOS Branch Entry must be enabled (XMEMOPT=2). If you are also using XDM-CRAM, set XMEMOPT=82. The cross-memory environment will be initialized as part of XDM and IOS initialization.

New and updated parameters

This release provides new and updated parameters for zIIP support that were modeled from their MP counterparts.

New parameters:

- AMPSUBZ: Specifies the number of active zIIP subtasks
- NMPSUBZ: Specifies the number of zIIP subtasks attached during initialization (enclave SRBs)
- SCHDOFLZ: Adjusts the scheduling of zIIP subtask activation
- SCHDOPT1: Allows server swapping by an MP subtask
- ZQMAX: Limits number of work units for execution by zIIP subtask (new)

Updated parameter:

- SCHDOPT: New values support zIIP processing

SCHDOPT

The following SCHDOPT values are related to zIIP support:

- (existing) X'08': Do server I/O on MP subtask.
- (new) X'20': Do not let the maintask pick up zIIP workload, even if no other work is available.
- (new) X'40': Do not let an MP subtask pick up zIIP workload, even if no other work is available.
- (new) X'80': Allow zIIP to do server swapping. If you have SCHDOPT=80 set, then you probably want to set SCHDOPT1=01 as described in the "Usage Notes" below.

The SCHDOPT parameter setting of X'80' allows server swapping to be offloaded to zIIP processors.

The X'80' setting is only valid when CCASERVER is in memory (servers swapped into memory). If you set SCHDOPT=X'80' when CCASERVER is not in memory, the setting X'80' is reset and message 2914 is issued:

SCHDOPT INDICATION OF SERVER SWAPPING DONE BY ZIIP IS ONLY VALID WHEN CCASERVER IS IN MEMORY

Initialization then continues.

Usage Notes

- SCHDOPT=08: An MP subtask will always execute a swap unit of work before a user unit of work. This could mean that every MP subtask is busy swapping.
- SCHDOPT=80: The zIIP subtask is doing the swap work and the MP subtasks will do real user work.
- SCHDOPT=88: The zIIP subtask is behaving exactly as an MP subtask. Both zIIP and MP subtasks could be busy doing swapping while no work is actually being done.
- SCHDOPT=80 and SCHDOPT1=01: The zIIP subtask will do the swapping before doing any work and the MP subtasks will do real user work. But if there is no user work to do, the MP subtask will help the zIIP subtask by executing a swap unit of work. That way, if the zIIP subtask gets behind on its swapping, then the MP subtask can assist.

SCHDOPT1

The new parameter SCHDOPT1 uses an x'01' value. SCHDOPT1=x'01' allows an MP subtask to handle server swapping if the subtask has nothing else to do.

SCHDOPT1 is useful when zIIP is handling server swapping and needs help from an MP subtask.

ZQMAX: Limits number of work units for execution by zIIP subtask

The ZQMAX parameter limits the number of work units that may be placed for execution by the zIIP subtask.

The default and maximum value for ZQMAX is 65535.

- If ZQMAX is set to a non-zero value:
When a work unit is pushed onto the zIIP queue and the current zIIP queue length exceeds the value of ZQMAX, then that work unit is placed on the MP stack instead. Effectively, this limits the amount of work that is done on the zIIP and prevents zIIP overload.
- If ZQMAX is set to zero:
No users will be added to the zIIP queue. However, if SCHDOPT-X'80' is on and a zIIP subtask is active (AMPSUBZ>0), then server swapping will occur on the zIIP subtask without the overhead of pushing zIIP work units onto the zIIP queue.

Model 204 zIIP-related statistics and monitoring

The MONITOR TASKS command indicates a zIIP subtask with a lowercase letter z next to a subtask number. All information returned by the MONITOR command for the zIIP subtask includes accumulated data for the zIIP subtask and corresponding zIIP subtask. The following system statistics are new:

- CPUONZIP specifies the total CPU time in milliseconds used by all zIIP engines
- CPUTOTZE specifies the total CPU time in milliseconds used by workload eligible for offloading to a zIIP

Note

If CPUTOTZE is considerably larger than CPUONZIP, then you should define more zIIP subtasks.

zIIP utilization

You can control zIIP utilization by setting the IIPHONORPRIORITY parameter.

The z/OS parameter IIPHONORPRIORITY, which defaults to YES, honors the existing task processing priority. This means that a zIIP engine executes workload only when all regular engines are busy.

If zIIP usage is more important to you than regular CPU usage, then you should set IIPHONORPRIORITY=NO and the Model 204 SCHDOPT parameter should include X'20' and X'40' values. See [SCHDOPT](#), on page 62 for details.

zIIP-compatible functions

All Rocket Software supplied User Language \$functions are zIIP compatible.

New and enhanced zIIP statistics

The MONITOR STAT command has been enhanced to provide zIIP processing statistics.

MONITOR STAT output displays the following new statistics:

Statistic	Total CPU time in milliseconds used by...
CPUTOTZE	workload eligible for offloading to a zIIP engine
CPUONZIP	all zIIP engines
Statistic	Number of times that a unit of work was taken by...
MTDEQ	The maintask from the maintask queue
MTSDEQ	The maintask from an MP subtask queue
MTZDEQ	The maintask from a zIIP subtask queue
STDEQ	An MP subtask from an MP subtask queue
STZDEQ	An MP subtask from a zIIP subtask queue
ZTDEQ	A zIIP subtask from a zIIP subtask queue

MTZDEQ and STZDEQ show the number of times that the maintask or an MP subtask took work that was allocated to a zIIP subtask.

The maintask can take work from an MP subtask queue or a zIIP subtask queue. An MP subtask can take work from a zIIP subtask queue. A zIIP subtask can only take work from a zIIP subtask queue.

To see non-zero results for MTZDEQ or STZDEQ, you must turn off SCHDOPT x'20' and SCHDOPT x'40'. If SCHDOPT x'20' is on then the maintask is not allowed to take zIIP work. Likewise, SCHDOPT x'40' prevents the MP subtask from taking work from zIIP. So, if either bit is on, the corresponding statistic will always be zero.

A MONITOR TASKS command displays three new columns visible on screens formatted as 3270 Model 5 terminal:

- ZDEQ, which represents the number of units of work taken from a zIIP queue by either the maintask or an MP subtask. This column has meaning only for the maintask and MP subtasks. For zIIP subtasks this column always contains a zero.
- SDEQ, which represents the number of units of work taken from an MP subtask queue by the maintask. This column has meaning only for the maintask; for other tasks this column always contains a zero.
- ZIPCPU, which represents the total CPU time (in milliseconds) consumed by this zip engine. It is zero for MP subtasks and greater than zero when "z" precedes the task number and that zIIP subtask has been active (AMPSUBZ>0) at some point during the run.

Examples

Example 1 shows the results of the MONITOR STAT command. The zIIP statistics are bolded here for visibility.

```
AUDIT=16640 OUT=1225 IN=124 OUTXX=6836 INXX=182 DEV5=97 DEV6=581 DEV7=-
174 DEV8=30 WAIT=160665 DKRD=9078 DKWR=108152 SVRD=3766 SVWR=3775 CPU=44-
28 REQ=18 MOVE=117713 CNCT=556 SWT=19 RECADD=6362 RECDEL=10588 BADD=1443-
47 BDEL=288 BCHG=67379 IXADD=16116 IXDEL=31854 FINDS=661 RECDS=78622 DKA-
R=35853 DKPR=2579487 TFMX=387 USMX=13 SVMX=6 SFTRSTRT=1 DKPRF=864643 SMP-
LS=30 USRS=12933 SVAC=6000 BLKI=6000 PCPU=562 DIRRC=29040 STCPU=3186 ST-
DEQ=23899 STWAIT=30743 STPOST=23952 LKWAIT=24 LKPOST=24 STIMERS=3831 SVPA-
GES=311303 COMMITS=3599 BACKOUTS=9 UBUFHWS=129448 TSMX=386 DKRDL=89 DKWRL-
=9611 CPUTOTZE=5 CPUONZIP=5 MTDEQ=116771 MTZDEQ=2082 STZDEQ=506 ZTDEQ=96
DKSRR=6792 DKSAWW=296 DKSRRFND=1921 DKSTBLF=3621 DKSTBLA=8 DKSTBLB=4874 -
DKSTBLC=3033 DKSTBLD=1725 DKSTBLX=48 DKSTBLE=13286
```

Example 2 shows output from a MONITOR TASKS command from the same run, using a Model 5 terminal:

TASK	CPU	PR	PCPU	WAIT	POST	LKWAIT	LKPOST	MQWTM	DEQ	ZDEQ	SDEQ	LKWTIME	LKPTIME	ZIPCPU
0	1261	4337	290	163623	31854	12	12	1	119078	2085	0	0	0	0
1	3253	3628	896	31228	24302	12	12	11	24237	505	0	0	0	0
2	3	49	61	23	14	0	0	2294	34	1	0	0	0	0
z	3	5	3	1666	603	68	0	0	229	96	0	0	0	5

1 MB virtual pages

This release supports 1 MB virtual pages on z/OS.

Model 204 now supports 1 MB virtual pages in this environment:

- CPU: model z/10 and later
- O/S: z/OS 1.9 and later

1 MB pages can be used only with storage above the bar (2 GB).

Use of 1 MB pages provides a performance advantage because the operating system can now keep more real addresses in memory and minimize the translation of virtual to real addresses.

To use 1 MB pages, set the new Model 204 parameter, ZPAGEOPT, to indicate which areas should be allocated using 1 MB pages.

The size of the virtual storage area with 1 MB pages for an LPAR is defined at IPL time by the z/OS LFAREA (Large Frame AREA) parameter, set in SYS1.PARMLIB(IEASYSnn). This storage is not pageable.

Please consult your systems programmer to:

- check on the setting of the LFAREA parameter, and

- confirm that your operating system has all the latest maintenance from IBM regarding 1 MB pages.

The LFAREA is dynamic and shared by all regions. When z/OS experiences a shortage of 4 K pages, it steals 1 MB pages and subdivides them into 4 K pages. If the Model 204 request for virtual storage with 1 MB pages cannot be satisfied, then regular 4 K pages will be used for this purpose, without generating a warning.

When buffers are allocated with 1 MB pages, Model 204 must use IOS Branch Entry (XMEMOPT=2). EXCP and ECPVR do not support buffers allocated with 1 MB pages above the bar.

Parameters

The ZPAGEOPT parameter indicates which areas should be allocated by using 1 MB pages. For more information, see [ZPAGEOPT, on page 68](#).

Two new view-only parameters have also been added:

- **STORZPAG - CURRENT MBYTES STORAGE ALLOCATED WITH 1 MB PAGES**
The current number of megabytes of above-the-bar storage allocated with 1 MB pages.
- **STORMXZP - HWM MBYTES STORAGE ALLOCATED WITH 1 MB PAGES**
The high water mark of megabytes of above-the-bar storage allocated with 1 MB pages during the run.

The OPSYS parameter X'000020' value specifies that 1 MB pages are supported. See [OPSYS, on page 59](#).

MONITOR GSTORAGE command

The new MONITOR GSTORAGE command displays statistics regarding allocated above-the-bar storage. For more information, see [MONITOR GSTORAGE, on page 71](#).

Messages

Two new messages are associated with 1 MB pages and above-the-bar storage:

- **M204.2919:** HWM MBYTES ATB STORAGE ALLOCATED = %C
- **M204.2925:** HWM MBYTES STORAGE ALLOCATED WITH 1 MB PAGES = %C

FTBL and server storage above the bar

FTBL can now reside in non-swappable 64-bit storage (above the bar).

Note

This feature is available when Model 204 runs on IBM z/OS or IBM z/VM systems and storage above the bar is available.

64-bit support on IBM z/VSE systems will be available in Summer 2012.

Description

Version 7 Release 4 allows you to allocate designated server tables for each user in storage above the bar that will not be subject to server swapping. This feature enables you to divide the server storage into two parts: swappable and non-swappable. This division makes the swappable part of the server storage smaller and faster to swap.

Parameters to implement FTBL above the bar

There are two new parameters that control non-swappable server areas: `SERVNSSZ` and `SERVNSA`.

- `SERVNSSZ` (server non-swappable size) is the amount of space in bytes required for the above-the-bar server tables per user. This size is rounded up to the nearest 4 K. The total amount of storage allocated for non-swappable server areas equals `SERVNSSZ` multiplied by `NUSERS`.

When sizing `SERVNSSZ` you should use the largest FTBL sizes that might be needed.

- `SERVNSA` (server non-swappable areas) indicates the tables that you want to be above the bar. At this time, only FTBL may be selected and its appropriate setting is `x'02000000'`.

The non-swappable server area can be used with server swapping done either in storage or on disk. The non-swappable server area can also be used when there is no server swapping (`NUSERS=NSERVS`). Loading the non-swappable server area above the bar enables you to make servers below the bar smaller, thus saving storage below the bar.

To obtain below-the-bar storage savings when the non-swappable server area is used, decrease the value of the `SERVSIZE` parameter by the `LFTBL` value used in calculating the server size. For details about `LFTBL`, server sizing, and server storage, see the *Rocket Model 204 System Manager's Guide*.

Note

When FTBL is placed in the non-swappable server area above the bar, then the `UTABLE` command decreasing FTBL size will not free any storage in the regular server below the bar.

Usage Notes

If you want to use FTBL above the bar, you need to find out whether your \$functions or third party software reference FTBL structures. You can verify this by checking your code for these names: `KUPFT` (FTBL address) and `OGTD` (FTBL entry map). If there are no references to those names, then most likely your code does not address FTBL and no changes are needed. If your code addresses FTBL, then you must change the code to be 64-bit complaint.

Messages

Several new messages are associated with FTBL above-the-bar storage:

- **M204.2916:** `SERVNSSZ` REQUIRES AT LEAST ONE AREA TO BE SET IN `SERVNSA`
- **M204.2920:** Non zero `SERVNSSZ` is required
- **M204.2921:** Non swappable server areas are not supported in VSE
- **M204.2922:** `SERVNSA` bit setting is not valid
- **M204.2923:** Space required for non swappable server areas is larger than `SERVNSSZ`
M204.2923 is issued when the space problem occurs during initialization. M204.2923 sets the return code to 80, appears in the job log, and results in the Online being immediately terminated.
- **M204.2932:** Space required for non swappable server areas is larger than `SERVNSSZ`
M204.2932 is issued when the space problem is detected after initialization is complete, most likely during `UTABLE` command processing. It does not appear in the job log and does not result in the Online being terminated.

EBM and procedure pages allowed above the bar

Existence Bit Map (EBM) pages and procedure pages are now eligible to be placed in above-the-bar storage.

- Existence Bit Map (EBM) pages can now reside in the above-the-bar buffer pool. (Each Model 204 file contains one EBM page for each segment in a file. If a file has five segments that means there are five EBM pages for that file.)
- Procedure pages can now reside in the above-the-bar buffer pool. Pages from the procedure dictionary are still read into the below-the-bar buffer pool.

You can use the NUMBUG and MAXBUF parameters to make adjustments to the above-the-bar and below-the-bar buffer pools, respectively. See the *Model 204 System Manager's Guide* for details.

Coded and binary values no longer converted to string during SORT

Fields with the CODED or BINARY attributes are now left as such in sort work records.

Leaving fields as CODED or BINARY reduces the size of sort work records and also reduces CPU and CCATEMP I/O during SORT processing.

Chapter 2: Additional features

Topics covered in this chapter

Utilities enhancements

Files

Recovery, Reliability, and CCAJRNL

Model 204 SQL Catalog enhancements

MQ/204

Other features

This chapter describes additional features provided in this release.

Utilities enhancements

The following topic describes enhancements to utilities in version 7.4.0.

UTILJ enhancements

You can now use FILENAME= to select journal records.

You can now specify FILENAME=*filename* in the option list for UTILJ if a CCAOUT DD card is present. This option will select all journal records that are associated with *filename* for copying into the CCAOUT DD. In this way, you can extract a single file's worth of entries from a larger journal. This option applies to z/VM and z/OS systems only.

Files

Date/time stamp (DTS) file update under IFAM

IFAM applications can now open a date/time stamp (DTS) file and execute all update functions on the file.

Creating and using in-memory Model 204 files

You can create temporary in-memory Model 204 files, which incur less CPU overhead and none of the I/O associated with permanent Model 204 files. This feature is not supported on z/VSE systems.

If you need a temporary Model 204 file, you no longer have to allocate it on disk. It can be built in memory, ATB (above the bar), using the MEMORY PAGES keywords in the DEFINE and ALLOCATE commands. You can reduce CPU and eliminate file I/O when you create in-memory files. The following scenarios show how you might use in-memory files:

- You have a WORK file that applications use. If this is an in-memory file, all file read (DKRD) and write (DKWR) operations merely move pages from the in-memory file to and from the buffer pool for processing just like any other file, but there is no physical disk I/O. Any updated pages will be moved (DKWR) back to the in-memory file, but that also avoids physical disk I/O.
- You have a file that gets an extremely high number of updates. That file can be allocated as an in-memory file and RESTOREd from a backup copy. Then it can be used by any update

processes needed, and when complete, the file can be closed by all users and then DUMPed to the backup data set. This can often require less overall I/O and CPU than if the file had been processed as a regular Model 204 file on disk.

Use the DEFINE DATASET and ALLOCATE commands to reserve space for a temporary in-memory Model 204 file; then create the file:

```
ALLOCATE TEMPFILE WITH MEMORY PAGES 10000
CREATE TEMPFILE
PARAMETER ...
END CREATE
```

or

```
DEFINE DATASET TEMPFILE WITH SCOPE=SYSTEM MEMORY PAGES 15000
CREATE TEMPFILE
PARAMETER ...
END CREATE
```

All pages for an in-memory file reside in and use system memory (above the bar) while still taking advantage of the structure and capabilities of a Model 204 file. Using an in-memory file reduces CPU time processing, such as COMMIT statements and file page loading and swapping, as the file pages remain in storage even after the last physical file close. All in-memory file pages are shared among all users via the buffer pool, like any other Model 204 file.

In-memory files are created with a default of FRCVOPT=X'24':

```
FRCVOPT  X'24'          (NO CP LOGGING,NO RF LOGGING) FILE RECOVERY OPTIONS
```

However, remember that RESTOREing an in-memory file from a backup will copy the FRCVOPT from that backup copy into the in-memory file, thus overriding the default. In-memory files are also created with FOPT=0, which is TBO enabled, and TBO is supported. REGENERATE is also supported. RESTART recovery is not supported.

Given these caveats, if you want to provide for both REGENERATE recovery and TBO, the best parameter settings for in-memory files are FRCVOPT=X'20' and FOPT=0.

Freeing an in-memory file

When you no longer require an in-memory file, you can free the file.

If you know that you will need the file's data in the future, DUMP the file or otherwise copy the data first.

To free a MEMORY PAGES file, and release the memory associated with it, issue this command:

```
FREE filename
```

After the file has been freed, the data is no longer available and the memory is reused by the address space.

Viewing in-memory file storage consumption

To see how much storage an in-memory file is consuming, issue the MONITOR GS (or M GS) command to see information similar to the following example:

```
11.271 SEP 28 13.30.32          PAGE 1
NAME          START              SIZE (MB)  1M SIZE  PGFIX
CCASERV      00000048_04100000          65
CCATMP       00000048_0F800000         118
CCATBCD      00000048_16F00000          1
BUFFHASH     00000048_17700000          7
BUFFCNTL     00000048_18E00000         16
CCAAPSY      00000048_64000000          4
TEMPFIL2     00000048_66E00000         30    <-- (an in-memory file allocated with PAGES 5000)
BUFFERS      00000048_1B100000        591
Total amount of storage allocated above the bar:          832
Total amount of storage backed by 1 MB pages   :           0
Total amount of storage page fixed             :           0
```

DEFINE DATASET and ALLOCATE commands support for in-memory files

The DEFINE DATASET and ALLOCATE commands support temporary in-memory files with the corequisite keywords: MEMORY and PAGES.

- The MEMORY keyword designates an in-memory file, specifically an above-the-bar memory object.
- The PAGES keyword specifies the number of Model 204 (6K) pages to allocate for the file.

For example, the following command allocates an in-memory file named BUSES:

```
ALLOCATE BUSES WITH MEMORY PAGES=10000
```

The following command defines an in-memory file named BUSES:

```
DEFINE DATASET BUSES WITH SCOPE=SYSTEM MEMORY PAGES=10000 DIRECT
```

The difference between using the ALLOCATE or DEFINE DATASET command for in-memory files is:

- For a DEFINE DATASET command the memory object is not allocated until the file is first opened or created.
- For an ALLOCATE command the memory object is immediately allocated.

When you create an in-memory file, the default FRCVOPT is X'24', which means that this file does not participate in journal and checkpoint logging. The DUMP and RESTORE commands can process in-memory files.

Note

You cannot create in-memory files that are part of a multi-dataset file, as in the following example:

```
CREATE FILE filename FROM filename1,filename2
```

Opening and closing in-memory files

You must define the data set with the MEMORY and PAGES keywords, then create it, open it, and initialize it. Afterward, you can use the file as you would any other Model 204 file.

- The memory object and file pages are not freed if the file is physically closed.
- The FPL page enqueue list is maintained for in-memory files.
- The system enqueues used to protect FPL page updates are not used for in-memory files.

Recovery, Reliability, and CCAJRNL

IGNORE option for RESTART command

The RESTART command IGNORE option enables you to remove a problem file from restart, thus allowing successful recovery of the remaining non-problem files.

The RESTART syntax is:

```
RESTART ROLL BACK IGNORE (FILE1, FILE2, FILE3, FILE4)
- ERROR CONTINUE ROLL FORWARD
```

IGNORE *file_list* may appear anywhere in the RESTART command. If the file list is not the last parameter in the RESTART command, the list must be enclosed in left and right parentheses, even if only one file is specified.

The IGNORE option will force Model 204 recovery to skip processing of the listed file names. The file names may have been allocated either statically or dynamically.

The IGNORE option automatically forces the RERUNRB option. This means that all files not ignored will be rolled back, even if there had been a previous recovery run using the same RESTART and CCARF data sets.

A new informational message, [M204.2915](#), is issued during RESTART when a named file is ignored.

Example of processing output with IGNORE option on RESTART

The following example shows processing output with the IGNORE option:

```
M204.2512: ROLL BACK WILL USE THE FOLLOWING DATASET: RESTART
M204.1523: EOF REACHED IN FIRST PASS OF RESTART STREAM AFTER SEQ# 5558 OF 11.181 17:10:36.0
M204.2915: IGNORE SPECIFIED FOR FILE K100VAIL BUT RECOVERABLE UPDATES WERE AVAILABLE
M204.2915: IGNORE SPECIFIED FOR FILE DEBCPROC WHICH HAD NO RECOVERABLE UPDATES
M204.2915: IGNORE SPECIFIED FOR FILE K100UTAH BUT RECOVERABLE UPDATES WERE AVAILABLE
M204.0148: THE FOLLOWING FILES WILL BE ROLLED BACK:
M204.0149: K100FIJI
M204.0149: K100MOAB
M204.0149: K100HILO
M204.0149: K100KONA
M204.0149: K100THAI
M204.0149: K100SNOW
M204.0149: K100TBLE
M204.0149: K100TBLX
M204.0157: ROLLED BACK TO TRANSACTION CHECKPOINT 11.181 17:09:36.33
M204.0158: END OF ROLLBACK
```

Changes to journal record layouts

The following topics describe changes to journal record layouts in Model 204 version 7.4.0.

Increased journal block header information

Various journal analysis utilities require additional journal information at sites that embrace GDG streams and the SWITCH STREAM command as a means to keep their Onlines up for longer periods than before.

Several pieces of information have been added to the header for each journal block:

Name of information	EQU	Type of information
AEHVERSN	30,8	Copy of KBVERSN
AEHJOBNM	38,8	Job name
AEHJESID	46,8	JES Job ID
AEHSMFID	54,4	SMF System ID

In pre-7.4.0 versions of Model 204, each Online environment might produce a single merged journal daily. If an Online was bounced during the day, then the various journals for the day were merged into a single daily journal. Usually the daily merged journal contained records from only one run. The single merged journal could be used to automate media recovery, rather than manually assemble the recovery job journal concatenation.

Using the SWITCH STREAM command, a site can still prepare a daily merged journal. However, when the SWITCH STREAM command is used, each journal merged does not begin with the initialization of the Online. This means that no Type 12 records or M204.0061 initialization messages are included. Although Model 204 recovery, including media recovery, can successfully process these merged journals, various journal analysis utilities cannot.

To ensure that the required information is present in the first block of each journal data set created by the SWITCH STREAM command, the header was expanded. This may add less than 4% to the size of the journal.

Expanded journal entries

Increase in size of journal entries.

The size of the journal entries for field updates is increased by two bytes (or less than four percent). For details, see [Increased journal block header information, on page 47](#).

Model 204 SQL Catalog enhancements

This release contains the following Model 204 SQL Catalog enhancements.

LOB fields in SQL statements

The SQL Server supports the definition, update and retrieval of columns containing CLOB (Character Large Object) and BLOB (Binary Large Object) data.

CLOB/BLOB sizes

The maximum size LOB (Large Object) that may be transferred is dependent upon the amount of storage available to allocate the SQLBUF and the Universal Buffer. The SQLBUF and the Universal Buffer must be large enough to hold the entire LOB. Model 204 v7.4 SQL supports a maximum size CLOB/BLOB of up to 1 GB for UPDATE or INSERT and up to 2 GB for SELECT depending on the amount of server memory available.

The CHAR_MAX_LENGTH in the SQL catalog is set to a non-zero value for CLOB and BLOB fields. This value is for internal use only.

SQL statements supporting CLOB/BLOB data types

The CREATE TABLE, SELECT, INSERT, UPDATE, and DELETE SQL statements have been enhanced to support the CLOB and BLOB data types. See the following sections for details.

CREATE TABLE

To define a column containing CLOB or BLOB data, specify the keyword "CLOB" or "BLOB" for the data type in the CREATE TABLE statement.

The following example defines a TABLE called MANUALS that contains all of the manuals for Model204. RECTYPE and NAME are CHARACTER columns that identify the format and the name of the manual. MANUAL is a CLOB column that contains the actual text of the manual.

```
CREATE TABLE MANUALS
( RECTYPE
  CHAR(15),
  NAME
  CHAR(25)
  MANUAL
  CLOB)
```

The following example defines a table containing employee information and a picture of the employee.

```
CREATE TABLE EMPLOYEE
( ID
  CHAR(15),
  NAME
  CHAR(20),
  PICTURE
  BLOB)
```

SELECT

To download a CLOB or BLOB column, specify the column name in the SELECT statement as you would for any other column data type. However, you cannot perform a search on a CLOB/BLOB column. In other words a CLOB or BLOB column name cannot be specified in the WHERE clause.

For example, you can use the following SELECT statement to download the User Language Manual from the MANUALS table defined above:

```
SELECT MANUAL FROM MANUALS
WHERE NAME = 'User Language Manual'
```

INSERT

To insert a CLOB or BLOB column into a row, the INSERT statement specifies the CLOB or BLOB column name as it would any other type of column. The value of the CLOB or BLOB data must be specified as a parameter.

INSERT and UPDATE of LOB data through SQL requests can be CPU intensive. Therefore, Rocket Software recommends that when inserting or updating very large LOBs (over 100MB) or many small/medium size LOBs (0-100MB), Model 204 USER priority should be set to low for users doing such SQL requests, to allow other users in the Online to process normally. If possible, such updating should be done during non-peak hours. Normal single (or small number) INSERT/UPDATE of small LOBs should not noticeably affect other users.

We recommend that each customer determine the CPU usage effects of SQL LOB processing applications for each instance of the application before distributing such applications to users.

UPDATE

To update a CLOB/BLOB field in a row, the UPDATE statement specifies the CLOB or BLOB field name as it would any other type of field. The value of the CLOB or BLOB data must be specified as a parameter.

DELETE

To delete a row containing a CLOB or BLOB column, use the DELETE statement with its usual syntax.

Limitations

Following are the known limitations when using CLOB or BLOB fields in SQL statements.

LOB values cannot:

- be key values
- be compared in predicates
- appear in any clause which will involve a data comparison, including but not limited to:
 - GROUP BY
 - HAVING
 - ORDER BY
 - SELECT DISTINCT
 - WHERE
 - ON
 - IN
 - LIKE

Multiple CLOB/BLOB fields cannot be specified in an INSERT or UPDATE statement. To store multiple CLOB/BLOB columns per row, you must execute a separate UPDATE statement for each CLOB/BLOB. This limitation does not apply to SELECT; you can specify multiple CLOB/BLOB columns in the SELECT statement.

ONLINE Parameter Considerations

The SQLBUFSZ parameter defines the length of the maximum incoming SQL message. The incoming SQL message contains the SQL request in addition to the data. For performance reasons, you might want to adjust the value of the SQLBUFSZ parameter to accommodate the INSERT or UPDATE of very large CLOB/BLOB columns. The SQLBUFSZ is expanded dynamically based on the data that is sent. However, it is recommended that the value of the SQLBUFSZ parameter be set to the length of the average size CLOB/BLOB that is inserted or updated.

Catalog changes

The data columns have been mapped as follows:

Model 204 Field attribute	SQL data type
CLOB	CLOB
BLOB	BLOB

SQL Catalog (CCACAT file) implementation

Any new SQL Catalog that you create by following the SQL Installation instructions in your *Rocket Model 204 Installation Guide* will be completely compatible with the new BLOB/CLOB data types.

If you have an older, pre-v7.4 SQL Catalog, you can perform the procedure below to use the catalog without recreating it. This saves the time required to rerun all DDL to recreate the current tables.

To use your pre-v7.4 SQL catalog:

1. Back up the SQL Catalog.
2. OPEN CATPROC.
3. INCLUDE ODBCTABLES.INSTALL
This procedure updates the CCACAT file and installs the necessary SQL data types, BLOB/CLOB, for use in your DDL processing.
4. To test this installation:
 - a. CREATE a table that includes BLOB or CLOB data types.
 - b. Run the following SQL SELECT statement:

```
SELECT type_name, data_type FROM CATALOG.ODBC_TYPES
```

This returns the following results showing the new BLOB/CLOB data types.

TYPE_NAME	DATA_TYPE
CHAR	1
DEC	3
DOUBLE PRECISION	8
FLOAT	6
INT	4
NUMERIC	2
REAL	7
SMALLINT	5
BLOB	30
CLOB	40

MQ/204

MQ/204 DEFINE QUEUE statement enhancement

DEFINE QUEUE can now be omitted if the MQDEQMAN keyword is supplied.

When a queue is known to the default queue manager, as specified by the MQDEQMAN keyword, it is now possible to omit an explicit DEFINE QUEUE command. Instead MQPUT and MQGET processing can specify the external queue name.

If MQDEQMAN is not specified or the supplied queue name is not valid, the MQPUT or MQGET statement is rejected with the new message:

```
M204.0630: IMPLICIT DEFINE COMMAND REJECTED
```

Other features

Subroutine processing

This release includes an optimization for complex subroutine processing.

New demonstration database files

The demonstration database has been enhanced with additional files.

These files are allocated and restored as part of the standard installation process. For details about the demonstration database, see the *Rocket Model 204 Installation Guide* for your platform.

Chapter 3: Upward compatibility issues

Topics covered in this chapter

Increased SERVSZ requirement
CCAJRNL and CHPNT data sets
ANDIF and ORIF reserved words
Long DDNAME in OPEN or CREATE commands
BLDREUSE command with NEW option
\$DSN and \$DSNNUM functions
FLOD multistep processing
Horizon IPv6 addressing and symbolic addresses
STORE RECORD limitation

This chapter describes possible compatibility issues between Model 204 version 7.4.0 and previous releases.

Increased SERVSZ requirement

In version 7.4, servers and some server tables are always aligned on a 4 K page boundary. This can result in a server size increase if you did not use server alignment previously.

To avoid server errors, you might need to increase the setting of the SERVSZ parameter by up to 24528 bytes.

For more information about SERVSZ, see the description of server areas in the *Rocket Model 204 System Manager's Guide*.

CCAJRNL and CHPNT data sets

Due to changes in Journal record layouts, CCAJRNL and CHPNT data sets are not compatible with previous releases of Model 204.

For more information, see [Changes to journal record layouts, on page 47](#).

ANDIF and ORIF reserved words

ANDIF and ORIF require special attention when used in field names.

ANDIF and ORIF have been added to the list of reserved words that must follow certain conventions if used in a Model 204 field name.

Like the other reserved words in that list, ANDIF and ORIF:

- can be part of an unquoted string if the word is not surrounded by spaces,
- can be part of a quoted string if the word does not stand alone, and
- can be used as part of a field name if the field name is enclosed in single quotation marks.

For more information, see the section “Reserved words requiring special attention in field names” in the *Model 204 File Manager's Guide*.

Long DDNAME in OPEN or CREATE commands

CREATE and OPEN commands do not accept a DDNAME of more than eight characters.

In version 7.4.0, if a CREATE or OPEN command is issued with a DDNAME greater than eight characters, the command is rejected and the following message is issued:

```
M204.1903: %C EXCEEDS 8 CHARACTERS IN LENGTH
```

In earlier versions of Model 204, if the DDNAME contained more than eight characters, the first eight characters were used.

For example, in

```
CREATE FILE ARKANSAS FROM ARKANSAS ARKANSAS1
```

the ddnames ARKANSAS and ARKANSAS1 were both accepted as different ddnames. However, ARKANSAS was the only valid ddname and was used twice, resulting in subsequent file read errors.

To use the previous functionality of allowing more than eight characters, perform one of the following actions:

- specify CUSTOM=10 within the Online CCAIN, or
- reset CUSTOM to include 10 (for example use RESET CUSTOM=10 at the command level).

BLDREUSE command with NEW option

BLDREUSE NEW must execute in single-user (NUSERS=1) mode.

As of version 7.4.0, the BLDREUSE NEW command can only be executed when NUSERS=1; in other words, in single-user mode only. This command, with NEW, requires exclusive access to the file and completely rebuilds the reuse queue from scratch.

If the NEW option is not specified, then the BLDREUSE command can execute in multiuser or single user mode, as in previous releases.

\$DSN and \$DSNNUM functions

\$DSN and \$DSNNUM require OPEN files.

The functions \$DSN and \$DSNNUM now require that the database files and external files that you inspect be OPEN by the user who is executing the User Language request.

This ensures reliable output.

FLOD multistep processing

The size of the deferred index update records for ordered index entries has been increased by 1. This affects multistep FLODs.

Wherever you have reorganization procedures, you must change the line of code starting with RECORD TYPE=V from LENGTH=(269...) to LENGTH=(270...) in sort jobs, as illustrated in the following code:

```
SORT FIELDS=(5,85,CH,A)
RECORD TYPE=V,LENGTH=(270,90,270,90)
MODS
E15=(OI150S,1500,CCAEXITS,N),E35=(OI350S,800,CCAEXITS,N)
```

Horizon IPv6 addressing and symbolic addresses

Take care using symbolic addresses when your Online is running on a host that is enabled for IPv6 addressing. If your Online needs to communicate with IPv4 partner nodes, you must specify LOOKUP=IPV4ONLY on the LINK or PROCESSGROUP definition. For details, see [Compatibility issues, on page 20](#).

STORE RECORD limitation

A change to improve performance and reduce space requirements in VTBL and STBL has created a limit of 127 on the number of fields that can be added in a STORE RECORD statement using a subscripted variable. If you are currently adding more than this number, you will need to recode your application to add 127 or fewer initially, followed by a record update statement to add the additional fields.

Example:

```
STORE:
STORE RECORD
  FIELDX = %ARRAY(1)
  FIELDX = %ARRAY(2)
  ...
  FIELDX = %ARRAY(127)
END STORE
FRN IN STORE
  FOR %X FROM 128 TO 999
    ADD FIELDX = %ARRAY(%X)
  END FOR
END FOR
```

Chapter 4: Security enhancements

Topics covered in this chapter

[Mixed-case login password support](#)

This chapter describes security enhancements for this release.

Mixed-case login password support

Mixed-case passwords are supported for improved login security.

Mixed-case passwords are supported for:

- logins via the RACF, ACF2 and TOP SECRET interfaces, and
- logins using CCASTAT passwords.

To enable mixed-case login password support, set the CUSTOM parameter in CCAIN. For more information, see [CUSTOM, on page 57](#).

To store a mixed-case-login password in CCASTAT, specify the *LOWER command before any LOGCTL command that adds or changes a login password.

CCASTAT passwords can never be displayed, so if a user's password is rejected, use the LOGCTL command to change that user's password and try again.

Note

Mixed-case passwords are not supported for files. Lowercase passwords stored in CCASTAT for files can never be used to open a file or file group.

Example of lowercase login and password

To use passwords containing lowercase characters, the Online environment must have a CUSTOM parameter setting that includes '11' in the CCAIN parameter stream:

```
//CCAIN DD *  
LOGADD=200,CUSTOM=11
```

Note

With this setting in place, automatic translation of password strings into uppercase is disabled. Any existing passwords that were saved in uppercase would need to be entered in uppercase.

To add a login id and a password with lowercase characters, issue the commands in the following example:

```

LOGIN SYSADMIN
MYLPSWD
*****
* Deactivate automatic translation of lowercase characters
* to uppercase characters
*****
*LOWER
*****
*** Add new login id and password with lowercase characters
*****
LOGCTL A NEWLID
*** M204.0374: ENTER PASSWORD,PRIVILEGES,PRIORITY
MiXCaSe,X'10',STANDARD
*** M204.0379: ENTER TERMINAL LIST, ALL, NONE, ADD, DEL, OR RETURN
ALL
*****
* Activate automatic translation of lowercase characters
* to uppercase characters
*****
*UPPER
*****
* Login with new Login ID
*****
LOGIN NEWLID
MiXCaSe

```


Chapter 5: New and enhanced parameters and commands

Topics covered in this chapter

[Parameters](#)
[Commands](#)

This chapter describes the Model 204 commands and parameters that have been added or enhanced in this release.

Parameters

The following Model 204 parameters have been added or enhanced in Version 7 Release 4.0.

ASTRPPG

The maximum value of ASTRPPG is now 3000.

Class

TABLES

Default

400

Setting

During file creation

Meaning

The number of Table A strings per page

The minimum value of ASTRPPG is 1; the maximum is 3000.

CUSTOM

The CUSTOM=11 value, new in this release, supports mixed-case passwords for improved security. The CUSTOM=10 value, new in this release, supports a DDNAME length of longer than eight characters.

Status

CUSTOM=10 New in version 7.4.0; CUSTOM=11 New in version 7.4.0

Class

SYSTEM

Default

None

Setting

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

Meaning

The CUSTOM parameter enables special CCA-written modifications to standard execution.

You do not have to enclose a single CUSTOM setting with parentheses, for example:

CUSTOM=8

Multiple values must be specified in a list and enclosed in parentheses, for example:

CUSTOM=(1,2,3,4,6,8,9)

The current usages are:

Setting	Meaning
CUSTOM=(1)	Century can be omitted on date input
CUSTOM=(2)	Century is omitted on date output
CUSTOM=(3)	\$DAYI should return day values 2 through 8. \$DAY should equate value 2 through 8 as Monday through Sunday and full length day name should be returned. Value 9 returns MONFRI.
CUSTOM=(4)	Display field defaults to DDL ABBREV
CUSTOM=(5)	Enable MQ debug
CUSTOM=(6)	The LOGOUT message is: M204.2628: uuuuuuuuuu aaaaaaaaa LOGOUT yy mmm dd hh.mm
CUSTOM=(7)	Unused as of Model 204 V7R1.0
CUSTOM=(8)	Connect* clients can issue the LOGIN command from Remote Command Line (RCL) threads. Clients can log in with a new user name without closing the current connection to start another one.
CUSTOM=(9)	Suppresses all output from all forms of the PRIORITY command
CUSTOM=(10)	Allows file names that are longer than eight characters to be entered.
CUSTOM=(11)	Controls mixed case password support. If it is set, then mixed case passwords are allowed for both external and CCASTAT security. If it is not set, all passwords are uppercase.

Using CUSTOM=10

Including CUSTOM=10 in the CCAIN parameter stream enables you to specify a DDNAME longer than eight characters. The first eight characters will be used.

Using CUSTOM=11

Including CUSTOM=11 in the CCAIN parameter stream enables you to specify passwords containing lowercase characters:

```
//CCAIN DD *
LOGADD=200,CUSTOM=11
```

DSPOPT

The X'01' and X'02' settings of DSPOPT no longer apply to servers in memory in 64-bit systems.

Status

X'01', X'02' settings changed in Version 7.4.0

Class

SYSTEM

Default

X'00'

Setting

On User 0's parameter line

Meaning

- X'01' – For servers in memory, do server swapping by moving 4 K pages, which is faster than moving byte by byte.
- X'02' – Put servers in a hiperspace instead of a dataspace.

As of 7.4.0, these settings do not apply to servers in memory in 64-bit systems, because these servers are always aligned on a 4 K page boundary.

However, these settings still apply to real dataspace for non-64-bit systems.

For information about all DSOPT settings, see the *Rocket Model 204 Parameter and Command Reference*.

JOBSTEP

Class

SYSTEM

Default

None

Setting

View-only

Meaning

The JOBSTEP parameter is now viewable. This parameter should be blank unless you are executing a JCL PROC.

When Model 204 runs as part of a JCL PROC, and you execute the same JCL PROC repeatedly in a given JCL stream, you can identify each JCL PROC by its JOBSTEP value.

JOBSTEP: the JOBSTEP name on the EXEC JCL card that runs the JCL PROC.

STEPNM (if JOBSTEP is not blank): the name on the EXEC JCL card that specifies the module name in the PGM=JCL parameter.

Example:

```
//PROCNM PROC ZGM=ONLINE
//ONLINE EXEC PGM=&ZGM, <<<<< STEPNM parameter value
....
// PEND
//JOBSTNM EXEC PROCNM ZGM=ONLINE <<<< JOBSTEP parameter value
....
V JOBSTEP
JOBSTEP JOBSTNM JOB STEP NAME
V STEPNM
STEPNM ONLINE STEP NAME
```

The JOBSTEP parameter also appears in the informational message [0061](#) during system initialization.

OPSYS

New values have been added to OPSYS to support EAV volumes, 1MB pages, and the Breaking Event Address Register.

The OPSYS parameter specifies the operating system under which Model 204 is running.

The following new values are provided for OPSYS:

- X'000080': Breaking Event Address Register supported

- X'000040': EAV volumes supported
- X'000020': 1 MB pages are supported

All valid OPSYS settings are as follows:

Value	Purpose
X'800000'	Obsolete bit; ignore.
X'400000'	z/OS (any version). Bit is set whenever environment is z/OS.
X'200000'	IUCV communication available and in use.
X'100000'	z/OS supports real 64-bit storage.
X'020000'	64-bit virtual storage is supported.
X'008000'	z/VSE operating system.
X'004000'	z/VSE 370 is supported.
X'002000'	z/VSE 31-bit addressing is supported.
X'001000'	z/VM (any version). Bit is set whenever environment is z/VM. z/VM Release 7 or earlier, if not combined with X'000400'.
X'000800'	z/OS/ESA (z/OS SP2 or later). This option can be summed with X'400000'.
X'000400'	z/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'	Large data sets are supported.
X'000080'	Maintains the Breaking Event Address Register (BEAR), making its stored value available to Model 204 debugging code. The BEAR, available on z9 or higher systems, stores the address of a breaking event in CPU sequential processing, such as a branch.
X'000040'	EAV volumes are supported.
X'000020'	1 MB pages are supported.
X'000008'	I/O from 64-bit virtual storage areas is supported.

SCHDOFL

Status

Existed in previous releases but was not documented

Class

SYSTEM

Default

2

Setting

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

Meaning

Specifies the target number of threads on the MP offload queue per active task. Setting this parameter makes the Model 204 MP subtask scheduler more responsive to instantaneous increases in load.

The default value of 2 means that if there are more than twice as many units of MP offloadable work waiting to be processed in an MP subtask as there are active subtasks, and fewer than AMPSUBS subtasks are currently running, another MP offload subtask (not to exceed AMPSUBS) is activated. Subtasks are deactivated, but remain available, when they have no work to do. This deactivation helps minimize MP overhead.

SCHDOFL has no effect if MP/204 is not in effect in an Online (NMPSUBS=0) or if the number of active subtasks (AMPSUBS) is set to 0 or 1. (If AMPSUBS=1, an MP subtask is activated as soon as there is any offloadable work and then runs until there is no more offloadable work for it to process.)

Usage Notes

The default SCHDOFL value of 2 is recommended in most cases.

- Setting the value to 1 potentially increases throughput but also increases overhead, which is not recommended unless your site has plenty of spare CPU capacity.
- A value of 0 is not recommended because it will likely result in the unnecessary activation of MP subtasks – by the time some of them are dispatched, other MP subtasks will have handled all pending units of work.
- A value greater than 2 might decrease CPU overhead while reducing throughput by reducing parallelism.

SCHDOFLZ

Status

New in Version 7.4.0

Class

SYSTEM

Default

2

Setting

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

Meaning

Applies to z/OS only

On a z/OS system, Model 204 zIIP support enables you to offload Model 204 work from regular processors to zIIP processors.

SCHDOFLZ specifies the target number of threads on the zIIP offload queue per active zIIP subtask. Setting this parameter makes the Model 204 zIIP subtask scheduler more responsive to instantaneous increases in load.

The default value of 2 means that if there are more than twice as many units of zIIP offloadable work waiting to be processed in a zIIP subtask as there are active subtasks, and fewer than AMPSUBZ subtasks are currently running, another zIIP offload subtask (not to exceed AMPSUBZ) is activated. Subtasks are deactivated, but remain available, when they have no work to do. This deactivation helps minimize zIIP overhead.

SCHDOFLZ has no effect if MP/204 is not in effect in an Online (NMPSUBZ=0) or if the number of active subtasks (AMPSUBZ) is set to 0 or 1. (If AMPSUBZ=1, a zIIP subtask is activated as soon as there is any offloadable work and then runs until there is no more offloadable work for it to process.)

Usage Notes

The default SCHDOFLZ value of 2 is recommended for most sites.

- Setting the value to 1 potentially increases throughput but also increases overhead, which is not recommended unless your site has plenty of spare CPU capacity.
- A value of 0 is not recommended because it will likely result in the unnecessary activation of zIIP subtasks – by the time some of them are dispatched, other zIIP subtasks will have handled all pending units of work.
- A value greater than 2 might decrease CPU overhead while reducing throughput by reducing parallelism.

SCHDOPT

The SCHDOPT parameter has the following new values:

- X'20': Do not let the Main Task pick up zIIP workload, even if no other work is available.
- X'40': Do not let an MP subtask pick up zIIP workload, even if no other work is available.
- X'80': Allow zIIP to do server swapping.

Valid settings of SCHDOPT, which can be summed, are described in the following table.

- Setting X'10' is relevant to all sites.
- All other settings are relevant only to sites running MP/204.

Value	Purpose
X'00'	No scheduler overhead tracking, no forced offload. This is the default.
X'01'	Maintain MAINTASK scheduler overhead.
X'02'	Run MP work on an MP subtask even if MAINTASK is not busy.
X'04'	Defer page releases until server swap.
X'08'	Do server I/O on MP subtask.
X'10'	In CSLICE, check maximum values even if there is an SWAIT.
X'20'	Run zIIP work on a zIIP subtask even if MAINTASK is not busy.
X'40'	Run zIIP work on a zIIP subtask even if MP subtasks are not busy.
X'80'	<p>The SCHDOPT parameter setting of X'80' allows server swapping to be offloaded to zIIP processors. If you have SCHDOPT=80 set, then you probably want to set SCHDOPT1=01 as described in the "Usage Notes" below.</p> <p>The X'80' setting is only valid when CCASERVR is in memory (servers swapped into memory). If you set SCHDOPT=X'80' when CCASERVR is not in memory, the setting X'80' is reset and message 2914 is displayed:</p> <p>SCHDOPT INDICATION OF SERVER SWAPPING DONE BY ZIIP IS ONLY VALID WHEN CCASERVR IS IN MEMORY</p> <p>Initialization then continues.</p>

Note

If X'80' is on and X'08' is off, then only the zIIP subtask will do server swapping.

Usage Notes

- If X'80' is on and X'08' is off, then only the zIIP subtask will do server swapping.

- SCHDOPT=08: An MP subtask will always execute a swap unit of work before a user unit of work. This could mean that every MP subtask is busy swapping.
- SCHDOPT=80: The zIIP subtask is doing the swap work and the MP subtasks will do real user work.
- SCHDOPT=88: The zIIP subtask is behaving exactly as an MP subtask. Both zIIP and MP subtasks could be busy doing swapping while no work is actually being done.
- SCHDOPT=80 and SCHDOPT1=01: The zIIP subtask will do the swapping before doing any work and the MP subtasks will do real user work. But if there is no user work to do, the MP subtask will help the zIIP subtask by executing a swap unit of work. That way, if the zIIP subtask gets behind on its swapping, then the MP subtask can assist.

SCHDOPT1

The new parameter SCHDOPT1 uses an x'01' value. SCHDOPT1=x'01' allows an MP subtask to handle server swapping if the subtask has nothing else to do.

SCHDOPT1 is useful when zIIP is handling server swapping and needs help from an MP subtask.

For more information about zIIP server swapping, see [SCHDOPT](#), on page 62.

SERVNSA

The SERVNSA and SERVNSSZ parameters are used with FTBL storage above the bar.

Class

USER

Default

0

Setting

On User 0's parameter line

Meaning

When storing FTBL above the bar, two parameters control non-swappable server areas: SERVNSA and SERVNSSZ.

- SERVNSA (server non-swappable areas) indicates the tables that you want to be above the bar. At this time, only FTBL may be selected and its appropriate setting is x'02000000'.
- SERVNSSZ (server non-swappable size) is the amount of space in bytes required for the above-the-bar server tables per user. This size is rounded up to the nearest 4 K. The total amount of storage allocated for non-swappable server areas equals SERVNSSZ multiplied by NUSERS.

When sizing SERVNSSZ you should use the largest FTBL sizes that might be needed.

For more information about FTBL storage above the bar, see the *Rocket Model 204 System Manager's Guide*.

Usage Notes

The non-swappable server area can be used with server swapping done either in storage or on disk. The non-swappable server area can also be used when there is no server swapping (NUSERS=NSERVS). Loading the non-swappable server area above the bar enables you to make servers below the bar smaller, thus saving storage below the bar.

To obtain below-the-bar storage savings when the non-swappable server area is used, decrease the value of the SERVSIZE parameter by the LFTBL value used in calculating the server size. For details about LFTBL, server sizing, and server storage, see the *Rocket Model 204 System Manager's Guide*.

SERVNSSZ

The SERVNSSZ and SERVNSA parameters are used with FTBL storage above the bar.

Class

USER

Default

0

Setting

On User 0's parameter line

Meaning

When storing FTBL above the bar, two parameters control non-swappable server areas: SERVNSSZ and SERVNSA.

- SERVNSSZ (server non-swappable size) is the amount of space in bytes required for the above-the-bar server tables per user. This size is rounded up to the nearest 4 K. The total amount of storage allocated for non-swappable server areas equals SERVNSSZ multiplied by NUSERS.

When sizing SERVNSSZ you should use the largest FTBL sizes that might be needed.

- SERVNSA (server non-swappable areas) indicates the tables that you want to be above the bar. At this time, only FTBL may be selected and its appropriate setting is x'02000000'.

For more information about FTBL storage above the bar, see the *Rocket Model 204 System Manager's Guide*.

Usage Notes

The non-swappable server area can be used with server swapping done either in storage or on disk. The non-swappable server area can also be used when there is no server swapping (NUSERS=NSERVS). Loading the non-swappable server area above the bar enables you to make servers below the bar smaller, thus saving storage below the bar.

To obtain below-the-bar storage savings when the non-swappable server area is used, decrease the value of the SERVSZ parameter by the LFTBL value used in calculating the server size. For details about LFTBL, server sizing, and server storage, see the *Rocket Model 204 System Manager's Guide*.

SMTPCADD

The SMTPCADD parameter supports IPv6 network addresses. Symbolic names, such as CCAMVS1, are looked up using the GETADDRINFO service, which is capable of returning IPv6 addresses.

You can explicitly type in addresses:

- for IPv4 addresses use dotted decimal notation
- for IPv6 network addresses use colon-separated notation (see \$SNDMAIL description for details)

For details, see [\\$SNDMAIL function, on page 14](#).

SMTPSADD

The SMTPSADD parameter supports IPV6 network addresses. Symbolic names, such as CCAMVS1, are looked up using the GETADDRINFO service, which is capable of returning IPV6 addresses.

You can explicitly type in addresses:

- for IPV4 addresses use dotted decimal notation
- for IPV6 network addresses use colon-separated notation

For details, see [\\$SNDMAIL function](#), on page 14.

STORCURG

The new STORCURG parameter is view-only:

STORCURG - CURRENT MBYTES ATB STORAGE ALLOCATED

This parameter indicates the number of megabytes of above-the-bar storage currently allocated.

SQLBUFSZ

Class

USER

Default

0

Setting

By any user

Meaning

The size of the Model 204 SQL Server buffer that assembles incoming SQL messages from a Horizon, CRAM SQL, or IUCV SQL buffer

The SQLBUFSZ value defines the maximum length of an incoming SQL message. The incoming SQL message contains the SQL request in addition to the data. Because the largest incoming SQL message is likely to be a DDL transaction, set SQLBUFSZ large enough to accommodate your largest DDL transaction plus 50 bytes of overhead. The maximum value you can specify is slightly over two billion. The recommended initial value for the SQLBUFSZ parameter is 100000.

For...	Set SQLBUFSZ...
A General Data Stream (GDS) passed from the conversation buffer to CMIS (SQL communications interface) receiving buffer and moved to the Model 204 SQL buffer	Greater than or equal to the maximum incoming SQL message length that can be received by the SQL server.

Specify SQLBUFSZ on the first IODEV 19, 17, or 13 line. To define SQL and RCL IODEV threads on IODEV 49, see the *Model 204 System Manager's Guide*.

Handling SQL statements greater than 32k bytes

When handling an SQL statement that is greater than 32K, you must consider that 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. For example, set SQLBUFSZ=60000 to run a DDL statement that is 52K.

You can also monitor the Model 204 SQLI since-last statistic to determine the size of the largest SQL input request. For more information about the SQLI statistic, see the *Rocket Model 204 System Manager's Guide*.

Handling BLOBs and CLOBs

When handling large objects (BLOBs and CLOBs), for performance reasons, you might want to adjust the value of the SQLBUFSZ parameter to accommodate the INSERT or UPDATE of

very large CLOB/BLOB columns. The SQLBUFSZ is expanded dynamically based on the data that is sent. However, it is recommended that you set the value of the SQLBUFSZ parameter to the length of the average size CLOB/BLOB that is inserted or updated.

STORMAXG

The new STORMAXG parameter is view-only:

STORMAXG - HWM MBYTES ATB STORAGE ALLOCATED

This parameter indicates the high water mark of megabytes of above-the-bar storage allocated during a run.

STORMXZP

Class

SYSTEM

Default

None

Setting

View-only

Meaning

The high water mark of megabytes of above-the-bar storage allocated with 1 MB pages during the run.

STORMXZP applies only to z/OS.

STORZPAG

Class

SYSTEM

Default

None

Setting

View-only

Meaning

The current number of megabytes of above-the-bar storage allocated with 1 MB pages.

STORZPAG applies only to z/OS.

TCPOPTS

TCPOPTS is a view-only parameter showing the TCP/IP option in use, either IPv6 or IPv4 addressing. The value of TCPOPTS is set by Model 204 based on the addresses that you supplied for the SMTPCADD and SMTPSADD parameters (specifying the SMTP client and server default addresses).

The values for TCPOPTS are described in the following table.

Value	Description
X'00'	SMTP is an IPv4 address (both the SMTP server and client addresses are in IPv4 format). This is the default.
X'01'	SMTP client is an IPv6 address.
X'02'	SMTP server is an IPv6 address.

The settings can be combined; X'03' indicates that both the SMTP client and server addresses are in IPv6 format.

For details on SMTPCADD and SMTPSADD, see the [\\$SNDMAIL function, on page 14](#).

TEMPBOP

Status

New in version 7.4.0

Class

SYSTEM

Default

1% (truncated) of TEMPSIZE

Setting

On EXEC card or on User 0 parameter line

Meaning

The TEMPBOP parameter defines the number of CCATEMP pages that must be free to start a sub-transaction.

An update transaction consists of one or more sub-transactions. A typical sub-transaction consists of reading a Model 204 page to be updated, writing the image of the page to the checkpoint stream, making an update to the page, creating back out and constraint entries in CCATEMP (in case the update is backed out) and writing the update to the journal stream.

If TEMPBOP pages are not free in CCATEMP when a sub-transaction starts, the following message is displayed:

M204.0441: CCATEMP FULL

and the user's entire transaction is backed out.

The M204.0441 message indicates that you have a potential problem; for example, there is a runaway transaction, CCATEMP is too small, or a transaction is sitting in a wait holding large amounts of CCATEMP space. The update is canceled, but the Online is not forced down.

If TEMPBOP pages are not free in CCATEMP as the sub-transaction proceeds, the following message is displayed:

M204.2904: INSUFFICIENT CCATEMP PAGES FOR BACKOUT DATA

and all in-flight transactions are backed out and the Online comes down. If this occurs, increase the sizes of CCATEMP and TEMPBOP.

TEMPBOP value

TEMPBOP can be set on the EXEC card PARMS or as a USER0 parameter. If not specified, TEMPBOP is calculated during Model 204 initialization, and the value is 1% (truncated) of TEMPSIZE.

For example, if TEMPSIZE = 80, then TEMPBOP = 0, and if TEMPSIZE=141, TEMPBOP = 1.

MINIMUM value = 0

MAXIMUM value = 16777216

ZPAGEOPT

The ZPAGEOPT parameter specifies which areas should be allocated using 1 MB pages.

ZPAGEOPT is a user 0 parameter with the following values:

Value	Description
X'0001'	Use 1 MB pages for servers above the bar
X'0002'	Use 1 MB pages for buffers above the bar
X'0004'	Use 1 MB pages for buffer hash cells above the bar
X'0008'	Use 1 MB pages for buffer control blocks above the bar
X'0010'	Use 1 MB pages for CCATEMP pages in storage above the bar
X'0020'	Use 1 MB pages for CCATEMP in storage control blocks
X'0040'	Use 1 MB pages for the record locking table when it resides above the bar
X'0080'	Use 1 MB pages for APSY saved compilations in storage above the bar

ZQMAX

The ZQMAX parameter limits the number of work units that may be placed for execution by the zIIP task.

The default and maximum value for ZQMAX is 65535.

For details, see [ZQMAX, on page 37](#).

Commands

The following Model 204 commands have been enhanced in this release.

ALLOCATE

EATTR parameter (z/OS only)

Specifying the EATTR parameter with the ALLOCATE command enables a data set to be allocated on an EAV (Extended Address Volumes) volume. (An EAV volume has more than 64K cylinders.) For details, see [EAV, on page 36](#).

The EATTR parameter is effective only in z/OS version 1.12 and later; in all other environments it is ignored with no error message.

MEMORY and PAGES keywords

Specifying the keywords MEMORY and PAGES with the ALLOCATE command enables you to use temporary in-memory files.

- The MEMORY keyword designates an in-memory file, specifically an above-the-bar memory object.
- The PAGES keyword specifies the number of Model 204 (6K) pages to allocate for the file.

For example, the following command allocates an in-memory file named BUSES:

```
ALLOCATE BUSES WITH MEMORY PAGES=10000
```

For details, see [in-memory files, on page 44](#).

CLOSE LINK

CLOSE LINK has a new ABANDON parameter to disable an opened Horizon VTAM link.

Syntax

```
CLOSE LINK linkname [FORCE | IMMD | ABANDON]
```

where:

- *linkname* is the link specified on the DEFINE LINK command.
- FORCE specifies that any conversations using the link are ended immediately, the SNA Communications Server ACB is closed, and all internal buffers and control blocks are released. If you want a more gradual termination of network activities, refer to the STOP command in the *Model 204 Parameter and Command Reference*.
- IMMD specifies that the ACB for the link in question should be closed immediately. By closing the ACB inline, you can significantly speed up CLOSE LINK processing, because it supplants the scheduling of the CLOSE ACB on a Horizon pseudo subtask.
- ABANDON specifies that if the FORCE argument has not successfully closed the link, because there are VTAM requests outstanding, ABANDON processing will deactivate the link and reclaim as much storage as possible. You can reestablish communication using an alternate link.

Usage Notes

CLOSE LINK FORCE disables the link specified in a DEFINE LINK command for any Model 204 non-CNOS intersystem link except TPROCESS, Transfer Control, and UL/DB2. Conversations using the link are ended immediately and no new ones can start. The SNA Communications Server ACB or CRAM or IUCV channel is closed, and all internal resources, buffers, and control blocks are released.

If the link you are closing supports a combination of CNOS and non-CNOS sessions, the CLOSE LINK without the FORCE parameter drains the conversations on CNOS sessions first, and then it forces the conversations on non-CNOS sessions.

The CLOSE LINK ABANDON command eliminates the need to bring down an Online if a VTAM link fails, allowing the possibility to establish communications using an alternate link.

The CLOSE LINK command is not required at shutdown; an implicit CLOSE LINK is performed during Model 204 termination.

COMPACTB

The COMPACTB command combines as many extension records as possible into one extension record for the current or specified file.

COMPACTB checking for bump or long request conditions has been enhanced to eliminate possible monopolization of resources when files with Table X are processed.

When compacting files with Table X, the check for bump or long request conditions is done for every:

- 30 compacted records, or

- 30 processed Table B pages, or
 - 30 logically deleted records,
- whichever comes first.

COMPACTB checking for files with no Table X has not changed; it is done for every 30 Table B records.

DEFINE DATASET

EATTR parameter (z/OS only)

Specifying the EATTR parameter with the DEFINE DATASET command enables a data set to be allocated on an EAV (Extended Address Volumes) volume. (An EAV volume has more than 64K cylinders.) For details, see [EAV, on page 36](#).

The EATTR parameter is effective only in z/OS version 1.12 and later; in all other environments it is ignored with no error message.

MEMORY and PAGES keywords

The DEFINE DATASET command supports temporary in-memory files with the corequisite keywords: MEMORY and PAGES.

- The MEMORY keyword designates an in-memory file, specifically an above-the-bar memory object.
- The PAGES keyword specifies the number of Model 204 (6K) pages to allocate for the file.

Example

The following command defines an in-memory file named BUSES:

```
DEFINE DATASET BUSES WITH SCOPE=SYSTEM MEMORY PAGES=10000 DIRECT
```

For details, see [in-memory files, on page 44](#).

LOGCTL

LOGCTL now allows you to specify mixed-case passwords for improved login security.

Mixed-case login passwords

To store a mixed-case-login password in CCASTAT, specify the *LOWER command before any LOGCTL command that adds or changes a login password.

CCASTAT passwords can never be displayed, so if a user's password is rejected, use the LOGCTL command to change that user's password and try again.

Note

Mixed-case passwords are not supported for files. Lowercase passwords stored in CCASTAT for files can never be used to open a file or file group.

To enable mixed-case login password support, the CUSTOM parameter in CCAIN must also be set.

For details, see [Mixed-case login password support, on page 55](#).

Example

To add a login id and a password with lowercase characters, issue the commands in the following example:

```

LOGIN SYSADMIN
MYLPSWD
*****
* Deactivate automatic translation of lowercase characters
* to uppercase characters
*****
*LOWER
*****
*** Add new login id and password with lowercase characters
*****
LOGCTL A NEWLID
*** M204.0374: ENTER PASSWORD,PRIVILEGES,PRIORITY
MiXCaSe,X'10',STANDARD
*** M204.0379: ENTER TERMINAL LIST, ALL, NONE, ADD, DEL, OR RETURN
ALL
*****
* Activate automatic translation of lowercase characters
* to uppercase characters
*****
*UPPER
*****
* Login with new Login ID
*****
LOGIN NEWLID
MiXCaSe

```

MONITOR GSTORAGE

The new MONITOR GSTORAGE (monitor grande storage) command displays statistics on allocated storage and 1 MB pages. It can be abbreviated as MONITOR GS or M GS.

MONITOR GSTORAGE displays the data in the following format:

```
NAME  START  SIZE (MB)  1M SIZE  PGFIX
```

where:

- NAME - name of the memory object.
Buffers are allocated in chunks and presented in one line using name "BUFFERS" and accumulated data for all chunks.
- START - starting address of the memory object. For buffers it is the address of the first chunk.
- SIZE (MB) - Size in megabytes of the memory object. For buffers it is the accumulated size of all chunks.
- 1M SIZE - Size in megabytes of memory object backed by z/OS 1 MB pages.
For buffers some chunks may be backed by 1 MB pages and some by 4 K pages, so values in columns SIZE (MB) and 1M SIZE may be different.
- PGFIX - a letter "Y" in this column indicates the storage is page fixed.

There are three statistical messages printed below the table, indicating total amount of storage allocated above the bar, total amount of storage allocated with 1 MB pages, and total amount of page fixed storage. All amounts are in megabytes. Data presented by the MONITOR GSTORAGE represent current values.

Example

The following example shows output from a MONITOR GSTORAGE command, showing the amount of storage that could – and could not – be allocated above the bar and in 1 MB pages:

```

11.238  AUG 26  14.30.43          PAGE 1
NAME      START              SIZE (MB)  1M SIZE  PGFIX
RECLTBL   00000048_00100000          1         1      Y
CCASERV   00000048_00C00000         10        10
CCATEMP   00000048_08C00000        118
CCATDBCD  00000048_10300000          1         1
BUFFHASH  00000048_10500000          1         1      Y
BUFFCNTL  00000048_10A00000          4         4
CCAAPSY   00000048_1FE00000          2
BUFFERS   00000048_11200000        119        72
Total amount of storage allocated above the bar:      256
Total amount of storage backed by 1 MB pages :      89
Total amount of storage page fixed :      2

```

MONITOR TASKS

Status

zIIP-related columns ZDEQ, SDEQ, ZIPCPU new in V7R4.0

Privileges

System administrator

Function

Usage statistics for MP subtasks

Syntax

MONITOR TASKS

Example

MONITOR TASKS

Usage notes

At installations using the MP (multiprocessing) or zIIP feature, the system administrator can use the MONITOR TASKS command to view usage information for all attached Online tasks, including the maintask.

The MONITOR TASKS command displays the data shown in the table below. When the terminal is MODEL=2, as specified by the MODEL parameter, the ZDEQ, SDEQ, LKWTIME, LKPTIME, and ZIPCPU columns are not displayed.

Table 1: MONITOR TASKS display information

Statistic	Meaning
TASK	The task number. 0 always denotes the maintask, 1 the highest priority offload subtask, 2 the next highest priority subtask, and so forth.
CPU	The total amount of CPU time used by the task (in milliseconds).
PR	Total elapsed time during which the task has been available for offload processing.
PCPU	CPU used, as a percentage of elapsed time (CPU/PR).
WAIT	The number of times the task has issued a real operating system WAIT, excluding waits for MP locks.
POST	The number of times the task issued a real operating system POST, excluding posts for MP locks.

Table 1: MONITOR TASKS display information (continued)

Statistic	Meaning
LKWAIT	The number of times the task issued a real operating system WAIT to wait for an MP lock to become available.
LKPOST	The number of times the task issued a real operating system POST to indicate the availability of an MP lock to another task.
MQWTM	The average length of time (in microseconds) required for a unit of work to get to the task from another task. This is a measure of queue delay between maintask and subtasks.
DEQ	The number of units of work taken by the corresponding task – maintask, MP subtask, or zIIP subtask – regardless of the queue that the work was taken from.
ZDEQ	The number of units of work taken from a zIIP queue by either the maintask or an MP subtask. This column has meaning only for the maintask and MP subtasks. For zIIP subtasks this column always contains a zero. Displayed only when the terminal is MODEL=5.
SDEQ	The number of units of work taken from an MP subtask queue by the maintask. This column has meaning only for the maintask; for other tasks this column always contains a zero. Displayed only when the terminal is MODEL=5.
LKWTIME	Lock wait time—Total elapsed time in milliseconds, this task spent waiting for MP locks. Displayed only when the terminal is MODEL=5.
LKPTIME	Lock preemption time—Total elapsed time in milliseconds, this task spent waiting, due to operating system preemption. This is the elapsed time between when an MP lock becomes available (lock post) making the task ready to run, and when the task actually gets the CPU. That preemption delay is caused by the operating system dispatching other tasks ahead of this task. Displayed only when the terminal is MODEL=5.
ZIPCPU	The total CPU time (in milliseconds) consumed by this zip engine. It is zero for MP subtasks and greater than zero when “z” precedes the task number and that zIIP subtask has been active (AMPSUBZ>0) at some point during the run. Displayed only when the terminal is MODEL=5.

MONITOR TASKS indicates a zIIP subtask by displaying a lowercase letter z beside the subtask number.

Most of the statistics described in the table above are also written to the journal (CCAJRNL) whenever system final or partial statistics are logged. For more information about statistics in journal entries and for general information about task off-loading and related MP concepts, refer to the *Model 204 System Manager's Guide*.

Example

The columns in MONITOR TASKS, with MODEL=2, have been rearranged and the display now looks like this.

TASK	CPU	PR	PCPU	WAIT	POST	LKWAIT	LKPOST	MQWTM	DEQ
0	234	834	280	76	14	0	0	179	78
1	0	6	0	10	8	0	0	8000	1
2	0	10	0	1	1	0	0	0	0
z 3	0	16	0	2	1	0	0	0	0
z 4	0	19	0	1	1	0	0	0	0

When the terminal is MODEL=2, the ZDEQ, SDEQ, LKWTIME, LKPTIME, and ZIPCPU columns are not displayed.

zIIP processing information

MONITOR TASKS displays three zIIP-related columns, ZDEQ, SDEQ, and ZIPCPU, on screens formatted as a 3270 Model 5 terminal.

See the preceding table for details on these statistics.

Example

The following display illustrates the output of a MONITOR TASKS command with zIIP processing statistics, as displayed on the terminal where MODEL=5. The LKWTIME and LKPTIME columns are also displayed.

TASK	CPU	PR	PCPU	WAIT	POST	LKWAIT	LKPOST	MQWTM	DEQ	ZDEQ	SDEQ	LKWTIME	LKPTIME	ZIPCPU
0	225	824	273	70	14	0	0	189	74	19	10	0	0	0
1	0	6	0	10	8	0	0	8000	1	0	0	0	0	0
2	0	10	0	1	1	0	0	0	0	0	0	0	0	0
z 3	0	16	0	2	1	0	0	0	0	0	0	0	0	0
z 4	0	19	0	1	1	0	0	0	0	0	0	0	0	0

OPEN LINK

OPEN LINK contains a new REPLACE keyword for replacing an abandoned VTAM Horizon link.

When a VTAM Horizon link has been abandoned, using the new ABANDON keyword of the CLOSE LINK command, it is sometimes possible to open a new Horizon VTAM link to replace the abandoned link and to take over its associated REMOTE, SESSIONGROUP and PROCESSGROUP definitions, using the new REPLACE keyword in an OPEN LINK command.

```
OPEN LINK newlink REPLACE oldlink
```

This is possible if the CLOSE LINK *linkname* ABANDON command left the old link in an abandoned state.

If, however, the CLOSE LINK command completed in such a way that the link can be re-opened, the following new message is produced:

```
M204.2890: LINK linkname CLOSED NORMALLY. IT CAN BE RE-OPENED
```

In this case the keyword REPLACE of the OPEN LINK command cannot be used to replace the link. Instead the link should be re-opened as normal. There are new messages, 2893 to 2897, associated with this enhanced OPEN command.

RESTART

The RESTART command IGNORE option enables you to remove a problem file from restart, thus allowing successful recovery of the remaining non-problem files.

See [IGNORE option for RESTART command, on page 47](#) for more information.

USE PROC

The USE PROC command has been enhanced with two new keywords: HDRS and CC.

USE PROC lets you save the USE data set output to a temporary user procedure. The HDRS and CC options are used to format the output. They are specified in the command syntax before the other USE PROC keywords: A(append), I(insert), R(replace), and S(shift).

- The HDRS keyword allows the HDRCTL parameter and the SET HEADER and SET TRAILER User Language statements to be recognized for the output.

- The CC keyword inserts a one-byte carriage control character at the beginning of each line of output. If you specify the CC option, the maximum line length of lines output to the procedure is reduced by one, to a value of 254 bytes.

For example:

```
USE PROC -3 HDRS CC R
```

spools the USE output to temporary user procedure number -3. Each logical output page of the procedure will display a header and trailer line. Each line of the output will be a maximum of 254 characters, with a one-byte carriage control character at the beginning.

The R option indicates that this procedure will delete any existing temporary procedure -3 and create a new one.

The number of lines of data contained on each logical page is determined by the UDDLPP parameter. For more information, see "UDDLPP: Maximum lines printed per page" in the *Rocket Model 204 Parameter and Command Reference*.

Chapter 6: Horizon IPv6 addressing support

Topics covered in this chapter

[Horizon and sockets](#)

[DEFINE LINK command for TCP/IP Horizon](#)

[DEFINE PROCESSGROUP command for TCP/IP Horizon](#)

Horizon supports IPv6 network addresses for TCP/IP, initially for z/OS only.

This support is possible because you can specify IPv6 addresses as the LOCALID argument on the DEFINE LINK command, and as the REMOTEID on the DEFINE PROCESSGROUP command.

In addition, when you can specify a symbolic address, for example as the LOCALID on the DEFINE LINK command, the address lookup uses the GETADDRINFO service instead of the GETHOSTBYNAME service, which can return IPv6 addresses to a host that has an IPv6 address.

This can be further controlled by the use of the new LOOKUP parameter on the DEFINE LINK command, to restrict the addresses returned by the GETADDRINFO service.

For details, see [DEFINE LINK: Horizon for TCP/IP, on page 19](#) and [DEFINE PROCESSGROUP: Horizon for TCP/IP, on page 19](#).

The following error message is issued when there is an incoming connection request, but the addressing family is neither AF_INET (IPv4) nor AF_INET6 (IPv6).

```
M294.2907: LINK lnkname REJECTED ACCEPT. SOCKADDR IS string
```

The socket address structure is displayed in the message, together with the link name. For details, see message [2907](#).

Note

Customers using symbolic addresses that are enabled for IPv6 could encounter compatibility issues. For details, see [Compatibility issues, on page 20](#).

Horizon and sockets

Most changes arise as a result of increasing the length of the local ID and remote ID from 15 to a new maximum of 45 bytes. 45 is the maximum length of 8 4-byte hexadecimal halfwords separated by 7 colons.

The \$SOCKET function has been enhanced so that IPv6 network addresses are now supported. For details, see [\\$SOCKET function, on page 15](#).

The ACCEPT processing code has been changed to cater for a masked IPv6 address using a colon-separated mask in the same way that incoming connection IPv4 requests can be validated against a dotted-decimal mask such as 255.255.255.0.

The MONITOR LINK and MONITOR PROCESSGROUP commands have been enhanced to handle the increased size of the local and remote network addresses, so that a full-length local or remote ID is displayed correctly.

For details, see [New MONITOR LINK format, on page 15](#) and [New MONITOR PROCESSGROUP format, on page 16](#).

DEFINE LINK command for TCP/IP Horizon

The DEFINE LINK command for TCP/IP Horizon connections has been enhanced to allow for IPv6 network addresses.

For details, see [DEFINE LINK: Horizon for TCP/IP, on page 19](#).

DEFINE PROCESSGROUP command for TCP/IP Horizon

The DEFINE PROCESSGROUP command for TCP/IP Horizon is enhanced so that the REMOTEID and MASK parameters can be specified in colon-separated IPV6 address format.

For details, see [DEFINE PROCESSGROUP: Horizon for TCP/IP, on page 19](#).

Chapter 7: Connect* Suite enhancements

Topics covered in this chapter

[Connect* for JDBC](#)

[Connect* for .NET Framework](#)

[IPv6 address support for Connect*](#)

This chapter describes enhancements to the Connect* Suite in the current release.

Note

Connect* for ODBC 7.4, which will provide IPv6 addressing capability and BLOB/CLOB support, is not yet available.

Your 7.1 version of Connect* for ODBC is compatible with Rocket Model 204 7.4.0.

Connect* for JDBC

This section describes the Connect* for JDBC enhancements in the current release.

Documentation

This release of Connect* for JDBC includes HTML help files that provide complete documentation of the J204 JDBC driver. For detailed documentation, browse the `index.HTML` file in the `\doc` directory where the driver is installed.

BLOB and CLOB support

Connect* for JDBC now provides support for Model 204 BLOB and CLOB data types in SQL update and retrieval statements.

Use the following `ResultSet` methods to retrieve large object column data after `SELECT` statement execution.

```
BLOB
byte[] getBytes()
Object getObject()
InputStream getBinaryStream()
CLOB
byte[] getBytes()
Object getObject()
String getString()
InputStream getAsciiStream()
InputStream getBinaryStream()
java.io.Reader getCharacterStream()
InputStream getUnicodeStream()
```

In `INSERT` and `UPDATE` statements, use parameters for large object columns. First, create a prepared statement, then set the values of the parameters with the following `PreparedStatement` methods.

```

BLOB
  setBytes()
  setObject()
  setBinaryStream()
CLOB
  setBytes()
  setObject()
  setAsciiStream()
  setBinaryStream()
  setCharacterStream()
  setUnicodeStream()

```

Log file location under Windows

The location of J204Driver log files has changed with this release for applications run under Windows Vista and Windows 7.

The driver places the log file in the user's home directory.

- Under Windows 7, the directory would be:

`C:\Users\user_name`

- Under Windows XP, the directory would be:

`C:\Documents and Settings\user_name`

For other operating systems, the log file is placed in the same directory as the `j204.jar` file.

Software Requirements

Model 204	Model 204 version 7.4.0 is required for use of BLOBs and CLOBs. If BLOBs and CLOBs are not used, all supported versions of Connect* are compatible with Model 204 version 7.4.0. Connect* version 7.4.0 is compatible with versions of Model 204 before 7.4.0.
Java	Java SE Development Kit (JDK) or Java SE Runtime Environment (JRE) Version 1.5 or later must be installed.
Platforms	Any operating system that supports Java Runtime Environment.

Connect* for .NET Framework

This section describes the enhancements for Connect* for .NET Framework.

Documentation

On installation, a help file is created that provides complete documentation of the new Connect* for .NET Framework features. See the `CCA.Data.chm` file for details.

Visual Studio Integration

Connect* for .NET Framework now supports integration of the Model 204 .NET Framework data provider with Microsoft Visual Studio™ 2005. Later versions are not yet supported.

Connect* for .NET Framework installation registers Model204Client as a Data Designer Extensibility (DDEX) provider. This enables you to use Model204Client with Visual Studio™ wizard and visual designer data tools to build Windows applications. Model 204 data objects

appear as an object hierarchy in the Server Explorer window, and they can be dragged and dropped into various designer tools provided by the IDE.

For documentation on using the data provider in Visual Studio™, see the *Using Model204Client in Microsoft Visual Studio* topic in the CCA.Data.chm help file that is installed with Model204Client.

BLOB and CLOB support

Connect* for .NET Framework now provides support for Model 204 BLOB and CLOB data types in SQL update and retrieval statements.

Use these DataReader methods to retrieve data from BLOB columns:

```
long GetBytes()  
object GetValue()
```

Use these DataReader methods to retrieve data from CLOB columns:

```
long GetBytes()  
long GetChars()  
string GetString()  
object GetValue()
```

In INSERT and UPDATE statements, use parameters for large object columns. First prepare the statement, then set the values of the parameters with the Parameter.Value property. The CCA.Data help file includes examples of storing and retrieving LOB columns.

Log file location under Windows

The location of Model204Client log files has changed with this release. The driver places log file in the user's home directory.

- Under Windows 7, the directory would be:

```
C:\Users\user_name
```

- Under Windows XP, the directory would be:

```
C:\Documents and Settings\user_name
```

Software Requirements

Model 204	Model 204 version 7.4. Previous versions are not supported.
Platforms	OS versions supported are Windows XP, Windows Vista, and Windows 7.
Microsoft.NET Framework	Microsoft .NET Framework 2.0.

IPv6 address support for Connect*

All Connect* clients now support Internet Protocol Version 6 (IPv6) 128-bit addresses as well as IPv4 32-bit addresses.

Connect* clients have the option of specifying explicit server addresses in either of the following formats:

- existing IPv4 dotted decimal format, for example:

```
74.6.238.254
```

- new IPv6 colon-separated hexadecimal format, for example:


```
2001:0DB8:AC10:FE01:0000:0000:0000:0000
```

Consecutive zero halfwords can be replaced by a double colon, for example:

```
2001:0DB8:AC10:FE01::
```

If an application specifies the host address as a symbolic name, then the client first attempts an IPv6 connection to the server. If that fails, the client attempts an IPv4 connection. An error is not returned to the application unless both attempts fail.

Requirements

- Client
To take advantage of IPv6 addresses, you might have to install and enable IPv6 support on your client operating system. See your client O/S documentation for instructions.
- Server
See the section in this guide on [socket enhancements](#), on [page 15](#) for Model 204. The Model 204 server must be configured to support IPv6 addresses to take advantage of this feature.

Chapter 8: New and updated messages

Topics covered in this chapter

Messages

This chapter describes error and informational messages that have been added or enhanced in this release.

Messages

The following Model 204 messages have been added or enhanced in Version 7 Release 4.0.

FIM169:

Field is Date time stamp, fieldname cannot be changed

Explanation:

The Datetime Stamp field corresponds to the value of DTSMN and cannot be changed.
In previous releases, this message was displayed with number **FIM159**

FIM170:

WARN: There is no valid Datetime stamp field defined in file.

Explanation:

The user is attempting to use a Datetime Stamp feature, but the file does not have the field defined.
In previous releases, this message was displayed with number **FIM160**

XRF117:

INTERNAL LISTING ERROR. TYPE=xxxx. CONTACT CCA SUPPORT.

Explanation: If this message appears in the last line of your CCAPRINT output, check for incorrect input in:

```
//CCAPROCI DD *
```

Edit the XREF job while in the XREF subsystem (PF4=EDITjob) and ensure that the first lines in the XREF step are exactly these and nothing else:

```
//XREF EXEC PGM=M204XREF  
//CCAPROCI DD *  
//*M204XREF  
/*
```

Then resubmit the job. If the error reappears, contact Technical Support and report the TYPE code.

0061:

SMF SYSTEM ID = nnn, JOB NAME = xxx, STEP NAME = yyy, JOB STEP = zzz, JES ID = jjj

Explanation: This informational message is sent to CCAPRINT and CCAAUDIT during Model 204 initialization. The message element values vary by operating system:

Message element	Value		
	on z/OS	on CMS	on DOS
SMF system ID	CPU identifier	CMS	NONE
Job name	Name on the JOB card in the startup JCL	CMS userid of the host machine	Unspecified
Step name	Step name in the startup JCL	Unspecified	Unspecified
Job step	Job step name in the startup JCL	Unspecified	Unspecified
JES ID	JES (Job Entry Subsystem) job number identifier	Unspecified	Unspecified

Return Code: I 0 0 AUDITAD

0172:

END OF UPDATE nn AT hh:mm:ss.th

Explanation: The update successfully completed at the specified time, and a log of updates performed was written to the journal data set (if roll forward logging is active).

Return Code: I 0 0 AUDITAD NOTERM

0173:

START OF UPDATE nn AT hh:mm:ss.th

Explanation: This informational message is written at the start of each update unit.

Return Code: I 0 0 AUDITAD NOTERM

0410:

INVALID USE OF FEW VALUED OR MANY VALUED

Explanation: In the REDEFINE FIELD command, the only situation where you can specify the attributes MV (many-valued) or FV (few-valued) is when a field is being redefined from NON-FRV NON-CODED to FRV NON-CODED or if the field was originally defined as CODED. See the *Rocket Model 204 Parameter and Command Reference*.

Response: Correct and retry.

Return Code: C 0 4 AUDITER

0443:
TABLE %C FULL IN %F

Explanation: You attempted to add data to Table D for the file indicated, but the table is full. If the file is a transaction back out file, Model 204 will initiate back out for the transaction under way and cancel the current request. The file can still be used. If the file is not a transaction back out file, Model 204 logs you out and closes any files you had open. Any files that were being updated are marked physically inconsistent.

Response: Notify your file manager.

File Manager Response:

If the file was a transaction backout file, there will not be anything wrong with the file. You might want to expand Table D, because it is probably close to full.

If the file is not a transaction back out file, you need to recover the file using restart recovery (roll back and roll forward), the file reorganization procedure described in the *Model 204 File Manager's Guide*, or the RESTORE command. The RESTORE command restores from a dumped copy of the file and increases the size of Table D for the file named in the message.

Return Code: E 48 48 CANCEL AUDITAD

0797:
BSIZE*BRECPPG EXCEEDS MAXIMUM VALUE

Explanation: The product of BSIZE and BRECPPG must not exceed 16,777,216. This specifies the maximum number of records allowed in a single Model 204 file. The CREATE or INCREASE command is rejected.

File Manager Response: Retry the command specifying values for BSIZE and BRECPPG whose product does not exceed the limit. Refer to the *Model 204 File Manager's Guide* for more information about file sizes.

Return Code: C 0 4 AUDITER

1040:
RUN ABORTED

Explanation: The X'01' bit of the RCVOPT parameter has been set to enable checkpoint logging. The checkpoint dataset cannot be opened. The run is terminated.

System Manager Response: Correct the RCVOPT parameter on User 0's parameter line, if incorrect, and retry. Under CMS, check for a CHKPOINT FILEDEF. Under OS, check for a CHKPOINT DD statement. Under DOS, check for CHPNT DLBL and EXTENT statements. Fix and retry.

Return Code: E 4 4 AUDITMS

1165:
INCLUDE TOO DEEP

Explanation: Your request tried to include more levels than specified by the MAXINCL parameter.

Response: Rearrange the included procedures by combining procedures or by placing more than one include in a procedure, or increase the value of the MAXINCL parameter. See the *Rocket Model 204 Parameter and Command Reference*.

Return Code: C 0 4 AUDITER

1222:
filename IS FULL

Explanation: An attempt has been made to enqueue on a file resource for a full file, to start a transaction for a full file, or to store a record in a full file.

Response: Notify your file manager.

File Manager Response: Analyze previous messages to find out which table became full:

- If it is TABLE D or TABLE X, try to increase it using the INCREASE TABLED or INCREASE TABLEX command.
- If it is TABLE B and the file is not a hash or sort file, try to increase TABLE B using the INCREASE TABLEB command.
- In all other cases or when an increase is not possible, reorganize the file with larger appropriate table(s).

However, the file might be physically inconsistent as well. In this case, refer to the discussion in the *MODEL 204 File Manager's Guide* section on maintaining file integrity.

Return Code: E 48 48 AUDITAD

1264:
PREALLOCATED FIELD BLOCK TOO BIG IN FILE filename (WITH BRLIMSZ = nnnn)

Explanation: The total preallocated length of a preallocated field (a field with the OCCURS attribute) is the length of the field multiplied by the value of the OCCURS attribute. The preallocated block size is the sum of the preallocated length of each preallocated field in the file.

You have attempted to define a field with the OCCURS attribute. When the total preallocated length of the field is added to the current length of the preallocated block, the preallocated block exceeds its size limit. The preallocated block size limit for any file organization is (PAGESZ - 53). The field definition fails.

For files created with the RECRDOPT = 1 option, BRLIMSZ defines maximum size for the sum of the lengths of all preallocated fields + 2 + 3 + (1 : only if XSIZE > 0). BRLIMSZ has been exceeded and the field definition fails.

Response: Make this field a non-preallocated field, re-evaluate the fields that are preallocated in this file, or CREATE the file without the RECRDOPT = 1 option.

Return Code: C 0 4 AUDITER

1268:
BUG .. SBNU\$+nnnn: DAMAGED RECORD - recnum IN FILE filename

Explanation: A record in a Model 204 file has been damaged. File reorganization might be required. First, attempt to print the contents of the record with PAI in an FRN loop. That will cause another snap but might provide enough information to reconstruct the record. Then delete the record with a logical delete and reorganize the file when possible.

Response: Notify your system manager.

System Manager Response: Contact Technical Support. Have the following documentation available:

- Audit trail
- Snap
- User Language request or the host language program of the user who received the message

Return Code: E 0 4 SAVE DUMP SNAP AUDITAD

1290:
INVALID PRIORITY SPECIFICATION

Explanation: You issued a PRIORITY command with a class other than LOW, STANDARD, or HIGH or an invalid numeric value for priority. Numeric values must be between 0-255 inclusive.

System Manager Response: Correct the priority value and retry the command.

Return Code: C 0 4 AUDITMS

1992:
RECOVERY: PROCESSING ROLL FORWARD BLOCK# %C %C

Explanation: This message is informational only. It details the sequence number and date/time stamp of the current journal block being processed by roll forward. It will be issued each time the minute value in the time stamp changes - that is, once for each minute of data processed by roll forward. This provides a kind of thermometer on roll forward's progress and confirms that roll forward is actively reapplying updates.

Return Code: I 0 0 OPR

2119:
AT-MOST-ONE CONSTRAINT VIOLATION IN RECORD *recordnumber*, FIELD IGNORED: *fieldname* = *value*

Explanation: An attempt was made to store, add or insert a second occurrence of an AT-MOST-ONE field in the record indicated. The second occurrence, also indicated in the message, is not added/inserted in Table B, or in any index (if applicable). The requested update is not performed. This message is only issued when no ON FIELD CONSTRAINT CONFLICT unit exists for the request.

When the failed request was from User Language, the request will be canceled and its active transaction, if any, will be backed out. For failed requests from Host Language Interface, the message is accompanied by Return Code 202 (without request cancellation). For failed requests during Fastload (FLOD or FILELOAD), Fastload continues, but the file is marked "logically inconsistent" (see the FISTAT parameter).

Response: User Language Procedures which update AT-MOST-ONE fields should always contain ON FCC units to handle AT-MOST-ONE conflicts (\$UPDSTAT code 2) when they occur. Similarly, Host Language Interface programs should code routines to handle AT-MOST-ONE violations, as indicated by the Return Code 202. Fastload must be corrected, following a RESET of FISTAT (see below), to properly load AT-MOST-ONE fields.

File Manager Response: From User Language or Host Language Interface, no response is necessary.

From Fastload (FLOD or FILELOAD), the FISTAT parameter will have been set to x'40', "logically inconsistent", as a warning that the state of file integrity is questionable depending on whether or not the failed request "cascaded" invalid Fastload decisions and results. Consult with the User to determine the logical consistency ramifications of the particular Fastload in question. RESET FISTAT to x'00' after a plan for corrective action, if any, has been determined.

Return Code: C 0 4 AUDITER

2606:
ONLY *number parameter* LICENSED, RUN TERMINATED

Explanation: The specified number of parameter=IODEV or parameter=NMPSUBS has exceeded the number that your contract allows.

System Manager Response: Perform one of the following actions: reduce the quantity of the specified IODEVs, reduce the number of NMPSUBS, or contact Technical Support to arrange for a greater supply.

Return Code: C 80 80 AUDITER

2677:

FILE *filename* ALL FREE PAGES ALLOWED FOR COMPACTION HAVE BEEN USED. COMMAND COMPACTB ENDS.

Explanation: The compactor has used all free pages it was allowed to use.

File Manager Response: Increase the percentage of free pages to use by resetting the FREE argument on the COMPACTB command.

Return Code: C 0 4 AUDITER

2684:

CHECKPOINT CONFIGURATION CONFLICT -
CHKPNTS IS OPEN SO CPTIME MUST BE NON-ZERO
| CHKPNTS IS OPEN SO DKUPDTWT MUST BE ZERO
| A NON-TBO FILE WHICH PRE-IMAGES BUT DOES NOT JOURNAL HAS BEEN OPENED FOR UPDATE,
SUB-TRANSACTION CP IS NOT ALLOWED
| CHKPNTS IS NOT OPEN AND CPTYPE = 1
| CHKPNTS IS NOT OPEN AND CPTS IS NON-ZERO
| {CHKPOINT|CHKPNTS} CONTAINS A RING DEFINITION
| {CHKPOINT|CHKPNTS} HAS MORE THAN 16 LEVELS OF STREAM DEFINITION RECURSION
| {CHKPOINT|CHKPNTS} CONTAINS A CMS FORMATTED FILE
| CHKPNTS IS OPEN AND JOURNALLING NOT ACTIVE
| CPTYPE = 1 IS INCOMPATIBLE WITH NUSERS = 1

Explanation: Parameters associated with checkpoints are consistency checked when checkpoint streams are opened, at the beginning of Online processing and also during the processing of a RESET command. The indicated error has been found with the result that the Online either failed to come up or the RESET command was rejected.

Note

If CPTYPE=1, then neither CHKPOINT nor CHKPNTS can be defined as a ring stream or a CMS formatted data set.

System Manager Response: If the Online failed, change the appropriate CCAIN parameter line entries and/or CHKPOINT and CHKPNTS stream definitions. If the RESET command failed, take appropriate action.

Return Code: I 0 0 AUDITMS

2702:

FILE *filename* MISMATCH BETWEEN TABLE %C QUEUE LENGTH AFTER REBUILD AND NUMBER OF PAGES ADDED TO THE QUEUE: %C

Explanation: The BLDREUSE command with option NEW found that the new queue length (BQLEN) is not equal to the number of pages added to the queue during queue rebuild.

File Manager Response: Run the command again. If the error persists, report it to Technical Support.

Return Code: C 0 4 AUDITER

2749:

FILE *filename* NUMBER OF BASIC RECORDS PROCESSED: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2750:

FILE *filename* NUMBER OF EXTENSION RECORDS BEFORE COMPACTION: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2751:

FILE *filename* NUMBER OF EXTENSION RECORDS AFTER COMPACTION: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2752:

FILE *filename* NUMBER OF NOT PROCESSED (LOCKED) RECORDS: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2753:

FILE *filename* NUMBER OF FREE PAGES USED: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2754:

FILE *filename* NUMBER OF DELETED LOGICALLY DELETED RECORDS: %C

Explanation: This informational message indicates final statistics for the COMPACTB command.

Return Code: I 0 0 AUDITMS

2755:

FILE *filename* UNEXPECTED ERROR DURING PHYSICAL DELETE. COMPACTION ENDS.

Explanation: An error happened during COMPACTB command processing with the DELETE option.

File Manager Response: Contact Technical Support and provide the required documentation.

Return Code: C 0 4 AUDITER

2763:

FILE *filename* TOTAL NUMBER OF TABLE %C PAGES PROCESSED : %C

Explanation: This informational message indicates final statistics for the BLDREUSE command.

Return Code: I 0 0 AUDITMS

2764:

FILE *filename* NUMBER OF TABLE %C PAGES ADDED TO THE REUSE QUEUE: %C

Explanation: This informational message indicates final statistics for the BLDREUSE command.

Return Code: I 0 0 AUDITMS

2765:**FILE *filename* TABLE %C REUSE QUEUE LENGTH BEFORE REBUILD: %C****Explanation:** This informational message indicates final statistics for the BLDREUSE command.**Return Code:** I 0 0 AUDITMS

2766:**FILE *filename* TABLE %C NUMBER OF PAGES THAT WERE ON THE REUSE QUEUE: %C****Explanation:** This informational message indicates final statistics for the BLDREUSE command.**Return Code:** I 0 0 AUDITMS

2767:**FILE *filename* TABLE %C REUSE QUEUE LENGTH AFTER REBUILD: %C****Explanation:** This informational message indicates final statistics for the BLDREUSE command.**Return Code:** I 0 0 AUDITMS

2799:
\$SNDMAIL ERROR %C**Explanation:**

An error has occurred while executing the \$SNDMAIL function.

If there is a parameter error, the variable text is IN PARAMETER *parametername errordetail*.

If the error arises from a socket call, the variable text is *tcperrorcode* ON *socketcall*, where *socketcall* is one of the following calls:

- READ
- SELECTEX
- CONNECT
- WRITE
- REGISTER
- SHUTDOWN
- CLOSE

In other cases the variable text is one of the following descriptions:

- GETTING STORAGE
- IN IMAGE PROCESSING
- WITH TCP LINK
 - SOCD NOT RELEASED
- READING DATA - TRUNCATION
 - SOCKET NOT FOUND
 - FUNCTION NOT SUPPORTED ON THIS SYSTEM
 - NO SRCD
 - TOO MANY SOCKETS
- WITH UNIVERSAL BUFFER
- READING FROM SMTP SERVER
- WRITING TO SMTP SERVER
- USING GETHOSTBYNAME LOOKUP

User Response: Check the error text. In most cases the error will be due to an error in the parameters. If this is a GETHOSTBYNAME error, check that the supplied client and server names are correct.

Return Code: E 0 4 AUDITMS

2809:
FILE *filename* WRONG SEGMENT %C FOR PAGE %C

Explanation: Table E bitmap page has an incorrect number.

File Manager Response: You must regenerate the file.

Return Code: E 0 4 AUDITMS

2810:
FILE *filename* DESCRIPTOR NOT FOUND FOR OBJECT AT %C RECORD NUMBER %C

Explanation: A discrepancy is found between the object in Table E and the Table B record that points to the object.

File Manager Response: You must regenerate the file.

Return Code: E 0 4 AUDITMS

2811:
FILE *filename* NUMBER OF MOVED OBJECTS: %C

Explanation: This informational message indicates the final statistics for the Compactor.

Return Code: I 0 0 AUDITMS

2812:
FILE *filename* NUMBER OF MOVED PAGES: %C

Explanation: This informational message indicates final statistics for the Compactor.

Return Code: I 0 0 AUDITMS

2813:
FILE *filename* NUMBER OF RECORD LOCKING CONFLICTS: %C

Explanation: This informational message indicates final statistics for the Compactor.

Return Code: I 0 0 AUDITMS

2814:
FILE *filename* NUMBER OF MULTISEGMENT OBJECTS: %C

Explanation: This informational message indicates final statistics for the Compactor.

Return Code: I 0 0 AUDITMS

2815:
FILE *filename* TOTAL NUMBER OF GAPS: %C

Explanation: This informational message indicates final statistics for the Compactor.

Return Code: I 0 0 AUDITMS

2816:
FILE *filename* TOTAL GAP SIZE: %C

Explanation: This informational message for the TABLEE command indicates the total gap size in pages.

Return Code: I 0 0 AUDITMS

2817:
FILE *filename* SEGMENT: %C NUMBER OF GAPS: %C TOTAL GAP SIZE: %C

Explanation: This informational message for the TABLEE command indicates the number of gaps and the total gap size measured in pages.

Return Code: I 0 0 AUDITMS

2818:
FILE *filename* RECORD %C HAS NO FIELD POINTING TO OBJECT AT %C

Explanation: A discrepancy is found between the Table E object and the Table B record pointing to that object.

File Manager Response: You must regenerate the file.

Return Code: E 0 4 AUDITMS

2819:
FILE *filename* RECORD %C NOT FOUND FOR OBJECT AT %C

Explanation: A record in Table B that points to the object is not found.

File Manager Response: You must regenerate the file.

Return Code: E 0 4 AUDITMS

2820:
FILE *filename* DISCREPANCY FOUND FOR %C PAGE %C

Explanation: A discrepancy is found for the page.

File Manager Response: Regenerate the file.

Return Code: E 0 4 AUDITMS

2821:
FILE *filename* OBJECT AT %B STRETCHES PAST EHIGHPG

Explanation: A discrepancy is found in Table E with incorrect EHIGHPG.

File Manager Response: Regenerate the file.

Return Code: E 0 4 AUDITMS

2823:
FILE *filename* BUG %C DURING OBJECT MOVE

Explanation: A problem occurred while compacting an object.

File Manager Response: Run recovery to restore the file. Contact Technical Support and provide the required documentation.

Return Code: E 0 4 AUDITAD SNAP

2826:
'%F VERSION IS PRE 7.1. COMPACTION NOT ALLOWED

Explanation: An attempt was made to compact a file created prior to version 7.1.

File Manager Response:
Reorganize the file using Version 7.1.

Return Code: C 0 4

2829:
REORGANIZE %F TO IMPROVE EFFICIENCY FOR TABLE E ALLOCATION

Explanation: All Table E append pages have been allocated.

Programmer Response: Reorganize the file to improve efficiency when storing new Large Object data.

Return Code: I 0 0

2834:
ADDRESS LOOKUP FAILED FOR %C

Explanation: You have specified a domain name, but TCP/IP has been unable to convert this to an internet address.

Response: Check that the name you entered is correct and known to your TCP/IP name server.

Return Code: I 0 0

2835:
%F TABLE %C INCREASED BY %C PAGES AUTOMATICALLY

Explanation: This informational message indicates that a table was automatically increased when it became full. A response is not required.

Return Code: I 0 0 NOTERM

2836:
PARAMETER name INVALID - RESET TO value

Explanation: This is an informational message only. A parameter was specified with an incorrect value. It has been reset to an internally calculated value which will accommodate other parameter settings. See the description of the parameter named to see what that calculation involves.

System Manager Response: The parameter has been reset. No action is required.

Return Code: I 0 0

2837:
NO LOGICAL CONNECTION BETWEEN PROCESS %C AND REMOTEID %C

Explanation: There is an incoming request to start the named process, but the process is not one of the processes connected by definitions to the named remote ID. It could be that an incoming request to start the named process is arriving over an unexpected session.

Response: Check the definitions associated with the process, and also that the client application is using the correct VTAM LU name.

Return Code: E 0 4 LAST SAVE NOTERM

2840:
STATEMENT HAS TOO MANY INTERMEDIATE RESULTS

Explanation: This error generally occurs only with a STORE RECORD statement that references more than 127 subscripted values in a User Language array. An internal table of 512 bytes is allocated by the User Language compiler to hold temporary variables. This table consists of one fullword counter and 127 fullwords of data. Each of those 127 fullwords represents a temporary variable. This change in the handling of temporary variables improves performance and reduces space requirements in VTBL and STBL.

Response: Break up the STORE RECORD statement into multiple parts as in the following example:

```
STORE:
STORE RECORD
    FIELDX = %ARRAY(1)
    FIELDX = %ARRAY(2)
    ...
    FIELDX = %ARRAY(127)
END STORE
FRN IN STORE
    FOR %X FROM 128 TO 999
        ADD FIELDX = %ARRAY(%X)
    END FOR
END FOR
```

etc.

Return Codes: C 0 4 ECHO

2864:
FULL SEARCH OF ORDERED INDEX PERFORMED

Explanation: Pattern optimization cannot be performed; the entire Ordered Index is searched. To perform pattern optimization the following conditions must be met:

- The pattern and all of its members do not begin with set characters (parentheses), range characters (hyphens), or hexadecimal value 'FF'.
- There are no wildcard, placeholder, numeric digit, or alphabetic pattern characters (* + # @) beginning the pattern or any member in it.

Return Code: I 0 0 NOTERM

2883:
MIX OF JOURNAL VERSIONS IN RECOVERY JOURNAL

Return Code: E 0 4 SNAP OPR

2887:
ABANDON OPTION INVALID FOR THIS TYPE OF LINK

Explanation: A CLOSE LINK ABANDON command was issued for a link that is not a Horizon VTAM link. The ABANDON option is valid only for Horizon VTAM links.

Return Code: C 0 4

2888:
LINK FLAGS %X LOCKS HELD %X REMOTE PARTNER %C

Explanation: This informational message may be issued during the processing of a CLOSE LINK ABANDON command. The message displays the flags set and locks held for each session affected by the command and also the name of the partner LU.

Return Code: I 0 0

2889:
USER %C NO %C SESSION ABANDONED FOR LINK %C

Explanation: This informational message displays the name of the user and the user number for each user affected by a CLOSE LINK ABANDON command, and also the name of the link.

Return Code: I 0 0

2890:
LINK %C CLOSED NORMALLY. IT CAN BE RE-OPENED

Explanation: A CLOSE LINK ABANDON command was issued for this link. The command ran to completion, so it is safe to re-open and re-use the link.

Return Code: I 0 0

2891:
LINK %C ACPST WAITING ON ECB %X RPL %X

Explanation: This informational message may be output when Horizon is trying to close a link with the ABANDON option, or when it runs out of work areas queueing work onto the AC PST. It attempts to clear the backlog of work.

The message shows the address of the ECB that the AC PST is waiting on, and the beginning of the AC PST's VTAM RPL, in hexadecimal.

Return Code: I 0 0

2892:
SQEDS IN CHAIN %C LAST ONE IS %X

Explanation: This informational message may be output when Horizon is trying to close a link with the ABANDON option, or when it runs out of work areas queueing work onto the AC PST. It attempts to clear the backlog of work.

The message shows the number of elements in the work queue, and the element, in hexadecimal, at the top of the queue.

Return Code: I 0 0

2893:
%C IS NOT A VALID OPTION FOR THE OPEN LINK COMMAND

Explanation: Valid parameters for the OPEN LINK are the link name, and then optionally REPLACE followed by the name of a link name that has been abandoned with the CLOSE LINK *linkname* ABANDON command.

Return Code: C 0 4

2894:
LINK %C CANNOT BE REPLACED - IT IS NOT A VTAM LU 6.2 LINK

Explanation: You have entered an OPEN LINK *linkname* REPLACE *oldlinkname* command, but this is valid only if both links are VTAM LU6.2 (Horizon) links.

Return Code: C 0 4

2895:
LINK %C HAS NOT BEEN ABANDONED, IS OPEN, OR WAS CLOSED OK

Explanation: You have entered the OPEN LINK *linkname* REPLACE *oldlinkname* command, but the link being replaced (*oldlinkname*) is not in an abandoned state. Either the CLOSE LINK *linkname* ABANDON command was not used, or it was used and the link closed successfully.

Return Code: C 0 4

2896:
LINK %C HAS REPLACED LINK %C

Explanation: You have entered the command OPEN LINK *linkname* REPLACE *oldlinkname* and this command has completed successfully.

Return Code: I 0 0

2897:
%C %C NOW USES LINK %C

Explanation: You have entered the command OPEN LINK *linkname* REPLACE *oldlinkname* and the PROCESSGROUP or REMOTE entry named in the message has successfully been modified to point to the new link.

Return Code: I 0 0

2901:
FOPT=X'10' (DTS) IS NOT SUPPORTED DURING FLOD/FILELOAD

Explanation: Neither FLOD nor FILELOAD support the date/time stamp (DTS) feature. All previous DTS values will be restored but date/time stamps from the actual FLOD or FILELOAD will not be stored.

File Manager Response: Either remove the DTSFN parameter from the CCAIN STREAM or reset FOPT to turn off the x'10' option and rerun the job.

Return Code: E 0 8

2902:
ECTG SUPPORTED; XMEMOPT RESET FROM X'%X' TO X'%X'

Explanation: Your system supports the ECTG (extract timer) instruction. This instruction is more efficient than the TIMER PC call. Therefore, the XMEMOPT parm is adjusted to remove the TIMER PC option. In addition, the x'04' bit is forced on to insure that the XMEMSVC is used for initialization.

Response: Informational only. No action needed.

Return Code: I 0 0

2904:
INSUFFICIENT CCATEMP PAGES FOR BACKOUT DATA

Return Code: E 60 60 OPR

2905:
IBM SYSTEM INTERFACE MACRO %C WARNING, RETURN CODE = %X, REASON CODE = %X

Explanation: Model 204 has trapped a warning in the named IBM system interface macro. The return code and reason code are displayed.

System Manager Response: The return code and reason code may be found within related IBM documentation. The message has no effect on M204 processing.

Return Code: I 0 0 AUDITMS

2906:
IEANTCR CANNOT BE LOADED — CODE %C RC %C

Explanation: (z/OS only) When Model 204 starts, it tries to create a NAME/TOKEN pair at system level, using the IBM-supplied services program IEANTCR. This message shows that IEANTCR could not be loaded. It displays the abend code from the LOAD macro and the register 15 return code.

Return Code: E 0 4 AUDITMS

2907:
LINK %C REJECTED ACCEPT. SOCKADDR IS %X

Explanation: A connection request was received on the connection named in the message, but it was rejected because the family is not supported. The family is neither 2 (IPV4 addressing - AF_INET) nor 19 (IPV6 addressing - AF_INET6).

The message includes the full socket address structure returned from the ACCEPT call.

Return Code: I 0 0 AUDITMS

**2908:
PROCESSGROUP NOT FOUND FOR REMOTE %C**

Explanation: A CNOS server has failed to find a matching processgroup after receiving an incoming Attach request on the remote name specified in the message.

Check in particular that the sessiongroup name is the same on the processgroup definition and the sessiongroup definition.

Return Code: E 0 4 AUDITMS

**2909:
IEANTDL RETURN CODE %C — NAME/TOKEN NOT DELETED**

Explanation: (z/OS only) When Model 204 ends, it tries to delete a NAME/TOKEN pair at system level, using the IBM-supplied services program IEANTDL. This message shows that the invocation of IEANTDL has failed with the code given.

The code can be found in the *IBM Assembler Services Reference Manual*.

Return Code: E 0 4 AUDITMS

**2910:
IEANTDL CANNOT BE LOADED — CODE %C RC %C**

Explanation: (z/OS only) When Model 204 ends, it tries to delete a NAME/TOKEN pair at system level, using the IBM-supplied services program IEANTDL. This message shows that IEANTDL could not be loaded. It displays the abend code from the LOAD macro and the register 15 return code.

Return Code: E 0 4 AUDITMS

**2911:
ATTEMPTED CONNECT FROM PROCESSGROUP %C TO IPV6 SERVER %C, BUT LINK %C IS IPV4**

Explanation: The link named in the message is defined as using TCP IPv4 addressing, but the processgroup is attempting to connect to a remote server using IPv6 addressing. This will usually fail with TCP/IP socket error code 121 (EINVAL).

The link will be IPv4 if:

- the localid is a symbolic name that has no IPv6 address, or
- the host machine is not configured for IPv6 addressing, or
- LOOKUP=IPV4ONLY has been specified on the DEFINE LINK command.

The processgroup will attempt to connect to a remote IPv6 server if:

- its remoteid resolves to an IPv6 address and LOOKUP=IPV4 has not been specified on the DEFINE PROCESSGROUP command, or
- its remoteid is an explicit colon-separated IPv6 address.

Return Code: E 0 4 AUDITMS

**2912:
IEANTCR RETURN CODE %C - NAME/TOKEN NOT CREATED**

Explanation: (z/OS only) When Model 204 starts, it tries to create a NAME/TOKEN pair at system level, using the IBM-supplied services program IEANTCR. This message shows that the invocation of IEANTCR has failed with the code given.

The code can be found in the *IBM Assembler Services Reference Manual*.

Return Code: E 0 4 AUDITMS

**2913:
"NEW" OPTION REQUIRES SINGLE USER RUN**

Explanation: The BLDREUSE command with the NEW option can only be executed in a single user run (NUSERS=1).

User Response: Bring up an Online with NUSERS=1 and re-execute the command.

Return Code: E 0 4 AUDITMS

**2914:
SCHDOPT INDICATION OF SERVER SWAPPING DONE BY ZIIP IS ONLY VALID WHEN CCASERVER IS IN MEMORY**

Explanation: Setting of SCHDOPT=x'80' to allow zIIPs to do server swapping should only be used when CCASERVER is in memory (servers swapped into memory). Otherwise, the SCHDOPT setting of x'80' is reset and this informational message is issued.

System Manager Response: Correct either SCHDOPT setting or servers in memory indication.

Return Code: I 0 0

**2915:
IGNORE SPECIFIED FOR FILE *name message***

Explanation: This is an informational message indicating that during RESTART recovery the named file was requested to be omitted from processing by use of the IGNORE option.

There are two possible *message* endings:

- BUT RECOVERABLE UPDATES WERE AVAILABLE
There were updates against the file during the recovery period. Those updates were not applied.
- WHICH HAD NO RECOVERABLE UPDATES
No updates were found for the file and thus the IGNORE option was unnecessary.

User Response: No response is necessary.

Return Code: I 0 0 OPR

**2916:
SERVNSSZ REQUIRES AT LEAST ONE AREA TO BE SET IN SERVNSA**

Explanation: When the size of a non-swappable server area is indicated, at least one non-swappable server area must be indicated in SERVNSA.

Operator Response: Correct the SERVNSA parameter setting.

Return Code: E 0 4

**2917:
SERVER DATASET *name* IS ALLOCATED ABOVE 64K CYL. SERVERS ABOVE 64K CYL ARE NOT SUPPORTED.**

Explanation: An attempt has been made to open a server data set allocated on an EAV volume above the 64K cylinder boundary.

System Manager Response: Allocate a server data set below 64K cylinder or use servers in storage.

Return Code: E 0 4

2919:
HWM MBYTES ATB STORAGE ALLOCATED = %C

Explanation: This informational message, issued during Model 204 termination, indicates the high water mark (HWM) of megabytes (MBYTES) of ATB (above-the-bar) storage allocated during the run.

No response is required.

Return Code: I 0 0 AUDITAD, NOTERM

2920:
NON ZERO SERVNSSZ IS REQUIRED

Explanation: When non-swappable server areas are indicated in SERVNSA, the parameter SERVNSSZ must have a non-zero value.

Operator Response: Correct the SERVNSSZ parameter setting.

Return Code: E 0 4

2921:
NON SWAPPABLE SERVER AREAS ARE NOT SUPPORTED IN VSE

Explanation: Non-swappable server areas are not supported in z/VSE.

Operator Response: Remove the parameters for non-swappable server areas.

Return Code: E 0 4

2922:
SERVNSA BIT SETTING IS NOT VALID

Explanation: The bit setting in SERVNSA is not valid. For Model 204 version 7.4.0 the valid bits are: 02000000.

Operator Response: Correct the SERVNSA setting.

Return Code: E 0 4

2923:
SPACE REQUIRED FOR NON SWAPPABLE SERVER AREAS IS LARGER THAN SERVNSSZ

Explanation: The total amount of non-swappable server areas is larger than the non-swappable area size defined by the SERVNSSZ parameter. The total amount equals a sum of non-swappable area sizes plus 8 bytes per area.

System Manager Response: Either increase the value of SERVNSSZ or decrease the non-swappable table size.

Return Code: E 80 80 AUDITAD OPR

2925:
HWM MBYTES STORAGE ALLOCATED WITH 1MB PAGES = %C

Explanation: This informational message, issued during Model 204 termination, indicates the high water mark (HWM) of megabytes (MBYTES) of above-the-bar storage allocated with 1MB pages during the run.

No response is required.

Return Code: I 0 0 AUDITAD, NOTERM

2926:
1 MB PAGE FEATURE IS NOT AVAILABLE. PARAMETER ZPAGEOPT IS RESET TO ZERO.

Explanation: The virtual storage 1MB page feature is not available on this system. This informational message is issued during initialization and is visible in CCAAUDIT and CCAPRINT only.

System Manager Response: Remove the ZPAGEOPT parameter from CCAIN.

Return Code: I 0 0 AUDITMS

2928:
MAXIMUM SOCKET USED ON TCP/IP LINK %C WAS %C

Explanation: This informational message shows the highest socket number used on this Horizon TCP/IP link. The first %C is the link name, for example LINK1, and the second %C is the maximum socket used on that link at the time it was closed.

On TCPSE protocols, this information is useful for tuning the CONNECTIONS parameter on DEFINE LINK. If the maximum socket number is close to the value of CONNECTIONS, you might want to increase CONNECTIONS.

On other protocols (such as TCPVSE, used on z/VSE), high socket numbers can be used even if there are few simultaneous connections (sessions) in use at any time.

Return Code: I 0 0 AUDITMS

2930:
UNABLE TO ALLOCATE STORAGE OBJECT FOR *parameter*

Explanation: The parameter settings indicate that Model 204 should allocate storage from above the bar (ATB). However, this version of the operating system does not support 64-bit virtual storage.

System Manager Response: Review the parameter settings and remove any ATB specifications.

Return Code: I 0 0 OPR

2931:
OPERATING SYSTEM DOES NOT SUPPORT *parameter*

Explanation: The designated parameter was set but this version of the operating system does not support such an environment.

System Manager Response: Review the parameter settings and remove any conflicts.

Return Code: I 0 0 OPR

2932:
SPACE REQUIRED FOR NON SWAPPABLE SERVER AREAS IS LARGER THAN SERVNSSZ

Explanation: A size request for a table that is kept in the non-swappable server area cannot be satisfied because there is not enough above-the-bar storage space as set by the SERVNSSZ parameter. This error could occur when a UTABLE command is issued. The command is rejected and the table size is left unchanged.

User Response: Decrease the amount of increase in UTABLE command.

System Manager Response: If required, increase the non-swappable server area size (SERVNSSZ).

Return Code: E 0 4

Index

1 MB virtual pages	40
MONITOR GSTORAGE	71
64-bit storage	41

A

AAI	31
ABANDON	
with CLOSE LINK	69
above the bar	41
Access List Entry Token	35
ALETs	
and XDM	35
ALLOCATE command	68
AMPSUBS	36
AMPSUBZ	36
ANDIF	34
ANDIF and ORIF	
compatibility issue	52
ASTRPPG parameter	57
ATB	41
AUDIT ALL INFORMATION	31
audit trail	
writing to	31

B

BLDREUSE NEW	
compatibility issue	53
BLOB	48
limitations	50
SQLBUFSZ	50

C

CCAAUDIT	
writing to	31
CCACAT file	
BLOB/CLOB data types	50
CCATEMP	67
character string functions	21
CLOB	48
limitations	50
SQLBUFSZ	50
CLOSE LINK command	69
coded and binary keywords	43
commands	
ALLOCATE	68
CLOSE LINK	69
COMPACTB	69
DEFINE DATASET	70
DEFINE QUEUE	51
LOGCTL	70
MONITOR TASKS	72
OPEN LINK	74

RESTART	74
USE PROC	74
COMPACTB command	69
compatibility issues	
\$DSN and \$DSNNUM	53
ANDIF and ORIF	52
BLDREUSE NEW	53
CREATE	53
Horizon IPv6	
and symbolic addresses	54
IPv6 addressing	20
multistep FLODs	53
OPEN	53
STORE RECORD	54
Connect*	
and IPv6 addressing	80
Connect* compatibility	12
Connect* for .NET Framework	
BLOBs and CLOBs	79
documentation	79
log file location	79
software requirements	79
Visual Studio and	79
Connect* for JDBC	
BLOBs and CLOBs	78
log file location	78
software requirements	78
contact technical support	3
CPUID zap	12
CREATE	
with long DDNAME	53
CREATE TABLE statement	49
CUSTOM parameter	57
CUSTOM=10	58
CUSTOM=11	58

D

DDNAME	
long	53
DEFINE DATASET command	70
DEFINE LINK	
and TCP/IP Horizon	77
DEFINE LINK command	19
DEFINE PROCESSGROUP	19
and TCP/IP Horizon	77
DEFINE QUEUE command	51
DELETE statement	50
demonstration files	51
\$DEPAD function	21
\$DEPADL function	21
\$DEPADR function	21
\$DSN and \$DSNNUM	
compatibility issues	53
DSOPT parameter	58

E

EATTR	68, 70
EAV	36
ECF	21
ECPSIZE	21
EQ VALUE expression	26, 29
EQ VALUE retrieval	26, 29
EVCP	13
expression	
fieldname=value	24
Expression	
with POINT\$ condition	27
with SFGE\$ condition	28
with SFL\$ condition	28
Extended Address Volumes	36
External Call Facility	21

F

features overview	7
FILE RECORDS	25
files	
in-memory	44
FIND	26
POINT\$ condition	27
SFL\$ and SFGE\$ conditions	28
FIND ALL VALUES statement	27
FLODs	
multistep	
compatibility issue	53
FOR EACH RECORD IN ORDER BY statement	28
FOR EACH RECORD WHERE	29
FOR EACH VALUE	30
FOR RECORD NUMBER	30
FRN statement	30
FRV statement	30
FTBL	
above the bar	41

H

hardware requirements	12
Horizon	
and sockets	76
IPv6 address support	19
IPv6 addressing	76
Horizon IPv6 addressing	
and symbolic addresses	54

I

IGNORE option	
and RESTART	74
IGNORE option for RESTART	47
IIPHONORPRIORITY	39
in-memory file storage consumption	
viewing	45
in-memory files	44

ALLOCATE	46
and DEFINE DATASET	70
DEFINE DATASET	46
freeing	45
opening and closing	46
INSERT statement	49
installation format	12
installation requirements	12
IPv4-mapped addresses	17
IPv6 addressing	
\$SNDBMAIL	14
\$SOCKET	15
\$SOCKETX	17
and Connect*	80
and DEFINE LINK	77
and DEFINE PROCESSGROUP	77
and Horizon	76
and symbolic addresses	20
compatibility issues	20
DEFINE LINK	19
DEFINE PROCESSGROUP	19
IPv4-mapped addresses	17
message changes	15
MONITOR LINK	15
MONITOR PROCESSGROUP	16
SMTPCADD	64
SMTPSADD	64
TCPOPTS	66
IS LIKE pattern clause	21

J

JOBSTEP parameter	59
journal	
block header	47
expanded entries	48
journal records	
selecting	44

K

keyword conversion	43
--------------------------	----

L

legal notices	2
licensing	12
link	
disabling	69
linking RSQL	13
LOGCTL command	70
login password	
mixed-case	55

M

M204 and Connect* compatibility	12
M204 GEN EXEC	12

M204LINK.....	12
MEMORY and PAGES	
with ALLOCATE.....	68
messages	
for IPv6 addressing.....	15
new and updated.....	82
mixed-case login passwords.....	55
mixed-case passwords	
with LOGCTL.....	70
module size increase.....	13
MONITOR GSTORAGE.....	45, 71
MONITOR LINK format.....	15
MONITOR PROCESSGROUP format.....	16
MONITOR STAT.....	39
MONITOR TASKS.....	38
command.....	72
CPUONZIP.....	38
CPUTOTZE.....	38
MP subtask	
server swapping.....	62, 63
MP/204 licensing.....	12
MQDEQMAN keyword.....	51

O

OPEN	
with long DDNAME.....	53
OPEN LINK	
and REPLACE keyword.....	74
OPEN LINK command.....	74
operating system requirements.....	12
operators	
ANDIF and ORIF short-circuit.....	34
OPSYS parameter.....	59
Ordered Index search.....	21
ORIF.....	34

P

Parameter	
SERVNSA.....	63
SERVNSSZ.....	64
parameters	
ASTRPPG.....	57
CUSTOM.....	57
DSOPT.....	58
JOBSTEP.....	59
OPSYS.....	59
SCHDOFL.....	60
SCHDOFLZ.....	61
SCHDOPT.....	62
SCHDOPT1.....	63
SMTPCADD.....	64
SMTPSADD.....	64
SQLBUFSZ.....	65
STORCURG.....	65
STORMAXG.....	66
STORMXZP.....	66
STORZPAG.....	66

TCPOPTS.....	66
TEMPBOP.....	67
ZPAGEOPT.....	68
ZQMAX.....	68
passwords	
mixed-case with LOGCTL.....	70
pattern optimization.....	21
performance enhancements	
coded and binary keywords.....	43
POINT\$ condition in FIND.....	27
procedure pages	
above the bar.....	41

R

reliability enhancements.....	47
RESTART	
IGNORE option.....	47
RESTART command.....	74
RSQL.....	13

S

SCHDOFL	
parameter.....	60
SCHDOFLZ	
parameter.....	61
SCHDOPT parameter.....	62
SCHDOPT1 parameter.....	63
search pattern optimization.....	21
security enhancements.....	55
SELECT statement.....	49
server size requirement.....	52
server swapping.....	62, 63
SERVNSA.....	41
Parameter.....	63
SERVNSSZ.....	41
Parameter.....	64
SERVSIZE.....	52
SFGE\$ condition in FIND.....	28
SFL\$ condition in FIND.....	28
SMTPCADD parameter.....	64
SMTPSADD parameter.....	64
\$SNDMAIL function.....	14
\$SOCKET function.....	15
sockets	
and Horizon.....	76
\$SOCKETX function.....	17
software delivery.....	12
SQL Catalog	
BLOB/CLOB data types.....	50
SQL Server	
BLOB support.....	48
catalog changes.....	50
CLOB support.....	48
SQL statements	
LOB fields in.....	48
SQLBUFSZ	
BLOBs and CLOBs.....	50

parameter	65
STORCURG parameter	65
STORE RECORD	
THEN CONTINUE	31
STORE RECORD limitation	54
STORMAXG parameter	66
STORMXZP	
parameter	66
STORZPAG	
parameter	66
sub-transaction	67
subroutine processing	
optimization of	51
symbolic addresses	
and Horizon IPv6 addressing	54
and IPv6 addressing	20

T

TCP/IP	
IPv6 addressing	76
TCPOPTS parameter	66
technical support	3
TEMPBOP parameter	67
THEN CONTINUE	31
trademarks	2
transaction backout	67

U

UPDATE statement	49
USE PROC command	74
UTILJ utility	44

V

virtual pages	40
Visual Studio	
and Connect* for .NET Framework	79

X

XDM and ALETS	35
---------------------	----

Z

z/OS	
1 MB virtual pages	40
ALLOCATE EATTR parameter	68
EAV	36
Horizon and IPv6	76
zIIP processing	36
zIIP	
server swapping	62
zIIP processing	36
AMPSUBS	36

AMPSUBZ	36, 37
compatible functions	39
eligible components	37
IIPHONORPRIORITY	39
initialization	37
limits	38
MONITOR TASKS	38
NMPSUBZ	37
parameters	37
SCHDOPT	37
SCHDOPT1	38
statistics	38, 39
ZQMAX	38
ZPAGEOPT parameter	68
ZQMAX parameter	68