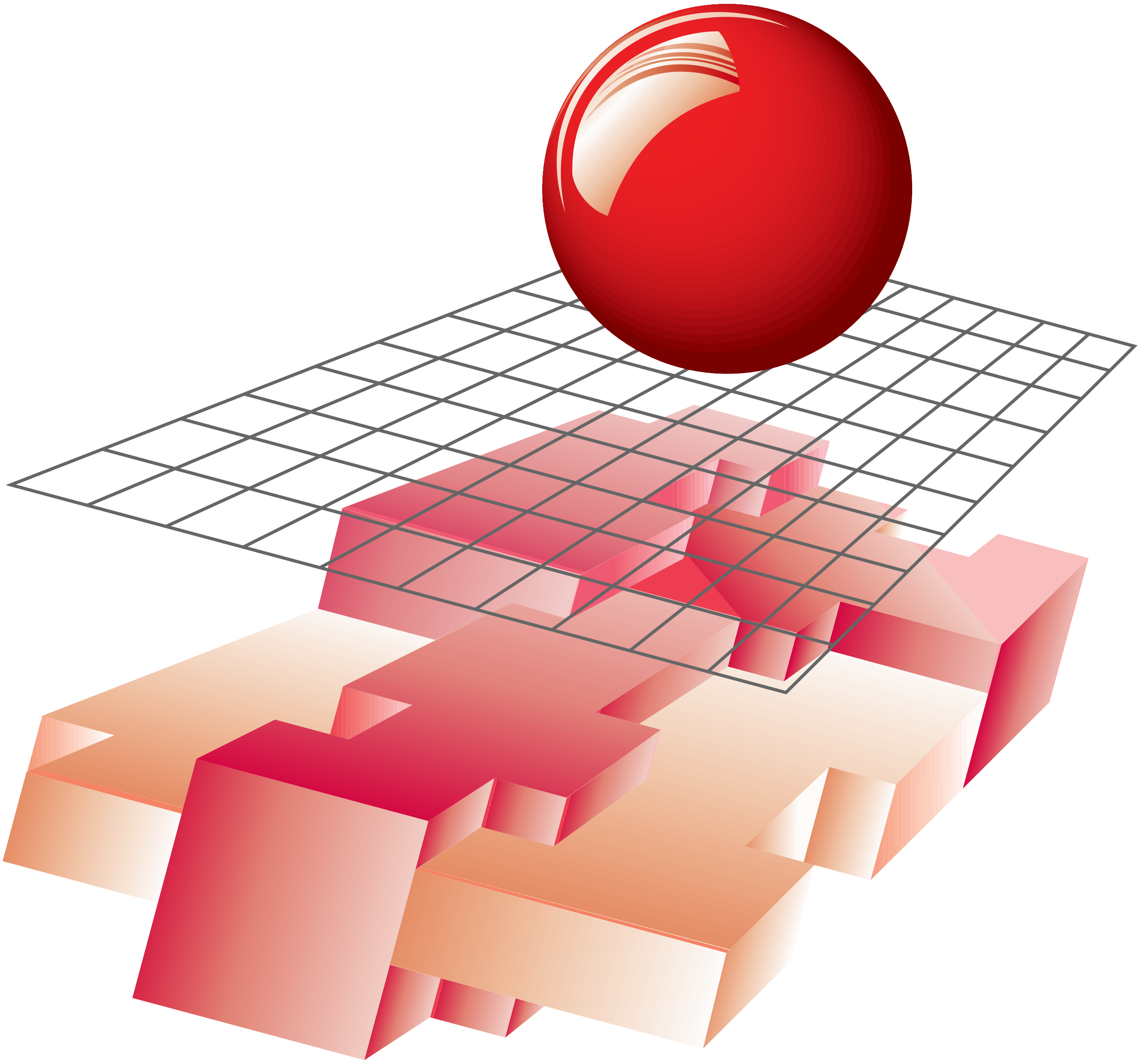
COMPAREX

HCCFA User Manual

Version 4.0 and 5.0



COMPAREX

COMPAREX
Informationssysteme GmbH
Gottlieb-Daimler-Straße 10
D-68165 Mannheim

Internet: http://www.comparex.de

Customers from other countries are kindly
requested to contact directly their local COMPAREX
office:

CXAE6211/04/00

HCCFA 4.0 and 5.0 Second Edition (March 2000)

Information contained in this manual is subject to change without prior notice. The reader is responsible for maintaining up-to-date editions.

Trademarks

Revision No.	Date	Description	Affected Pages
0	March 1999	Original edition.	–
1	May 1999	Changes with support of C2000-xx5/-xx6 (not -A25) models.	II, III, 1-1
2	June 1999	• Changes with support of M3000. • Corrections to LPAR frame images.	II, III, SAFETY-1, 1-1, 1-2, 2-2 thru 2-5
3	February 2000	Second edition with HCCFA release 5.0. • Addition of CLA482I and CLA483W messages relating to Dynamic IP reconfiguration support. • Modification of CLA332I and CLA333I messages relating to IC Channel support.oModification of CLA800E message relating to RAS enhancement. • Addition of explanation relating to the system area allocation of IC. • Changes to explanation relating to MLPF 3.5.2 support. • Minor corrections.	All

PREFACE

This document serves as a user manual relating to Highspeed Coupling Control Feature Assist (HCCFA) releases 4.0 and 5.0 and the HCCFA-based facility called Highspeed Coupling Control Feature (HCCF).

Information contained in this document is subject to change without notice.

Reference manuals

- COMPAREX M2000 Processor Group Operating Procedures
- COMPAREX M3000 Processor Group Operating Procedures
- COMPAREX C2000-xxA/xxB Processor Group Operating Manual
- COMPAREX C2000-xxC/xxD Processor Group Operating Manual
- COMPAREX MLPF 3.5.0 User Manual
- COMPAREX MLPF 3.5.1 and 3.5.2 User Manual
- IBM Enterprise Systems Architecture/390 Principles of Operation (SA22-7201)

Important Note: *Following information is valid throughout the complete manual:*

- *C2000-xx4 represents the processor models C2000-xx3, C2000-xx4, and C2000-A25*
- *C2000-xx6 represents the processor models C2000-xx5, and C2000-xx6 (but not C2000-A25)*
- *C2000-xxA represents the processor models C2000-xx7, C2000-xx8, C2000-xxA, and C2000-xxB*
- *C2000-xxC represents the processor models C2000-xxC and C2000-xxD*

If necessary, detailed models types are listed.

Table of Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

ADMF	Asynchronous Data Mover Facility
BCP	Background Command Processing (routine)
CC	condition code
CCW	channel command word
CDU	Coolant Distribution Unit
CE	customer engineer
CF	coupling facility
CFR	Coupling Facility Receiver (channel)
CFRM	coupling facility resource management
CFS	Coupling Facility Sender (channel)
CHP	channel path
CHPID	channel path ID
CPC	central processor complex
CPU	Central Processing Unit
CPUID	Central Processing Unit identification
CRW	channel report word
ED	external damage
EMIF	ESCON Multiple Image Facility
ES	expanded storage
ESA	Enterprise Systems Architecture
ESCON	Enterprise Systems Connection
ESW	extended status word
HCCF	Highspeed Coupling Control Feature
HCCFA	Highspeed Coupling Control Feature Assist
HSW	HVA status word
HVA	Hypervisor Assist
IBM	International Business Machines Corporation
ICMF	Integrated Coupling Migration Facility
IC	instruction counter
ID	identifier; identification
I/O	input/output (operation or device)
IOCDS	I/O Configuration Data Set
IOCP	Input/Output Configuration Program
IP	Instruction Processor
IPL	initial program loading
ISCH	Inter-System Coupling Channel
KB	kilobyte(s)
LOGDSP	Log Out Display (frame)
LPAR	logical partition; logically-partitioned (mode)
LPRCH	LPAR Channel Definition (frame)
LPRCTL	LPAR Control Definition (frame)
LPRDEF	LPAR Config Definition (frame)
LPRSSR	LPAR System Status Recording (frame)
MB	megabyte(s)
MCIC	machine check interruption code
MCP	Message Command Processing (routine)
MLPF	Multiple Logical Processor Feature
MS	main storage
MVS	Multiple Virtual Storage
OEM	original equipment manufacturer; original equipment manufacturing
OS	operating system(s)

PSW	program status word
RC	Region Code; error reference code
RMF	Resource Measurement Facility
SCDS	System Configuration Data Set
SCP	system control program
SCSW	subchannel status word
SLPMSG	LPAR Operator Message (frame)
SOP	Suboperation Code
SP	System Product
SVC	supervisor-call
SVP	Service Processor
sysplex	system complex
TOD	time-of-day

Note: Common abbreviations, macro names, module names and mnemonics in frames and messages are excluded.

⚠ Safety Summary

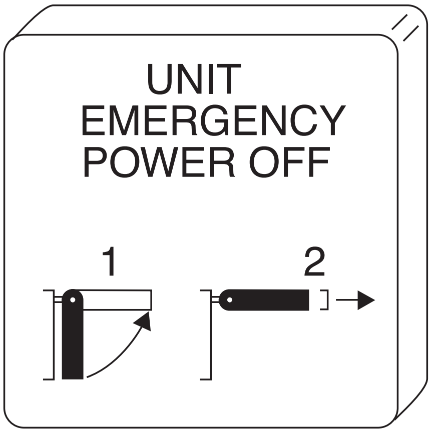
1. ⚠ General Safety Guidelines

Before operating the machine, read the following instructions carefully:

- Follow all the operating procedures provided in this manual.
- Pay special attention to and follow all the hazard warnings on the machine and in the manual. Failure to do so can cause injury to yourself or damage to the machine.
- The hazard warnings which appear on the warning labels on the machine or in the manual have one of the alert headings consisting of an alert symbol and a signal word, DANGER, WARNING or CAUTION as tabulated below. The signal word “NOTICE” is used to present warnings which are not directly related to personal injury hazards.
- If any physical accident such as abnormal noise, smell, smoke or falling down occurs on the processor complex while running, immediately power off the processor complex by pulling one of the UNIT EMERGENCY POWER OFF switches on the following units:
  - M3000: Processor Unit and each Power Distribution Unit
  - M2000: Processor Unit and Coolant Distribution Unit
  - C2000 and CF2000: Processor Unit
    See 2. below for details.
- Clearly identify each destination equipment of primary power sources with proper indication, e.g., a label on the switch on the power distribution panel or board.
- Do not perform any operation or action in any way other than as provided in this manual. When in doubt, call the designated field engineer.
- Keep in mind that the hazard warnings in this manual or on the machine cannot cover every possible case, as it is impossible to predict and evaluate all circumstances beforehand. Be alert and use your common sense.

⚠ **DANGER**	indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠ **WARNING**	indicates a potentially hazardous situation which, if not avoided, can result in death or serious injury.
⚠ **CAUTION**	indicates a hazardous situation which, if not avoided, will or can result in minor or moderate injury, or serious damage of product.
⚠	The alert symbol shown left precedes every signal word for hazard warnings, and appears in safety related descriptions in the manual.

2. ⚠ UNIT EMERGENCY POWER OFF Switches



- For the purpose of powering off in an emergency, UNIT EMERGENCY POWER OFF switches (illustrated right) are provided on the following units:
  - M3000: Processor Unit and each Power Distribution Unit
  - M2000: Processor Unit and Coolant Distribution Unit
  - C2000 and CF2000: Processor Unit
- In using a UNIT EMERGENCY POWER OFF switch, first pull it up and then pull it toward you as illustrated.
- Pulling one of the UNIT EMERGENCY POWER OFF switches instantly shuts down the processor complex other than the Service Processor (for the M2000) and the Console Devices (for the M3000, M2000, and the C2000, and the CF2000), ignoring the system's power off sequence. Jobs in process are aborted and their integrity after recovery is not guaranteed. Hence, this method should be used only in an emergency.
- None of the UNIT EMERGENCY POWER OFF switches can turn off any I/O device.
- When pulled, a UNIT EMERGENCY POWER OFF switch locks itself to prevent further powering on and requires trained and qualified personnel for recovery. Contact the designated field engineer at once.

3. Arrangement of Operation Environment

Here are some advices to make the operation comfortable.

3.1 Room Environment for Operation

- High temperature, high humidity, or dust may cause damage to the machine as well as the operator's health. To prevent such damage, keep your operating place away from sunlight or heater, and away from high humidity and dust.

- Of each Console Device, the main processor and the color monitor have air vents to release the generated heat. Therefore, blocking these vents will cause overheat. Keep enough space, i.e., 50 mm (2 inches) or more, for the airway.

3.2 Workspace for Operation

3.2.1 Desk and Chair

- Operating in an uncomfortable position will tire the operator. In order to operate in a relaxed posture, adjust the heights of desk and chair so that the operator's forearms and thighs are parallel to the floor.

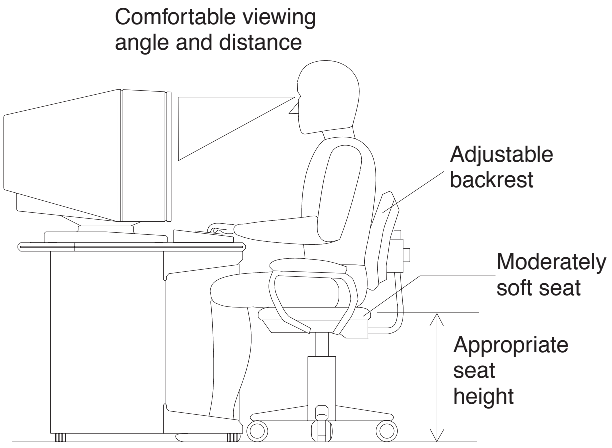
- Selecting a good chair is important to keep the operator in a comfortable position. A moderately soft seat and a backrest fit to the operator's lower back are recommended.

3.2.2 Keyboard

Adjust the keyboard feet, if necessary, to obtain the most comfortable keyboard angle. Adjust it so that soft typing is possible.

3.2.3 Display (Color Monitor)

- Place the display at such a height (on a monitor stand if appropriate) that the operator can view the display screen in parallel or slightly downward, and at such a distance that the operator can maintain a comfortable viewing distance of 510 to 610 mm (20 to 24 inches).
- The operator can change the angle and the direction of the display by tilting and swivelling it. Adjust it so that the operator can view the screen without twisting his or her body.
- To prevent glare and reflection, adjust the display angle, or block the sunlight with curtains or blinds. An antiglare filter might be useful, but its effect on the clarity of the displayed image should be considered.
- The operator can adjust the brightness and contrast of the displayed image using the buttons provided in the lower front of the display (normally covered with the front panel). Push-open the panel, and press the + or – (or ↑ or ↓) button as required for brightness (☼) or contrast (◑) until a proper display image is obtained. Generally, a slightly dark brightness is better for the operator's eyes.
- To keep the clarity of the screen, clean the screen surface with soft cloth periodically.



1 GENERAL INFORMATION

1.1 About This Document

This document describes Hitachi's Highspeed Coupling Control Feature Assist (HCCFA) release 4.0 and 5.0, and the HCCFA-based facility called Highspeed Coupling Control Feature (HCCF).

To meet several purposes, this document is organized in three parts as shown below.

- Part 1 consists of chapter 1, which introduces the purpose and the application of this document, provides an overview of features supported in each release level of HCCFA, and summarizes model-dependent deviations in HCCFA and HCCF functions.
- Part 2 serves as an HCCF user's guide, and consists of three chapters:
  - Chapter 2 introduces an overview of HCCF including HCCFA, explains how to create Parallel Sysplex environments with HCCF, and shows how to perform HCCFA/HCCF functions by means of HCCF operator commands.
  - Chapter 3 explains the HCCF operator commands in detail.
  - Chapter 4 explains messages issued by HCCFA.
- Part 3 serves as an HCCF system administrator's reference, and consists of two chapters:
  - Chapter 5 provides the estimation formulas for HCCF-LPAR storage allocation.
  - Chapter 6 provides HCCF levels and coupling facility limits supported by each HCCF level.

1.2 Applicable Processor Models

The contents of this document apply to the processor types and models equipped with HCCFA 4.0 and 5.0 as shown in Table 1-1. Unless otherwise specifically mentioned, description represented by processor group applies to all processor types and models.

Table 1-1 Application of HCCFA 4.0 and 5.0 by Processor Type and Model

HCCFA	**MLPF**	**Processor Type and Model**												
Release	**Release**	**M3000**		**M2000**			**C2000**					**CF2000**		
Level	**Level**	**A**	**B**	**xx5**	**xx3**	**xx8**	**xx4**	**xx5**	**xx5**	**xx5**	**xx5**	**xx3/xx4**	**xx5**	**xx6**
4.0	3.5.0	–	–	√	√	√	–	–	√	√	–	–	–	–
	3.5.1	√	√	–	–	–	–	–	–	√	√	–	–	–
	3.5.2	–	–	–	–	–	–	–	–	√	√	–	–	–
5.0	3.5.1	–	–	–	–	–	–	–	–	√	√	–	–	–
	3.5.2	√	√	–	–	–	–	–	–	√	√	–	–	–

1.3 Feature Introduced with HCCFA 4.0 and 5.0

HCCF level 4 support is introduced with HCCFA 4.0, and HCCF level 5 support is introduced with HCCFA 5.0.

1.4 Summary of Deviations by Processor Models

Deviations by the processor types and models exist in each part of this document as shown below. Throughout this document, references to ICMF are not applicable to the CF2000-0xA models.

1.4.1 Deviations in HCCF User's Guide

Table 1-2 provides the deviations in HCCF user's guide.

Table 1-2 Deviations in HCCF User's Guide

Chapter	**Section or**	**Deviation by Processor Models**	
	Subsection	**Subject**	**Description**
2	2.1, 2.2	References to single-system coupling, ICMF, and logically-partitioned HCCF	Not applicable to CF2000-0xA Models.
	2.3.2	ISCH environment with logically-partitioned HCCF	Not applicable to CF2000-0xA Models.
		TM field on LPRCTL frame	Not applicable to M3000.
		LPRDEF and LPRCTL frame examples	Applicable to M2000. See appropriate MLPF manual for M3000 and C2000 counterparts.
	2.3.3	ICMF environment	Not applicable to CF2000-0xA Models.
		LPRCTL frame example	Applicable to M2000. See appropriate MLPF manual for M3000 and C2000 counterparts.
3	3.7, 3.8	Reference to ICMF	Not applicable to CF2000-0xA Models.
4	4.2	CLA210I message	Reference to ICMF in message text is not applicable to CF2000-0xA Models.
		CLA332I and CLA333I messages	Reference to ICMF in message text and its explanation is not applicable to CF2000-0xA Models.
		CLA419W message	This message is for ICMF environment and not applicable to CF2000-0xA Models.

1.4.2 Deviations in HCCF System Administrator's Reference

No deviation by processor models exists in HCCF system administrator's reference.

1.4.3 (Intentionally Left Blank)

2 HCCF BASICS

2.1 What Is HCCF?

Highspeed Coupling Control Feature (HCCF) is a hardware feature that provides Parallel Sysplex coupling environments.

HCCF consists of the following components:

- Highspeed Coupling Control Feature Assist (HCCFA): Control code (macrocode) serving as the kernel of HCCF. HCCFA is integrated in the physical processor, and is loaded and activated automatically at an activation of the predefined HCCF-LPAR. Utilizing the main storage allocated to each LPAR, HCCFA in release 4.0 or above realizes shared storage for Parallel Sysplex in the single-system coupling environment and in the multisystem coupling environment.
- HCCF-supporting Multiple Logical Processor Feature (MLPF): Applicable release level of MLPF providing functional support for HCCFA such as simulation of messaging facilities, automatic HCCFA loading and automatic HCCFA logout.
- HCCF-supporting microcode: Applicable release level of microcode providing file access capability required by HCCFA.
- Inter-System Coupling Channel (ISCH): Messaging facility hardware optionally available for creation of multisystem coupling environment with HCCFA 2.0 or above. ISCH is called Coupling Facility Sender (CFS) channel in CPC side, and Coupling Facility Receiver (CFR) channel in HCCF side.
- Standalone HCCF: Coupling facility server hardware (CF2000-0xA Models) optionally available for creation of performance-oriented or large-cluster-oriented multisystem coupling environment.

HCCF with HCCFA in release 4.0 and 5.0 provides multisystem coupling environment called ISCH environment and single-system coupling environment called ICMF environment.

- ISCH environment: Multisystem Parallel Sysplex created by coupling multiple processors via CFS and CFR channels. Two types of ISCH environments are available: one with the standalone HCCF suitable to performance-oriented or large-cluster-oriented Parallel Sysplex, and the other with the logically-partitioned HCCF suitable to flexibility-oriented or small-cluster-oriented Parallel Sysplex. Figures 2-1 and 2-2 illustrate typical configurations of these ISCH environments.
- ICMF environment: Single-system Parallel Sysplex created by the standard function called Integrated Coupling Migration Facility (ICMF) which, as illustrated in Figure 2-3, couples multiple Logical Partitions (LPARs) with each other via simulated messaging facilities, i.e., simulated CFS channel and simulated CFR channel. This ICMF environment can share the logically-partitioned HCCF in the ISCH environment mentioned above.

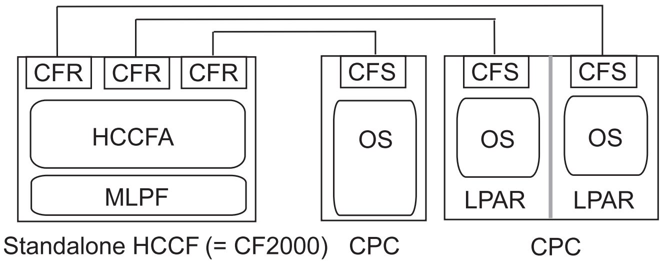


Figure 2-1 ISCH Environment with Standalone HCCF

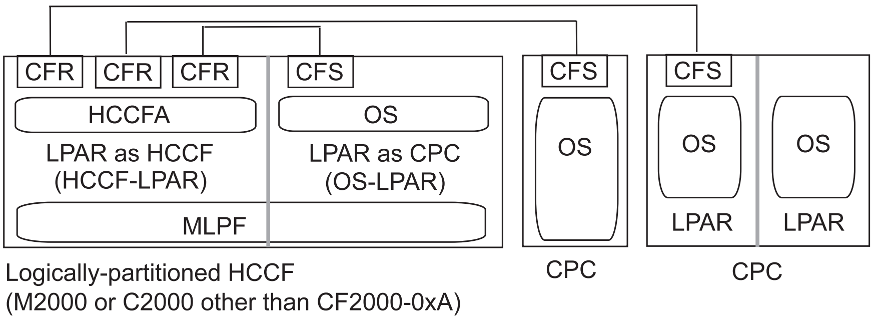


Figure 2-2 ISCH Environment with Logically-Partitioned HCCF

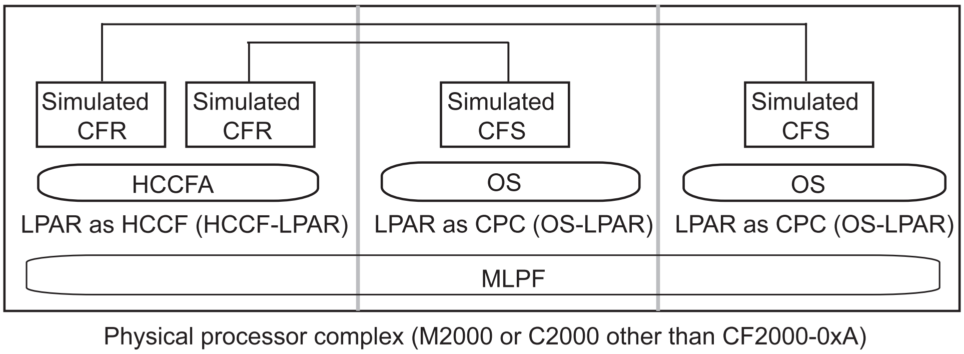


Figure 2-3 ICMF Environment

2.2 Configuration Requirements

ISCH environment with standalone HCCF

- One CF2000 for standalone HCCF with
  - Necessary amount of ISCHs all predefined as CFR channels
  - Ordinary configuration definitions on LPAR frames
- Necessary amount of CPCs with ISCHs (or equivalents) defined as CFS channels by IOCDS
- Applicable cables connecting CFR and CFS channels with each other type

ISCH environment with logically-partitioned HCCF

- One M3000, M2000, or C2000 for logically-partitioned HCCF with
  - Necessary amount of ISCHs
  - Definition of CFR and CFS channels by IOCDS
  - Ordinary configuration definitions on LPAR frames
  - Additional definitions on LPAR frames
    - Defining of at least one LPAR as an HCCF-LPAR dedicated to HCCFA
    - Defining of ISCHs assigned to the HCCF-LPAR(s) as CFR channels
    - Defining of ISCHs assigned to desired LPARs among those for use by the operating system (OS-LPARs) as CFS channels
- Necessary amount of CPCs with ISCHs (or equivalents) defined as CFS channels by IOCDS
- Applicable cables connecting CFR and CFS channels with each other type

ICMF environment

- One M3000, M2000 or C2000 other than CF2000-0xA
- Ordinary configuration definitions on LPAR frames
- Definition on a specific LPAR frame to define at least one LPAR (and up to two LPARs) as (an) HCCF-LPAR(s) dedicated to HCCFA
- Definition on a specific LPAR frame to enable ICMF onto the HCCF-LPAR(s) and onto desired LPARs among those for use by the operating system (OS-LPARs)

2.3 How To Create Parallel Sysplex Environment

2.3.1 Creation of ISCH Environment with Standalone HCCF

Creation of an ISCH environment with the standalone HCCF consists in the ordinary hardware installation process as follows:

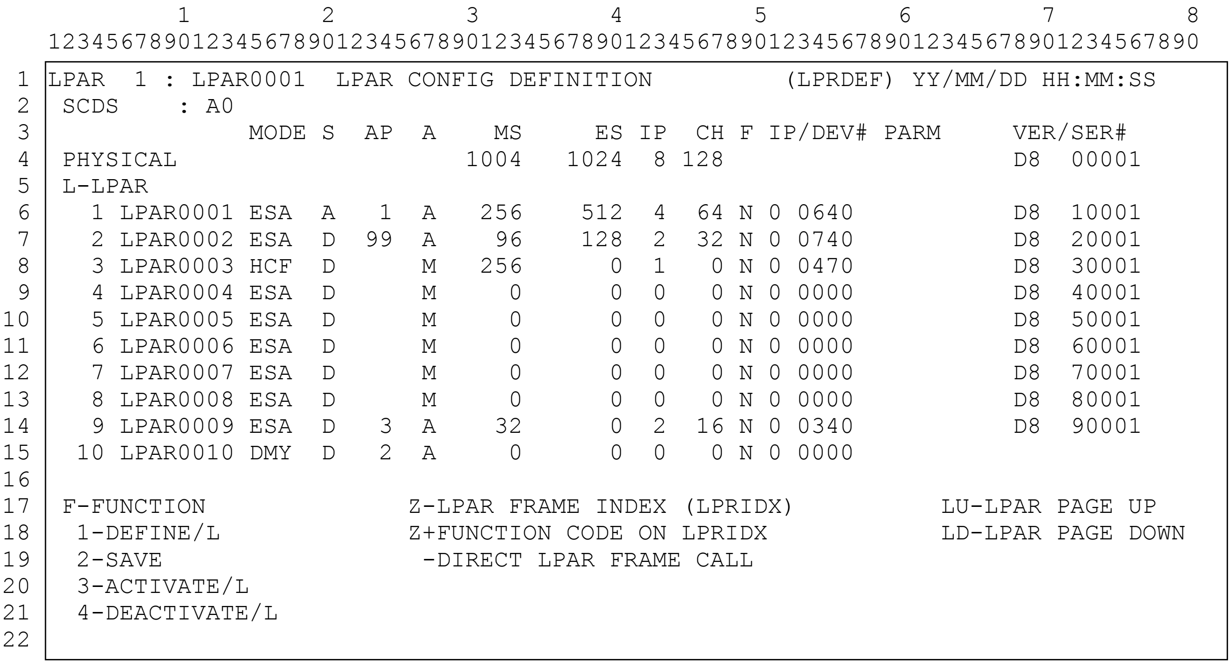
1. Install or procure one standalone HCCF (i.e., CF2000-0xA) equipped with a necessary amount of ISCHs predefined as CFR channels.
2. Perform ordinary configuration definitions on the CF2000-0xA, using LPAR frames according to applicable MLPF manuals. This step can be skipped if the CF2000-0xA has been a procured one with such definitions already done.
3. Procure necessary amount of CPCs equipped with ISCHs (or equivalents) defined as CFS channels. If ISCHs (or equivalents) are not defined, define them as CFS channels by IOCDS.
4. Connect CFR channels on the CF2000-0xA and CFS channels on the CPCs to each other type with applicable cables.

2.3.2 Creation of ISCH Environment with Logically-Partitioned HCCF

Creation of an ISCH environment with the logically-partitioned HCCF consists in the hardware installation process and the definition operations on specific LPAR frames as shown below. For details of LPAR frames, refer to an applicable MLPF manual.

1. Install one M3000, M2000, or C2000 other than CF2000-0xA Models, or procure already installed one, equipped with a necessary amount of ISCHs.
2. Define the ISCHs as CFR and CFS channels by IOCDS.
3. Perform ordinary configuration definitions on the M3000, M2000, or C2000, using LPAR frames. This step can be skipped if the M3000, M2000, or C2000 has been a procured one with such definitions already done.
4. Procure necessary amount of CPCs equipped with ISCHs (or equivalents) defined as CFS channels. If ISCHs (or equivalents) are not defined, define them as CFS channels by IOCDS.
5. Connect CFR channels and CFS channels, including those on the CPCs, to each other type with applicable cables.
6. In the HCCF side, call the LPRDEF frame (Figure 2-4) and redefine at least one LPAR into an HCCF-LPAR by specifying the following for the target LPAR. (Underlined portions in the figure denote input fields for definition parameters.) The other settings on this frame can remain unchanged.
   - If the MS size (MS field setting) of the target LPAR is less than 8 MB, enter a decimal number for an MS size of no smaller than 8 MB in the MS field.
   - Alter the operating mode (MODE field setting) of the target LPAR from "ESA" to "HCF" by entering HCF in the MODE field.
7. In the HCCF side, call the LPRCTL frame (Figure 2-5) and ensure that the settings of the following fields are all "N" (disabled). If not, enter N in the target field. The other settings on this frame can remain unchanged.
   - MF field for each HCCF-LPAR
   - MF field for each OS-LPAR to be coupled with an HCCF-LPAR
   - TM field for each HCCF-LPAR (not applicable to M3000)
   - TW field for each HCCF-LPAR

In an ISCH environment, no channel type other than CFR can be defined to an HCCF-LPAR on the LPRCH frame. If this restriction is not allowable in the user's configuration, choose and redefine a different LPAR as an HCCF-LPAR.

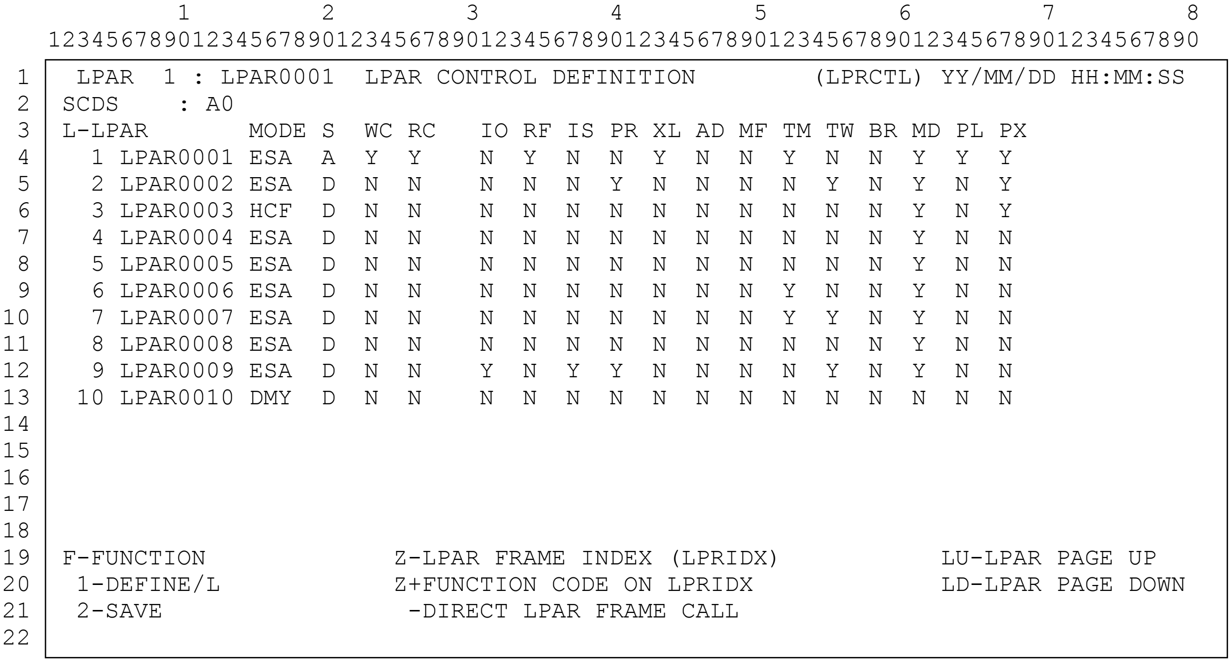


```
LPAR  1 : LPAR0001  LPAR CONFIG DEFINITION         (LPRDEF) YY/MM/DD HH:MM:SS
 SCDS    : A0
              MODE S  AP  A    MS     ES IP  CH F IP/DEV# PARM     VER/SER#
 PHYSICAL                    1004   1024  8 128                    D8  00001
 L-LPAR
   1 LPAR0001 ESA  A   1  A   256    512  4  64 N 0 0640           D8  10001
   2 LPAR0002 ESA  D  99  A    96    128  2  32 N 0 0740           D8  20001
   3 LPAR0003 HCF  D      M   256      0  1   0 N 0 0470           D8  30001
   4 LPAR0004 ESA  D      M     0      0  0   0 N 0 0000           D8  40001
   5 LPAR0005 ESA  D      M     0      0  0   0 N 0 0000           D8  50001
   6 LPAR0006 ESA  D      M     0      0  0   0 N 0 0000           D8  60001
   7 LPAR0007 ESA  D      M     0      0  0   0 N 0 0000           D8  70001
   8 LPAR0008 ESA  D      M     0      0  0   0 N 0 0000           D8  80001
   9 LPAR0009 ESA  D   3  A    32      0  2  16 N 0 0340           D8  90001
  10 LPAR0010 DMY  D   2  A     0      0  0   0 N 0 0000

 F-FUNCTION              Z-LPAR FRAME INDEX (LPRIDX)          LU-LPAR PAGE UP
  1-DEFINE/L             Z+FUNCTION CODE ON LPRIDX            LD-LPAR PAGE DOWN
  2-SAVE                  -DIRECT LPAR FRAME CALL
  3-ACTIVATE/L
  4-DEACTIVATE/L

```

Figure 2-4 LPRDEF Frame for Logically-Partitioned HCCF (Example of M2000)



```
  LPAR  1 : LPAR0001  LPAR CONTROL DEFINITION        (LPRCTL) YY/MM/DD HH:MM:SS
 SCDS    : A0
 L-LPAR       MODE S  WC RC    IO RF IS PR XL AD MF TM TW BR MD PL PX
   1 LPAR0001 ESA  A  Y  Y     N  Y  N  N  Y  N  N  Y  N  N  Y  Y  Y
   2 LPAR0002 ESA  D  N  N     N  N  N  Y  N  N  N  N  Y  N  Y  N  Y
   3 LPAR0003 HCF  D  N  N     N  N  N  N  N  N  N  N  N  N  Y  N  Y
   4 LPAR0004 ESA  D  N  N     N  N  N  N  N  N  N  N  N  N  Y  N  N
   5 LPAR0005 ESA  D  N  N     N  N  N  N  N  N  N  N  N  N  Y  N  N
   6 LPAR0006 ESA  D  N  N     N  N  N  N  N  N  N  Y  N  N  Y  N  N
   7 LPAR0007 ESA  D  N  N     N  N  N  N  N  N  N  Y  Y  N  Y  N  N
   8 LPAR0008 ESA  D  N  N     N  N  N  N  N  N  N  N  N  N  Y  N  N
   9 LPAR0009 ESA  D  N  N     Y  N  Y  Y  N  N  N  N  Y  N  Y  N  N
  10 LPAR0010 DMY  D  N  N     N  N  N  N  N  N  N  N  N  N  N  N  N





 F-FUNCTION             Z-LPAR FRAME INDEX (LPRIDX)           LU-LPAR PAGE UP
  1-DEFINE/L            Z+FUNCTION CODE ON LPRIDX             LD-LPAR PAGE DOWN
  2-SAVE                 -DIRECT LPAR FRAME CALL

```

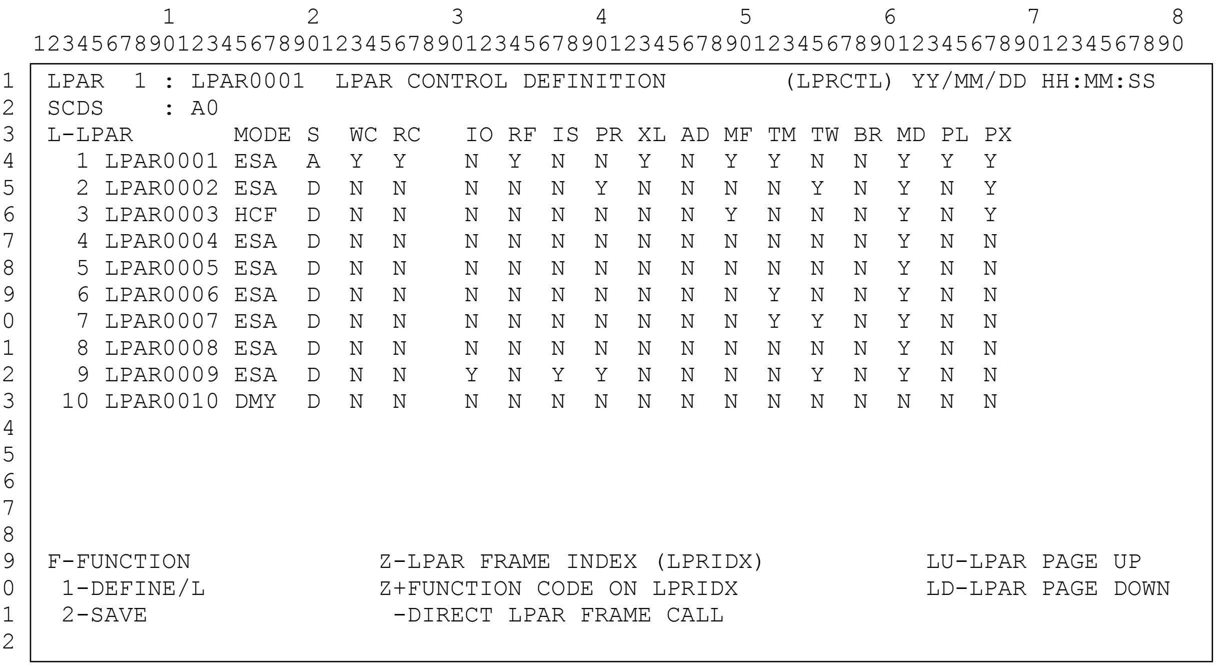
Figure 2-5 LPRCTL Frame for Logically-Partitioned HCCF (Example of M2000)

2.3.3 Creation of ICMF Environment

Creation of an ICMF environment consists in the definition operations on specific LPAR frames as shown below (no hardware installation process). For details of LPAR frames, refer to applicable MLPF manuals.

1. Procure one M3000, M2000, or C2000 other than CF2000-0xA models.
2. Perform ordinary configuration definitions on the M3000, M2000, or C2000, using LPAR frames. This step can be skipped if such definitions are already done.
3. On the LPRDEF frame, redefine at least one LPAR into an HCCF-LPAR in the same manner as in creating an ISCH environment with the logically-partitioned HCCF (section 2.3.2). The only difference is that the maximum number of HCCF-LPARs for the ICMF environment is limited to two.
4. On the LPRCTL frame exemplified in Figure 2-6, enter Y in the MF field for each HCCF-LPAR and each OS-LPAR to be coupled with an HCCF-LPAR. The number of such ICMF-specified OS-LPARs is not limited.
5. On the same frame, ensure that the settings of the TM (not applicable to F-10) and TW fields for each HCCF-LPAR are all "N" (disabled). If not, enter N in the target field. The other settings on this frame can remain unchanged.

In an ICMF environment, all channel definitions on the LPRCH frame are disabled for each HCCF-LPAR and each OS-LPAR to be coupled with an HCCF-LPAR. If this restriction is not allowable in the user's configuration, choose and redefine a different LPAR as an HCCF-LPAR or an OS-LPAR.



```
LPAR  1 : LPAR0001  LPAR CONTROL DEFINITION        (LPRCTL) YY/MM/DD HH:MM:SS
SCDS    : A0
L-LPAR       MODE S  WC RC   IO RF IS PR XL AD MF TM TW BR MD PL PX
  1 LPAR0001 ESA  A  Y  Y    N  Y  N  N  Y  N  Y  Y  N  N  Y  Y  Y
  2 LPAR0002 ESA  D  N  N    N  N  N  Y  N  N  N  N  Y  N  Y  N  Y
  3 LPAR0003 HCF  D  N  N    N  N  N  N  N  N  Y  N  N  N  Y  N  Y
  4 LPAR0004 ESA  D  N  N    N  N  N  N  N  N  N  N  N  N  Y  N  N
  5 LPAR0005 ESA  D  N  N    N  N  N  N  N  N  N  N  N  N  Y  N  N
  6 LPAR0006 ESA  D  N  N    N  N  N  N  N  N  N  Y  N  N  Y  N  N
  7 LPAR0007 ESA  D  N  N    N  N  N  N  N  N  N  Y  Y  N  Y  N  N
  8 LPAR0008 ESA  D  N  N    N  N  N  N  N  N  N  N  N  N  Y  N  N
  9 LPAR0009 ESA  D  N  N    Y  N  Y  Y  N  N  N  N  Y  N  Y  N  N
 10 LPAR0010 DMY  D  N  N    N  N  N  N  N  N  N  N  N  N  N  N  N





F-FUNCTION            Z-LPAR FRAME INDEX (LPRIDX)          LU-LPAR PAGE UP
 1-DEFINE/L           Z+FUNCTION CODE ON LPRIDX            LD-LPAR PAGE DOWN
 2-SAVE                -DIRECT LPAR FRAME CALL

```

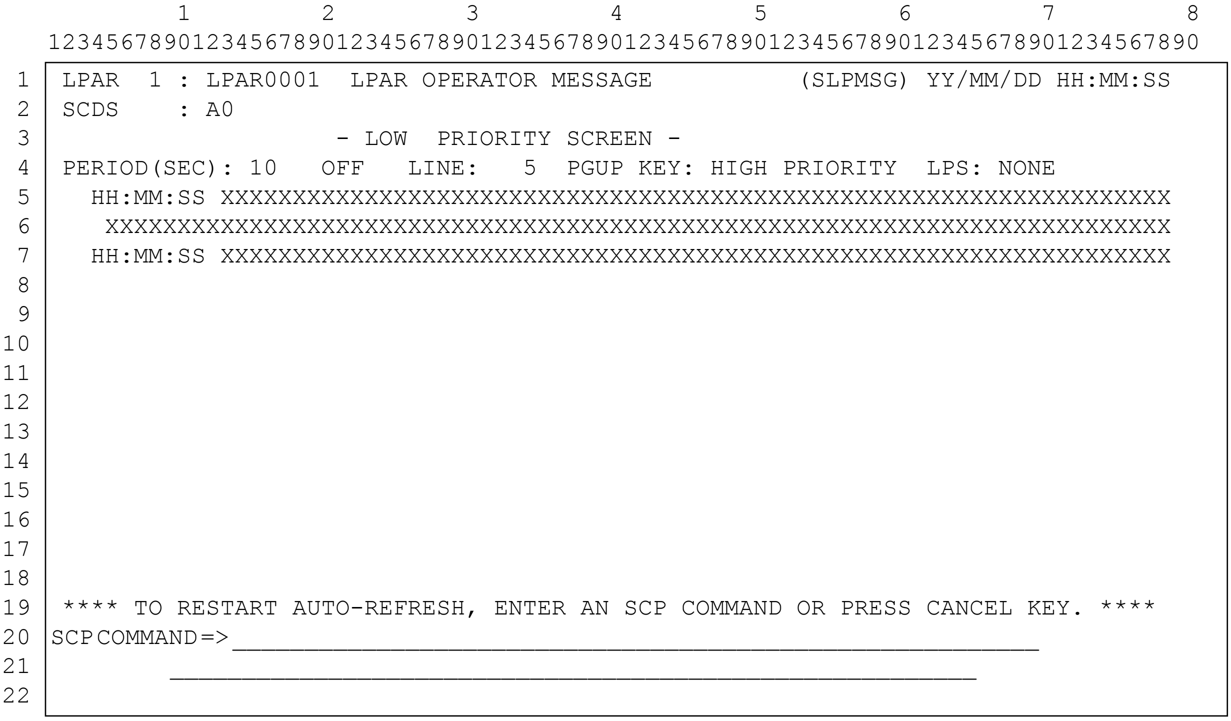
Figure 2-6 LPRCTL Frame for ICMF Environment (Example of M2000)

2.4 Operation Outline of HCCF

The entire operation for HCCF proceeds with the HCCF operator commands entered into the system control program (SCP) command input area in the LPAR Operator Message (SLPMSG) frame illustrated in Figure 2-7. (Underlined portions in the figure denote input fields.) Results of such command execution are displayed as HCCFA messages in the same frame.

Activation of HCCF is performed by activating the predefined HCCF-LPAR. Termination of HCCF is performed by entering the SHUTDOWN command (one of the HCCF operator commands).

For details of the HCCF operator commands, see chapter 3. For details of the HCCFA messages, see chapter 4.



```
 LPAR  1 : LPAR0001  LPAR OPERATOR MESSAGE          (SLPMSG) YY/MM/DD HH:MM:SS
 SCDS    : A0
                   - LOW  PRIORITY SCREEN -
 PERIOD(SEC): 10   OFF   LINE:   5  PGUP KEY: HIGH PRIORITY  LPS: NONE
   HH:MM:SS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   HH:MM:SS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX











 **** TO RESTART AUTO-REFRESH, ENTER AN SCP COMMAND OR PRESS CANCEL KEY. ****
SCP COMMAND =>________________________________________________________
        ________________________________________________________

```

Figure 2-7 Operation of HCCF on SLPMSG Frame

2.5 Saving of HCCF Definition Information

The following information defined with the HCCF operator commands is saved in the HCCF Configuration File at the command execution.

- Time difference defined by the TIMELAG command
- Setting of Dynamic CF Dispatching defined by the DYNACF command

The contents of the HCCF Configuration File are read at the HCCF-LPAR activation and used at the HCCFA activation. The HCCF Configuration File is able to store the definition information for HCCF-LPARs. The saved information remains valid regardless of the system's power-off.

3 HCCF OPERATOR COMMAND DETAILS

This chapter describes HCCF operator commands in alphabetical order of command name.

Symbolic conventions

- $\begin{Bmatrix} \mathtt{X} \\ \mathtt{Y} \end{Bmatrix}$: Indicates either X or Y can be selected. When choice can be omitted (indicated in a bracket) and no selection is made, an underlined choice will be taken as a default.
- `[X]` : Indicates X can be omitted.
- Δ: Indicates that one or more delimiters (spaces or commas) are required.
- `Normal_text:` Indicates that the text must be typed as it is.
- *`Italic_text`*: Indicates that what is meant by the text must be entered as a specific text or value.

Two types of operands

- Fixed-position operand: Parameter operand necessary to be specified at the system-defined operand position.
- Floating-position operand: System-defined keyword operand not necessary to be specified at a fixed operand position. Sequence of specifying such operands is at the user's choice.

3.1 DISPLAY Command

Function

Displays various configuration information relating to HCCF.

Input format

```
                ⎧ ⎧SYS    ⎫                                            ⎫
                ⎪ ⎩SYSTEM ⎭                                            ⎪
                ⎪ IP                                                   ⎪
                ⎪ ⎧STOR   ⎫                                            ⎪
                ⎪ ⎩STORAGE⎭                                            ⎪
                ⎪      ⎡ ⎧ALL  ⎫⎤                                      ⎪
                ⎪ CHP  ⎢,⎨     ⎬⎥                                      ⎪
                ⎪      ⎣ ⎩CHPID⎭⎦                                      ⎪
 ⎧D      ⎫      ⎪                                                      ⎪
Δ⎨       ⎬Δ     ⎨ TIMELAG                                              ⎬
 ⎩DISPLAY⎭      ⎪ SYSMODE                                              ⎪
                ⎪                   ⎧AREA                  ⎫           ⎪
                ⎪                   ⎪     ⎡ ⎧ALL  ⎫⎤       ⎪           ⎪
                ⎪ ⎧STR      ⎫       ⎪     ⎢ ⎪     ⎪⎥       ⎪           ⎪
                ⎪ ⎨         ⎬,      ⎨ SID ⎢,⎨CACHE⎬⎥       ⎬           ⎪
                ⎪ ⎩STRUCTURE⎭       ⎪     ⎣ ⎩LIST ⎭⎦       ⎪           ⎪
                ⎪                   ⎩structure_ID          ⎭           ⎪
                ⎪ UREV                                                 ⎪
                ⎩ DYNACF                                               ⎭
```

Operands

- `{SYS | SYSTEM}` : Information on the HCCF-LPAR where HCCFA is running.
  - IP address and its status – online or offline or, as from HCCFA 2.0, damaged.
  - CHPID and its status – online or offline or, as from HCCFA 2.0, damaged – and, with HCCFA 3.0, the message path status of that CHPID.
  - Main storage capacity.
  - Expanded storage capacity. Available with HCCFA 2.0 or above.
- `IP`: HCCF-LPAR–allocated IP address and its status – online or offline or, as from HCCFA 2.0, damaged.
- `{STOP | STORAGE}` : HCCF-LPAR–allocated capacity of main storage or, as from HCCFA 2.0, expanded storage.
- `CHP[,{ALL | CHPID}]` : HCCF-LPAR–allocated channel path information as specified in the second operand.
  - `ALL` (default when omitted): All the connected CHPIDs and their statuses – online or offline or, as from HCCFA 2.0, damaged – and, with HCCFA 3.0, the message path statuses of those CHPIDs.
  - *`CHPID`*: Status of the specified CHPID (0–FF) – online or offline or, as from HCCFA 2.0, damaged – and, with HCCFA 3.0, the message path status of that CHPID.
- `TIMELAG`: Time difference currently set by the TIMELAG command, and current time.
- `SYSMODE`: Processing mode of HCCFA currently set by the SYSMODE command.

- $\left\{\begin{array}{l}\texttt{STR}\\\texttt{STRUCTURE}\end{array}\right\}\texttt{,AREA}$:

  Structure area information, i.e., allocated area size and used area size.

- $\left\{\begin{array}{l}\texttt{STR}\\\texttt{STRUCTURE}\end{array}\right\}\texttt{,SID}\left[\texttt{,}\left\{\begin{array}{l}\underline{\texttt{ALL}}\\\texttt{CACHE}\\\texttt{LIST}\end{array}\right\}\right]$:

  List of structure IDs as specified in the third operand.

  - `ALL` (default when omitted): Structure IDs of all the created structures.
  - `CACHE`: Structure IDs of cache structures.
  - `LIST`: Structure IDs of list structures.

- $\left\{\begin{array}{l}\texttt{STR}\\\texttt{STRUCTURE}\end{array}\right\}\texttt{,}\textit{structure\_ID}$:

  Information on specified structure ID.

  - Structure ID.
  - Structure type (cache or list).
  - Allocated area size.

- `UREV`: Current unit revision of HCCFA. Available with HCCFA 2.0 or above.
- `DYNACF`: Current setting of Dynamic CF Dispatching. Not available in ICMF environment.

3.2 DYNACF Command

Function

Specifies and immediately saves the setting of Dynamic CF Dispatching (initially in the disabled status). The setting defined by this command remains valid even after HCCFA restart. This command is not available in ICMF environment.

Input format

$\Delta\texttt{DYNACF}\Delta = \left\{\begin{array}{l}\texttt{YES}\\\texttt{NO}\end{array}\right\}$

Operand

$\left\{\begin{array}{l}\texttt{YES}\\\texttt{NO}\end{array}\right\}$: Enables (YES) or disables (NO) Dynamic CF Dispatching.

3.3 LOGOUT Command

Function

Specifies logout of the HCCFA memory contents without disrupting HCCFA. This command is effective only when HCCFA is operating in CE mode.

Input format

$$\Delta\texttt{LOGOUT}\Delta\begin{array}{l}\begin{bmatrix}\texttt{PROG = }\begin{Bmatrix}\underline{\texttt{NO}}\\ \texttt{YES}\end{Bmatrix}\end{bmatrix}\\ \begin{bmatrix}\begin{Bmatrix}\texttt{SID = }\textit{structure\_ID}\\ \texttt{ADDR = }\textit{start\_address-end\_address}\end{Bmatrix}\end{bmatrix}\end{array}$$

Operands

- $\texttt{PROG = }\begin{Bmatrix}\underline{\texttt{NO}}\\ \texttt{YES}\end{Bmatrix}$:

  Selection whether the program area contents are included in the HCCFA logout or not. When omitted, “NO” is assumed as the default.

  - `NO`: The program area contents are excluded from the HCCFA logout.
  - `YES`: The program area contents are included in the HCCFA logout.
- `SID=`*`structure_ID`*: Means to choose a management information area for logout by specifying its structure ID each in hexadecimal representation (0–FFFF). Available with HCCFA 2.0 or above.
- `ADDR=`*`start_address-end_address`*: Means to choose a desired area for logout by specifying its start and end addresses each in hexadecimal representation (0–7FFFFFFF).

3.4 SHUTDOWN Command

Function

Terminates processing of HCCFA. Upon the command entry, the operator is requested to continue or cancel the shutdown processing through an HCCFA message CLA300A. Enter “GO” to terminate and “CANCEL” to cancel. When HCCFA is terminated by this command, the IP stops with the PSW having the format of “000A0000 00000000.”

Input format

```
ΔSHUTDOWN
```

Operand

None.

3.5 SYSMODE Command

Function

Specifies, and can alter, the processing mode of HCCFA-NORMAL (initial mode meant for end user's operation) and CE (meant for the field engineer's maintenance and servicing). The latter mode is prerequisite for use of the LOGOUT command.

Input format

$$\Delta\texttt{SYSMODE}\Delta\texttt{MODE = }\begin{Bmatrix}\texttt{NORMAL}\\ \texttt{CE}\end{Bmatrix}$$

Operand

$\texttt{MODE = }\begin{Bmatrix}\texttt{NORMAL}\\ \texttt{CE}\end{Bmatrix}$: Selection between NORMAL mode and CE mode.

3.6 TIMELAG Command

Function

Specifies, and can restore, time difference from the system-defined standard time.

Input format

$$\Delta\texttt{TIMELAG}\Delta\texttt{ = }\begin{Bmatrix}\texttt{GAIN = }\textit{hh:mm}\texttt{[:ss]}\\ \texttt{LOSE = }\textit{hh:mm}\texttt{[:ss]}\\ \texttt{RESET}\end{Bmatrix}$$

Operands

- `GAIN=`*`hh`*`:`*`mm`*`:[`*`ss`*`]`: Hour and minute [and second] each in two-digit decimal number (00–99) for gain from the standard time.
- `LOSE=`*`hh`*`:`*`mm`*`:[`*`ss`*`]`: Hour and minute [and second] each in two-digit decimal number (00–99) for delay from the standard time.
- `RESET`: Resetting of time difference to the standard time.

3.7 VARYONLINE Command

Function

Executes onlining of hardware resources configured in ISCH environment.

Input format

$$\Delta\begin{Bmatrix}\texttt{VARYON}\\ \texttt{VARYONLINE}\end{Bmatrix}\Delta\begin{Bmatrix}\texttt{IP = }\textit{IP\_No.}\\ \texttt{CHP = }\textit{CHPID}\end{Bmatrix}$$

Operands

- `IP=`*`IP_No.`*: Specifies the IP number to be placed into the online status in hexadecimal representation (0–F).
- `CHP=`*`CHPID`*: Specifies the CHPID to be placed into the online status in hexadecimal representation (0–FF). No CHPID of ICMF can be specified.

3.8 VARYOFFLINE Command

Function

Executes offlining of hardware resources configured in ISCH environment. The offlined resource is disconnected from HCCF-LPAR.

Input format

$$\Delta\begin{Bmatrix}\texttt{VARYOFF}\\\texttt{VARYOFFLINE}\end{Bmatrix}\Delta\begin{Bmatrix}\texttt{IP = }\textit{IP\_No.}\\\texttt{CHP = }\textit{CHPID}\texttt{[,FORCE]}\end{Bmatrix}$$

Operands

- `IP=`*`IP_No.`*: Specifies the IP number to be placed into the offline status in hexadecimal representation (0–F).
- `CHP=`*`CHPID`*`[,FORCE]`: Specifies the CHPID to be placed into the offline status in hexadecimal representation (0–FF). No CHPID of ICMF can be specified. When the message path is established between the specified channel path and the OS-LPAR, the channel path cannot be placed into the offline status. In such a case, take either of the following actions:
  - Cancel the message path from the OS-LPAR, and offline the channel path by the VARYOFFLINE command from HCCFA.
  - Specify FORCE operand of the VARYOFFLINE command which forcibly offlines the channel path even if the message path is established.

Remarks

The IP offlined by the VARYOFFLINE command stops with the PSW of “000A0000 00000200.”

4 HCCFA MESSAGES

4.1 Explanation on Formats

Display format of HCCFA messages

Each HCCFA message is displayed in the “CLAnnnl: text” format as explained below.

- CLA: Identifier of an HCCFA message.
- nnn: Message code represented by a decimal number from 001 to 999.
- l: Level code in one character representing a message level.
  - A: Operator action. The system is requesting an operator action.
  - E: Error. A failure occurred, and the system has failed to recover it and has been affected. The system has continued processing except the failure portion.
  - I: Information. Not an error but for information only.
  - W: Warning. A failure occurred but has been recovered without affecting the system.
- text: Message text that gives detailed information on the message.

Display format of time stamp

Each message display is usually preceded by a time stamp in the “yy/mm/dd hh:mm:ss” format, showing the time of the message output. Such time stamp can be advanced or delayed by entering the HCCF operator command TIMELAG. In the event of an error, some messages may not be preceded by any time stamp.

Explanation format

In section 4.2, each message is explained in the following format (example):

①→	`CLA001W`
②→	`COMMAND NAME INVALID`
③→	The specified command string contains an invalid command name.
④→	S: Ignores the specified command.
⑤→	O: Reenter the command string with a correct command name.

①: Message ID.
②: Message text.
③: Meaning/explanation.
④: System’s action after the message output.
⑤: Operator’s action required.

Symbolic conventions

- *`Italic_text`*: Indicates that what is meant by the text appears as a specific text or value in actual message display.
- `[X]`: Indicates that X may not be displayed.
- $\left\{\begin{matrix}\mathtt{X}\\\mathtt{Y}\end{matrix}\right\}$: Indicates that either X or Y is displayed.

4.2 Contents and Explanation of HCCFA Messages

`CLA001W`

`COMMAND NAME INVALID`

The specified command string contains an invalid command name.

S: Ignores the specified command.

O: Reenter the command string with a correct command name.

`CLA002W`

`OPERAND NOT SPECIFIED`

The specified command string does not contain (a) necessary operand(s).

S: Ignores the specified command.

O: Reenter the command string with (a) necessary operand(s).

`CLA003W`

operand_name `OPERAND NAME INVALID`

The specified command string contains an invalid operand indicated by the *operand_name.*

S: Ignores the specified command.

O: Reenter the command string with a correct operand name.

`CLA004W`

operand_name `OPERAND VALUE INVALID OR MISSING`

The operand value indicated by the *operand_name* is invalid or missing.

S: Ignores the specified command.

O: Reenter the command string with a correct operand value.

`CLA005W`

operand_name `OPERAND DUPLICATE OR EXCLUSIVE`

The operand indicated by the *operand_name* is assigned twice or in conflict with another operand.

S: Ignores the specified command.

O: Reenter the command string with correct operands.

`CLA071W`

`COMMAND NOT ACCEPTED:` *routine_name* `FUNCTION BUSY`

The specified command is not accepted since Control routine of HCCFA indicated by the `routine_name` (FRAME or PATCH) is busy because of the following reasons:

- FRAME: Frame control routine has accepted the maximum of 16 commands to be executed and cannot accept any more command.
- PATCH: Online Patch Control routine is running online patch. Command cannot be accepted until the ongoing online patch is completed.

S: Ignores the specified command.

O: Reenter the same command string after the busy status is over.

`CLA072W`

`LOGOUT NOT ACCEPTED: LOGOUT FUNCTION BUSY`

The specified LOGOUT command is not accepted since Logout Control routine of HCCFA is busy with processing of logout caused earlier. Only one logout is processed at a time.

S: Ignores the specified command.

O: Reenter the same command string after the busy status is over.

`CLA073W`

`command_name` `NOT ACCEPTED: CONFIG FUNCTION BUSY`

The specified command indicated by *`command_name`* is not accepted since dynamic reconfiguration control routine of HCCFA is busy with processing dynamic reconfiguration. Only one VARYONLINE/ VARYOFFLINE command is processed at a time.
S: Ignores the specified command.
O: Reenter the same command string after the busy status is over.

`CLA075W`

`COMMAND NOT ACCEPTED: SHUTDOWN PROCESS RUNNING`

The specified command is not accepted since SHUTDOWN command processing is in progress.
S: Ignores the specified command.
O: Stop using the command.

`CLA081W`

`command_name` `REJECTED:` *`routine_name`* `FUNCTION NOT ACTIVE`

The command indicated by the *`command_name`* is rejected because the HCCFA control routine indicated by the *`routine_name`* (COMMAND or CONFIG) is not active. The command may be rejected right after the HCCFA activation since Control routine of HCCFA is processing initialization.
S: Ignores the specified command.
O: See if CLA780E message for the same routine name has been output and, if so, follow that instruction. If not, reenter the same command string a little later.

`CLA082W`

`command_name` `REJECTED:` *`routine_name`* `FUNCTION DAMAGED`

The command indicated by the *`command_name`* is rejected because the HCCFA control routine indicated by the *`routine_name`* is damaged. For details of routine names, see CLA780E message.
S: Ignores the specified command.
O: Stop using the command.

`CLA083W`

`command_name` `REJECTED: INSUFFICIENT SYSTEM STORAGE`

The command indicated by the *`command_name`* is rejected because HCCFA cannot allocate sufficient work area for the command execution in the system storage.
S: Ignores the specified command.
O: Reenter the same command string when sufficient work area in the system storage becomes available.

`CLA090W`

`command_name` `DATA SAVE FAILED: I/O ERROR OCCURRED`

The data save operation to the HCCF configuration file with the command indicated by the command_name has failed because of an I/O error. The operator command ends normally but the data is stored only in MS. Hence the data is effective only in the current operaiton and lost after HCCFA restart. Details are given by a preceding or succeeding CLA880E message.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

`CLA200I`

`HCCFA ALL RIGHTS RESERVED,`
`COPYRIGHT (C)` *`Year_1`*`,[` *`Year_2`*`,] HITACHI, LTD.`

The copyright notice on HCCFA is displayed.
S: Continues processing.
O: No action.

CLA201I

HCCFA vv-rr

The version number of HCCFA is displayed.
S: Continues processing.
O: No action.

CLA210I

HCCF CONFIGURATION
IP=*IP_No.* [ISCH=I*SCH_No.]* [ICMF=I*CMF_No.]*
MS=*MS_size*MB ES=*ES_size*MB

The configuration information of HCCFA is displayed each in decimal representation.
S: Continues processing.
O: No action.

CLA230W

HCCFA *function_name* FUNCTION IS UNAVAILABLE

The function indicated by the function_name is detected to be unavailable at HCCFA activation. If the function is SAVE, the target information has not been saved in the HCCF configuration file. After this message appears, save operation to the HCCF configuration file is disabled.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

CLA235W

CONFIG FILE READ FAILED: I/O ERROR OCCURRED

Reading the HCCF configuration file has failed at HCCFA activation because of an I/O error. This error causes the following settings invalid:

- Time difference defined by the TIMELAG command
- Setting of Dynamic CF Dispatching defined by the DYNACF command

After this message appears, save operation to the HCCF configuration file is disabled.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

CLA236W

CONFIG FILE WRITE FAILED: I/O ERROR OCCURRED

Writing the HCCF configuration file has failed because of an I/O error. Details are given by a preceding or succeeding CLA880E message. This error causes no effect on HCCFA processing. After this message appears, save operation to the HCCF configuration file is disabled.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

CLA240E

HCCFA START FAILED: INSUFFICIENT SYSTEM STORAGE
MS=*main_storage_size*MB SYSTEM STORAGE=*system_area_size*MB

HCCFA cannot be started because main storage (MS) size is small and cannot reserve sufficient system area to activate HCCFA. HCCFA MS size is indicated by the *main_storage_size* in megabytes in decimal representation, and necessary contiguous system area size to activate HCCFA is indicated by the *system_area_size* in megabytes in decimal representation. This message may be displayed when configuration of HCCFA is either of the following statuses:

- Main storage size is small and expanded storage size is large.
- Main storage size is small and channel path count is large.

S: Aborts the starting of HCCFA and enters into a system wait showing the HCCFA Wait Code ‘0070.’
O: Take any of the following actions and restart HCCFA.

- Increase the size of main storage to be larger than that of system area.
- Reduce the size of expanded storage.
- Reduce the number of channel paths.

CLA241E

HCCFA START FAILED: MS DAMAGED *(lower_address-higher_address)* MS=*main_storage_size*MB SYSTEM STORAGE=*system_area_size*MB

HCCFA cannot be started because the main storage (MS) area indicated by `lower_address-higher_address` is damaged. HCCFA MS size is indicated by the `main_storage_size` in megabytes in decimal representation, and necessary contiguous system area to activate HCCFA is indicated by *system_area_size* in megabytes in decimal representation.
S: Aborts the starting of HCCFA and enters into a system wait by system wait code '0070.'
O: Report the message ID and the message text to the field engineer.

CLA300A

SHUTDOWN COMMAND ENTERED,
CONTINUE WITH SHUTDOWN PROCEDURE ?
PLEASE REPLY 'GO' OR 'CANCEL'

The SHUTDOWN command has been entered and the operator is requested to proceed or cancel.
S: Waits for the operator's key entry.
O: Enter GO to terminate HCCFA, or enter CANCEL to cancel.

CLA301I

SHUTDOWN PROCESS STARTED

Processing of the SHUTDOWN command is activated.
S: Continues processing of the SHUTDOWN command.
O: No action.

CLA302I

SHUTDOWN PROCESS CANCELLED

Processing of the SHUTDOWN command is cancelled.
S: Cancels processing of the SHUTDOWN command.
O: No action.

CLA303A

REPLY INVALID, ENTER 'GO' OR 'CANCEL'

The operator's response to the CLA300A message has been invalid and the operator is requested to reenter a correct response.
S: Waits for the operator's key entry.
O: Enter GO to terminate HCCFA, or enter CANCEL to cancel.

CLA304I

SHUTDOWN PROCESS COMPLETED

Processing of the SHUTDOWN command is completed.
S: Terminates HCCFA.
O: No action.

CLA310I

TIME DELTA IS $\left[\begin{Bmatrix}+\\-\end{Bmatrix}\right]$ *h1h1:m1m1:s1s1* ,
TIME CHANGED TO *yy/mm/dd hh:mm:ss*

The time difference set by the TIMELAG command is plus (advanced) or minus (delayed) $h_1h_1:m_1m_1:s_1s_1$, and the resulting time stamp is changed to *yy/mm/dd hh:mm:ss*. The plus/minus symbol will not be displayed if advance or delay has not been specified.
S: Continues processing.
O: No action.

CLA311I

TIME DELTA IS $\left[\begin{Bmatrix}+\\-\end{Bmatrix}\right] h_1h_1{:}m_1m_1{:}s_1s_1$,

CURRENT TIME IS *yy/mm/dd hh:mm:ss*

The time difference set by the TIMELAG command is plus (advanced) or minus (delayed) $h_1h_1{:}m_1m_1{:}s_1s_1$, and the current time stamp is *yy/mm/dd hh:mm:ss*. The plus/minus symbol will not be displayed if advance or delay has not been specified.
S: Continues processing.
O: No action.

CLA315I

SYSTEM MODE CHANGED TO *processing_mode*

The system has entered the indicated processing mode – NORMAL or CE.
S: Continues processing.
O: No action.

CLA316I

SYSTEM MODE IS *processing_mode*

The system is running in the indicated processing mode – NORMAL or CE.
S: Continues processing.
O: No action.

CLA317I

DYNAMIC CF IS *status*

The Dynamic CF Dispatching is in the indicated status-ENABLE, DISABLE, or UNAVAILABLE.
S: Continues processing.
O: No action.

CLA318I

DYNAMIC CF IS CHANGED TO *status*

The Dynamic CF Dispatching has been changed to the indicated status-ENABLE or DISABLE.
S: Continues processing.
O: No action.

CLA319I

DYNAMIC CF IS ALREADY *status*

The Dynamic CF Dispatching is already in the indicated status-ENABLE or DISABLE.
S: Continues processing.
O: No action.

CLA320I

DYNAMIC CF IS UNAVAILABLE: CHP IS ICMF

The Dynamic CF Dispatching is unavailable because the channel path is ICMF. The DYNACF command execution has been rejected.
S: Continues processing.
O: No action.

`CLA330I`

MS=*MS_size*MB ES=*ES_size*MB

The main storage of the system has the capacity in megabytes indicated in decimal representation by the *MS_size*, and the expanded storage of the system has the capacity in megabytes indicated in decimal representation by the *ES_size*.
S: Continues processing.
O: No action.

`CLA331I`

IP *IP_No.:status* [*IP_No.:status* ...]

The IP indicated by the *IP_No*. is in the indicated status – + (online), – (offline) or * (damaged).
S: Continues processing.
O: No action.

`CLA332I`

$\left\{\begin{array}{l}\texttt{ISCH}\\\texttt{ICMF}\end{array}\right\}\textit{CHPID:CHP\_status}\ \texttt{[}\textit{MSGP\_status}\texttt{]}$

[*CHPID:CHP_status* [*MSGP_status*]...]

The ISCH or the ICMF indicated by the *CHPID* is in the indicated channel path status – + (online), – (offline) or * (damaged). When the channel path is online, its message path status is also shown with A (activated) or D (deactivated).
S: Continues processing.
O: No action.

`CLA333I`

$\left\{\begin{array}{l}\texttt{ISCH}\\\texttt{ICMF}\end{array}\right\}\textit{=CHPID CHP\_status}\ \texttt{[}\textit{MSGP\_status}\texttt{]}$

The ISCH or the ICMF indicated by the *CHPID* is in the indicated channel path status – ONLINE, OFFLINE or DAMAGED. When the channel path is online, its message path status is also shown with ACTIVATED or DEACTIVATED.
S: Continues processing.
O: No action.

`CLA340I`

STRUCTURE AREA
TOTAL=*total_area_size*KB (MS=*MS_area_size*KB
ES=*ES_area_size*KB) IN USE=*total_used_size*KB
(MS=*MS_used_size*KB ES=*ES_used_size*KB)

The structure area has the capacity indicated by the *total_area_size* in kilobytes, which consists of capacities of main storage (indicated in *MS_area_size* in kilobytes) and expanded storage (indicated in *ES_area_size* in kilobytes). The capacity indicated by the *total used_size* in kilobytes is used by HCCFA, which consists of capacites of main storage (indicated in *MS_used_size* in kilobytes) and expanded storage (indicated in *ES_used_size* in kilobytes). Each capacity is represented in decimal.
S: Continues processing.
O: No action.

`CLA341I`

$\texttt{STRUCT.}\ \textit{structure\_ID-}\left\{\begin{array}{l}\texttt{C}\\\texttt{L}\end{array}\right\}\texttt{[}\textit{structure\_ID-}\left\{\begin{array}{l}\texttt{C}\\\texttt{L}\end{array}\right\}\texttt{...]}$

A list of structures created in the system is displayed in pairs of the *structure_ID* and the structure type code (C for cache and L for list).
S: Continues processing.
O: No action.

CLA342I

SID=*structure_ID* TYPE= $\left\{\begin{array}{l}\texttt{CACHE}\\\texttt{LIST}\end{array}\right\}$ SIZE=*area_size*KB [CLOSED]

For the structure indicated by the *structure_ID*, the structure type (cache or list) and the allocated area size in kilobytes indicated in decimal representation by the *area_size* are displayed. When the structure is placed in the closed state, “CLOSED” is also displayed. The *area_size* may be altered when structure size is changed by dynamic reconfiguration.
S: Continues processing.
O: No action.

CLA345W

$\left\{\begin{array}{l}\texttt{CACHE}\\\texttt{LIST}\end{array}\right\}$ STRUCTURE NOT DEFINED IN SYSTEM

The indicated structure type (cache or list) is not defined in the system.
S: Continues processing.
O: No action.

CLA346W

SID=*structure_ID* NOT DEFINED IN SYSTEM

The structure indicated by the *structure_ID* is not defined in the system.
S: Continues processing.
O: No action.

CLA347W

MCP FUNCTION CLOSED

The messaging function is closed because of a failure in the Message Command Processing routine.
S: Continues processing.
O: No action.

CLA400I

IP=*IP_No.* VARIED ONLINE

The IP indicated by the *IP_ No.* has been placed into the online status.
S: Continues processing.
O: No action.

CLA401I

IP=*IP_No.* VARIED OFFLINE

The IP indicated by the *IP_ No.* has been placed into the offline status.
S: Continues processing.
O: No action.

CLA402W

IP=*IP_No.* ALREADY ONLINE

The IP indicated by the *IP_ No.* has already been placed into the online status.
S: Continues processing.
O: No action.

CLA403W

IP=*IP_No.* ALREADY OFFLINE

The IP indicated by the *IP_ No.* has already been placed into the offline status.
S: Continues processing.
O: No action.

CLA404W

IP=*IP_No.* NOT DIFINED IN CONFIGURATION

The IP indicated by the *IP_ No.* is not defined in the system.
S: Continues processing.
O: No action.

CLA405W

```
                          ⎧ONLINE PATCH RUNNING        ⎫
                          ⎪INSUFFICIENT SYSTEM STORAGE ⎪
IP=IP_No. ONLINE FAILED: ⎨IP DISABLED                 ⎬
                          ⎩SVP ACCESS ERROR            ⎭
```

The IP indicated by the *IP_ No.* cannot be placed into the online status because of the following reasons.

- ONLINE PATCH RUNNING: Online patch is running. The command is rejected until the ongoing online patch is completed.
- INSUFFICIENT SYSTEM STORAGE: System work area is insufficient since the other function uses all of the area.
- IP DISABLED: An unrecoverable error has occurred on an IP.
- SVP ACCESS ERROR: An error has been reported at an access to the SVP. Details are given by a preceding or succeeding CLA800E message.

S: Continues processing.
O: For ONLINE PATCH RUNNING and INSUFFICIENT SYSTEM STORAGE, reenter the same command strings after the inactive status is over. For other reasons, report the message ID and the message text to the field engineer.

CLA406W

IP=*IP_No.* OFFLINE FAILED: ONLINE PATCH RUNNING

The IP indicated by the *IP_ No.* cannot be placed into the offline status because online patch is running.
S: Continues processing.
O: Reenter the same command strings after the inactive status is over.

CLA407W

IP=*IP_No.* IS LAST ONLINE IP

The IP indicated by the *IP_ No.* cannot be placed into the offline status because it is the last onlined IP. At least one IP must be onlined to run HCCFA.
S: Continues processing.
O: Reenter the same command strings after the inactive status is over.

CLA410I

CHP=CHPID VARIED ONLINE

The channel path indicated by the *CHPID* has been placed into the online status.
S: Continues processing.
O: No action.

CLA411I

CHP=*CHPID* VARIED OFFLINE

The channel path indicated by the *CHPID* has been placed into the offline status.
S: Continues processing.
O: No action.

CLA412W

CHP=*CHPID* ALREADY ONLINE

The channel path indicated by the *CHPID* has already been placed into the online status.
S: Continues processing.
O: No action.

CLA413W

CHP=*CHPID* ALREADY OFFLINE

The channel path indicated by the *CHPID* has already been placed into the offline status.
S: Continues processing.
O: No action.

CLA414W

CHP=*CHPID* NOT DIFINED IN CONFIGURATION

The channel path indicated by the *CHPID* is not defined in the system configuration.
S: Continues processing.
O: No action.

CLA415W

CHP NOT DEFINED IN CONFIGURATION

No channel path is defined in the system configuration.
S: Continues processing.
O: No action.

CLA416W

```
                         ⎧ONLINE PATCH RUNNING         ⎫
                         ⎪INSUFFICIENT SYSTEM STORAGE  ⎪
CHP=CHPID ONLINE FAILED: ⎨I/O ERROR                    ⎬
                         ⎩SVP ACCESS ERROR             ⎭
```

The channel path indicated by the *CHPID* cannot be placed into the online status because of the following reasons.

- ONLINE PATCH RUNNING: Online patch is running. The command is rejected until the ongoing online patch is completed.
- INSUFFICIENT SYSTEM STORAGE: System work area is insufficient since the other function uses all of the area.
- I/O ERROR: An unrecoverable error has occurred on a channel path.
- SVP ACCESS ERROR: An error has been reported at an access to the SVP. Details are given by a preceding or succeeding CLA800E message.

S: Continues processing.
O: For ONLINE PATCH RUNNING and INSUFFICIENT SYSTEM STORAGE, reenter the same command strings after the inactive status is over. For other reasons, report the message ID and the message text to the field engineer.

`CLA417W`

```
                                 ┌ ONLINE PATCH RUNNING           ┐
CHP=CHPID OFFLINE FAILED:        │ INSUFFICIENT SYSTEM STORAGE    │
                                 │ I/O ERROR                      │
                                 └ SVP ACCESS ERROR               ┘
```

The channel path indicated by the *CHPID* cannot be placed into the offline status because of the following reasons.

- ONLINE PATCH RUNNING: Online patch is running. The command is rejected until the ongoing online patch is completed.
- INSUFFICIENT SYSTEM STORAGE: System work area is insufficient since the other function uses all of the area.
- I/O ERROR: An unrecoverable error has occurred on a channel path.
- SVP ACCESS ERROR: An error has been reported at an access to the SVP. Details are given by a preceding or succeeding CLA800E message.

S: Continues processing.

O: For ONLINE PATCH RUNNING and INSUFFICIENT SYSTEM STORAGE, reenter the same command strings after the inactive status is over. For other reasons, report the message ID and the message text to the field engineer.

`CLA418W`

`CHP=`*`CHPID`*` IS BUSY`

The channel path indicated by the *CHPID* cannot be placed into the offline status because a message path is established between the channel path and the OS-LPAR.

S: Continues processing.

O: Confirm if the specified CHPID is correct, and if so, offline the specified channel path by either of the following actions:

- Cancel the message path established with the OS-LPAR from the OS-LPAR side, and offline the channel path by the VARYOFFLINE command from the HCCFA side.
- Specify the FORCE operand of the VARYOFFLINE command and forcibly offline the channel path.

`CLA419W`

`CHP=`*`CHPID`*` IS ICMF`

The channel path indicated by the *CHPID* is ICMF and cannot be placed into the online/offline status by the VARYONLINE/VARYOFFLINE command.

S: Continues processing.

O: No action.

`CLA480I`

`CHP=`*`CHPID`*` FORCED ONLINE: PATH STATUS CHANGED`

The channel path indicated by *`CHPID`* has been forcibly placed into the online status by a certain LPAR frame or LPAR control command.

S: Places the channel path indicated by *`CHPID`* into the online status and continues processing.

O: No action.

`CLA481I`

`CHP=`*`CHPID`*` FORCED OFFLINE: PATH STATUS CHANGED`

The channel path indicated by *`CHPID`* has been forcibly placed into the offline status by a certain LPAR frame or LPAR control command.

S: Places the channel path indicated by *`CHPID`* into the offline status and continues processing.

O: No action.

```
CLA482I
```

```
IP RECONF PROCESSING NORMALLY ENDED
```

The IP configuration information has been normally updated by a certain LPAR frame.
S: Continues processing.
O: Ensure the updated IP configuration by the DISPLAY command.

```
CLA483W
```

```
                               ⎧SVP ACCESS ERROR             ⎫
IP RECONF PROCESSING FAILED:   ⎨                             ⎬
                               ⎩INSUFFICIENT SYSTEM STORAGE  ⎭
```

Updating of the IP configuration information specified by a certain LPAR frame has failed because of either of the following reasons:

- SVP ACCESS ERROR: An error has been reported at an access to the SVP. Details are given by a preceding or succeeding CLA880E message.
- INSUFFICIENT SYSTEM STORAGE: System work area is insufficient since the other function uses all of the area.

S: Continues processing without updating the IP configuration information.
O: Take either of the following actions depending on the reason:

- SVP ACCESS ERROR: Follow the instruction of CLA880E message.
- INSUFFICIENT SYSTEM STORAGE: Retry the same operation after the used status is over.

```
CLA500I
```

```
                 ⎧vvrrnn ⎫
UNIT REVISION IS ⎨       ⎬
                 ⎩UNKNOWN⎭
```

Unit revision number of HCCFA is displayed. When unit revision has not been obtained, “UNKNOWN” is displayed. In the system which supports online patch, “UNKNOWN” may be displayed since it takes time to obtain unit revision. In the system which does not support online patch, “UNKNOWN” is always displayed.
S: Continues processing.
O: Reenter the same command strings a little later when “UNKOWN” is displayed on online patch supporting system.

```
CLA501I
```

```
ONLINE PATCH COMPLETED
```

Request for online patch is accepted and its processing is completed.
S: Continues processing.
O: No action.

```
CLA502W
```

```
                      ⎧LOGOUT FUNCTION BUSY⎫
ONLINE PATCH FAILED:  ⎨TIME OVER           ⎬
                      ⎩SYSTEM ERROR        ⎭
```

Request for online patch is accepted but its processing is failed because of the following reasons.

- LOGOUT FUNCTION BUSY: Logout processing is in progress.
- TIME OVER: Online patch execution within the predefined time has failed. Details are given by a preceding or succeeding CLA880E message.
- SYSTEM ERROR: An unrecoverable error has occurred. For any SVP error, details are given by a preceding or succeeding CLA880E message. For other reasons, logout data is dumped.

S: Terminates online patch and continues HCCFA processing.
O: For LOGOUT FUNCTION BUSY and TIME OVER, reenter the same command strings after the inactive status is over. If trials of the same command execution fail, or for SYSTEM ERROR, report the message ID and the message text to the field engineer.

CLA760W

MAIN STORAGE DAMAGED: *lower_address-higher address*

A solid error on main storage (MS) area indicated by *lower_address-higher_address* is reported at the MS configuration check on HCCFA activation, and cannot use that area.
S: Closes the MS area indicated by *lower_address-higher_address* and continues processing.
O: Report the message ID and the message text to the field engineer.

CLA761W

EXTENDED STORAGE DAMAGED: *lower_frame-higher_frame*

A solid error on expanded storage (ES) indicated by *lower_frame-higher_frame* is reported at the ES configuration check on HCCFA activation, and cannot use that area. See explanation of CLA763E message for details of the frame and the frame number.
S: Closes the ES area indicated by *lower_address-higher_address* and continues processing.
O: Report the message ID and the message text to the field engineer.

CLA762E

MAIN STORAGE DAMAGED: ADDR=*address* PAGE=*page_No.*

A hardware error on main storage (MS) indicated by *address* in hexadecimal is reported at an MS access, and a real page indicated by *page_No.* in hexadecimal is closed. Details of a hardware error are given by a preceding or succeeding CLA800E message.
S: Closes the MS real page indicated by *page_No.* and continues processing.
O: Report the message ID and the message text to the field engineer.

CLA763E

EXTENDED STORAGE DAMAGED: BLOCK=*block_No.* FRAME=*frame_No.*

A hardware error on expanded storage (ES) area indicated by *block_No.* in hexadecimal is reported at an ES access, and the frame indicated by *frame_No.* in hexadecimal is closed. Block and frame are the units of access to ES by HCCFA and indicate contiguous 4-kilobyte and 256-kilobyte unit areas on ES respectively. Block No. and frame No. are the numbers to identify these blocks and frames, which define the first 4-kilobyte block and 256-kilobyte frame as 0 and have all the succeeding blocks and frames numbered in ascending numerical order.
S: Closes ES of 256 kilobytes indicated by *frame_No.* and continues processing.
O: Report the message ID and the message text to the field engineer.

CLA780E

HCCFA *routine_name* FUNCTION DAMAGED: CODE=*abend_code*

Because of the reason indicated by the *abend_code*, the HCCFA control routine indicated by the *routine_name* is damaged, causing the following effects:

- BCP (Background Command Processing routine): Processing of structure allocation is slowed.
- COMMAND (Command Control routine): Further command execution is disabled.
- CONFIG (Configuration Frame Control routine): Further VARYON/VARYOFF command execution is disabled.
- FRAME (Frame Control routine): Further frame operation for HCCFA is disabled.
- I/O PATROL (I/O Patrol routine): Further patrolling for I/O errors is disabled.
- SVP PATROL (SVP Patrol routine): Further patrolling for the SVP errors is disabled.
- LOGOUT (Logout Control routine): Further collecting of logout data is disabled.
- MCP (Message Command Processing routine): Message command processing is slowed.
- PATCH (Online Patch routine): Further execution of online patch is disabled.
- RECOVERY (Recovery Control routine): Releasing of memory area preoccupied by a failing control routine, if any, is disabled.
- SYSTEM CONTROL (System Control routine): Recovery of channel paths disabled.

S: Performs HCCFA logout, once disconnects the failing routine from HCCFA, and attempts recovery. If recovery is successful, then reconnects the routine and outputs the CLA787I message; thereafter, continues processing without causing any corresponding effect. If recovery is not successful, continues processing with the failing routine kept disconnected.
O: Report the message ID and the message text to the field engineer.

`CLA781I`

`HCCFA LOGOUT STARTED`

The HCCFA logout processing has started by the occurrence of a logout factor. Once started, any other HCCFA logout request is rejected until completion of the ongoing logout.
S: Starts the HCCFA logout processing.
O: Report the message ID and the message text to the field engineer.

`CLA782I`

LOGOUT TITLE=*logout_title*

Title of the ongoing HCCFA logout is displayed as follows:

- HCCF LOGOUT CMD: Logout caused by the LOGOUT command.
- HCCF ABEND (abend code appended): Logout caused by an abend due to a failure in one of the HCCFA control routines.
- HCCF SYSTEM FAIL: Logout caused by a failure in one of the HCCFA control routines and not leading to an abend.
- HCCF WAIT (wait code appended): Logout caused by a wait due to a failure in HCCFA.
- HCCF OPERATOR: Logout caused by activation of HCCFA restart processing.

S: Continues the HCCFA logout processing.
O: Report the message ID and the message text to the field engineer.

`CLA783W`

```
                     ⎧LOGOUT FILE ALREADY USED⎫
HCCFA LOGOUT FAILED: ⎨                        ⎬
                     ⎩I/O ERROR OCCURRED      ⎭
```

The HCCFA logout processing has failed because the logout file is busy or an unrecoverable I/O error occurred on the logout file. In the latter case, details are given by a preceding or succeeding CLA880E message.
S: Aborts the HCCFA logout processing.
O: Report the message ID and the message text to the field engineer.

`CLA784I`

`HCCFA LOGOUT COMPLETED`

The HCCFA logout processing has completed.
S: Completes the HCCFA logout processing.
O: Report the message ID and the message text to the field engineer.

`CLA785E`

HCCFA SYSTEM DAMAGED: CODE=*wait_code*

The HCCFA system has been damaged for the reason indicated by the *wait_code*.
S: Enters into a wait. If HCCFA logout has been specified, the system performs the HCCFA logout and the HCCFA restart.
O: Report the message ID and the message text to the field engineer.

`CLA786W`

`HCCFA LOGOUT FILE IS FULL`

The HCCFA logout file has been full before all logout data is dumped.
S: Terminates the HCCFA logout processing when the logout file is full.
O: IP logout is performed by the LOGOUT command, change the command operand to reduce the amount of logout data, and reenter the same command strings. If logout is performed by any other means, report the message ID and the message text to the field engineer.

```
CLA787I
```

```
HCCFA routine_name RECOVERED
```

The HCCFA control routine indicated by the *routine_name* has been recovered. This message applies to HCCFA 2.0 or above. See CLA780E message for the details of the routine name.
S: Continues processing.
O: No action.

```
CLA800E
```

```
MACHINE CHECK OCCURRED: IP=IP No. MCIC=MCIC
⎧CRW = channel_report_word⎫
⎨RC = region_code         ⎬
⎩ED = ED_code             ⎭
```

A machine check has occurred on the IP indicated by the `IP_No.` Details are indicated by the machine check interruption code (MCIC) and any of the channel report word, the region code, or the ED code.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

```
CLA810E
```

```
                         ⎧MACHINE CHECK⎫
IP=IP No. FORCED OFFLINE:⎨             ⎬
                         ⎩CHECK STOP   ⎭
```

The IP indicated by the `IP_No.` has been forcibly offlined because of a machine check or a check stop. In the former case, details are given by a preceding or succeeding CLA800E message.
S: Disconnects the subject IP and continues processing.
O: Report the message ID and the message text to the field engineer.

```
CLA850E
```

```
CHP=CHPID (subchannel_No.) I/O ERROR: error_code
SCSW=subchannel_status_word ESW=ESW1 ERW=ESW2
```

An I/O error has occurred on the channel path indicated by the `CHPID` and `subchannel_No.` Details are given by the error code, the subchannel status word (SCSW) and the extended status words (ESWs) as follows:

- Error code: Shows the I/O error type–MSE (main storage error), CHM (channel machine check), LIR (link initialization error), LKE (link error other than link initialization), CC3 (inoperable channel path reported with CC=3 at I/O processing completion), or IOE (I/O error for other reasons).
- Subchannel status word (SCSW): Shows the SCSW at the I/O error occurrence.
- Extended status word 1 (ESW1): Shows the ESW1 at the I/O error occurrence.
- Extended status word 2 (ESW2): Shows the ESW2 at the I/O error occurrence.

S: Takes the following actions depending on the error code:
- MSE: Rebuilds the I/O buffer and continues processing.
- CHM: Continues processing.
- LIR: Continues processing.
- LKE: Continues processing.
- CC3: Abends the subject I/O request.
- IOE: Abends the subject I/O request.

O: Report the message ID and the message text to the field engineer.

```
CLA851E
```

```
CHP=CHPID (subchannel_No.) I/O ERROR: SUBCHANNEL INOPERABLE
```

A subchannel-inoperable I/O error (CC=3 at activation of I/O processing) has occurred on the channel path indicated by the *CHPID* and `subchannel_No.`
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

`CLA860E`

`SUBCHANNEL=`*`subchannel_No.`*` FORCED OFFLINE: I/O ERROR`

The subchannel indicated by the *subchannel_No.* has been forcibly disconnected and placed into the offline status because I/O errors occurred on the same subchannel. Details are given by a preceding CLA850E message or a CLA851E message.
S: Places the subject subchannel into the offline status and continues processing.
O: Report the message ID and the message text to the field engineer.

`CLA861E`

`CHP=`*`CHPID`*` FORCED OFFLINE: I/O ERROR`

The channel path indicated by the *CHPID* has been forcibly disconnected and placed into the offline status by one of the following reasons. Details are given by a preceding CLA850E message or a CLA851E message.

- All the subchannels corresponding to the subject channel path were offlined.
- An unrecoverable failure occurred on the subject channel path.

S: Places the subject channel path into the offline status and continues processing.
O: Report the message ID and the message text to the field engineer.

`CLA862E`

`CHP=`*`CHPID`*` LINK FAILED`

On the channel path indicated by the *CHPID*, a channel path disconnection or a link failure has occurred. In the event of a link failure, the cause is shown by a preceding CLA850E message.
S: Continues processing with the subject channel path closed or offlined.
O: Report the message ID and the message text to the field engineer.

`CLA865W`

`SUBCHANNEL=`*`subchannel No.`*` RECOVERY IN PROGRESS: MS ERROR`

Because of a main storage error on the subchannel indicated by the *subchannel_No.*, rebuilding of the I/O buffer has been performed. Details are given by a preceding CLA850E message.
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

`CLA880E`

`SVP ACCESS ERROR: SOP=`*`suboperation_code`*
`HSW=`*`hardware_status_word`*` RC=`*`error_reference_code`*

An error has been reported at an access to the SVP. The error-detected access request is indicated by the suboperation code, and the detailed error information is indicated by the hardware status word and the error reference code.
S: Closes the failing SVP function.
O: Report the message ID and the message text to the field engineer.

`CLA881E`

```
                     ⎧TIME OVER   ⎫
SVP ACCESS FAILED:  ⎨BUSY         ⎬SOP=suboperation_code
                     ⎩SYSTEM ERROR⎭
```

The suboperation request to the SVP indicated by the *suboperation_code* has been failed because of the following reasons.

- TIME OVER: Suboperation to the SVP is requested but completion message of operation has not been reported.
- BUSY: Suboperation to the SVP is not accepted three times or more consecutively because of the SVP busy status.
- SYSTEM ERROR: An invalid suboperation is requested to the SVP.

S: Continues processing.
O: Report the message ID and the message text to the field engineer.

CLA882E

SVP FUNCTION DAMAGED: FUNC=*function_name*

An error has occurred on the SVP function indicated by the *function_name*. Details are given by a succeeding CLA880E message. .
S: Continues processing.
O: Report the message ID and the message text to the field engineer.

CLA883E

SVP FUNCTION CLOSED: FUNC=*function_name*

The SVP function is closed because three or more consecutive errors have occurred on the SVP function indicated by the *function_name*. Details are given by preceding CLA880E and CLA882E messages.
S: Closes the failing SVP function.
O: Report the message ID and the message text to the field engineer.

5 STORAGE ALLOCATION FOR HCCF-LPAR AT OS GENERATION

5.1 Overview of Storage Allocation for HCCF-LPAR

This chapter provides the estimation formulas for allocating areas and objects in HCCF-LPAR. Figure 5-1 shows the image of HCCF-LPAR main storage.

The meaning of each area is as follows:

- System area: Control area for HCCFA.
- Dump area: Area for allocating dump tables which store information useful in subsystem error analysis and troubleshooting.
- Structure area: Area for registering cache and list structures.

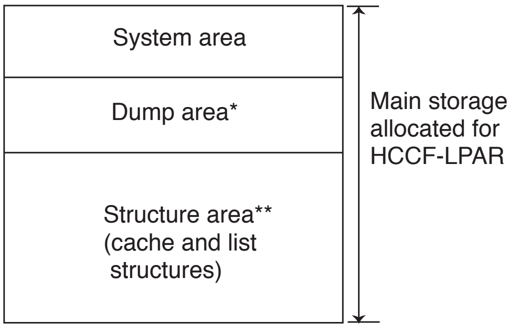

System area
Dump area*
Structure area** (cache and list structures)
Main storage allocated for HCCF-LPAR

Figure 5-1 Image of HCCF-LPAR Main Storage

* Dump area will be allocated in expanded storage (ES) if ES is defined to HCCF-LPAR.
** Elements will be allocated in ES if ES is defined to HCCF-LPAR.

5.2 Symbolic Conventions

Subsequent sections provide the estimation formulas according to the following conventions:

- Rounding up, rounding down, and boundary alignment.
  - ↑ (Formula) ↑: Rounds up the result to the nearest integer.
  - ↓ (Formula) ↓: Rounds down the result to the nearest integer.
  - ↑ (Formula) $\uparrow_n$: Rounds up the result to the nearest multiple of n. This formula equals to ↑ (Formula / n) ↑ * n.
- Selection of maximum or minimum
  - max(A,B): Selects the maximum of the two.
  - min(A,B): Selects the minimum of the two.
- Multiplication and power
  - A*B: Multiplies A by B.
  - A**B: Raises A to the Bth power.
- Size and unit
  - KB: Means kilobyte (= 1,024 bytes).
  - MB: Means megabyte (= 1,048,576 bytes = 1,024 KB).
  - [X]: Means unit X.

5.3 System Area Allocation for HCCF-LPAR

System area size is estimated by the formulas shown below. In each formula, parameters and variants represent the following:

- `MS_SIZE`: Main storage size in megabytes defined to HCCF-LPAR.
- `ES_SIZE`: Expanded storage size in megabytes defined to HCCF-LPAR.
- `CHP_CNT`: The number of channel paths defined to HCCF-LPAR.
- `fixed_alloc`: Allocation area size factor not affected by any parameter.
- `ms_alloc`: Allocation area size factor determined by MS_SIZE.
- `es_alloc`: Allocation area size factor determined by ES_SIZE.
- `chp_alloc`: Allocation area size factor determined by CHP_CNT.

In ISCH or IC environment, the system area size is estimated by the formula

```
↑(fixed_alloc + ms_alloc + es_alloc + chp_alloc)↑256KB
```

where the variants denote the following:

- `fixed_alloc: 4325376`
- `ms_alloc: ↑(MS_SIZE / 256)↑ * 550912 + MS_SIZE * 32`
- `es_alloc: ↑(ES_SIZE / 256)↑ * 20480 + ES_SIZE * 32`
- `chp_alloc:`
  `↑(CHP_CNT / 4)↑ * 4 * 49152 + (↑(CHP_CNT / 4)↑ * 4 * 2`
  `+ ↑(↑(CHP_CNT / 4)↑ * 4 / 8)↑ + 1) * 65536`

In ICMF environment, the system area size is estimated by the formula

```
↑(fixed_alloc + ms_alloc + es_alloc + chp_alloc)↑256KB
```

where the variants denote the following:

- fixed_alloc: 4325376
- `ms_alloc: ↑(MS_SIZE / 256)↑ * 550912 + MS_SIZE * 32`
- `es_alloc: ↑(ES_SIZE / 256)↑ * 20480 + ES_SIZE * 32`
- `chp_alloc:`
  `CHP_CNT * 49152 + (CHP_CNT * 2+ N(CHP_CNT / 8)N + 1) * 65536`

5.4 Dump Area Allocation for HCCF-LPAR

Dump area is the area reserved in HCCF storage to allocate dump tables. Dump tables store information useful in subsystem error analysis and troubleshooting.

Note: *Dump area size is specified by the DUMPSPACE parameter of the coupling facility resource management (CFRM) policy.*

This size of each dump table depends on the scope of dump information and the features of structure to be stored in the table. The features of structure include the number of entries registered to the structure, and presence or absence of any subarea.

HCCF allocates dump area by rounding up the specified size to the nearest multiple of the storage increment. Multiple dump tables can be allocated at a time, and hence the dump area size should be larger than the total size of all dump tables.

HCCF has the function to respond with the maximum dump area size among those previously used. The response may be useful as one of the factors in determining the dump area size.

If expanded storage (ES) has been defined to the subject HCCF-LPAR and such HCCF ES has enough free space at the time of dump area allocation, dump areas are allocated predominantly to ES.

5.5 Structure Area Allocation for HCCF-LPAR

5.5.1 Overview of Structure Area Allocation

HCCF allocates each structure area from the target structure size and the parameters specified by MVS. HCCF responds to MVS with the actually allocated structure size and the object count possible to be registered. The object count means the directory entry count and the data area element count of a cache structure, the list entry count and the list element count of a list structure (including the case of being used as a lock structure), and the event monitor control count of a keyed list structure.

Note: *The target structure size is determined by the SIZE parameter of the CFRM policy or by the STRSIZE operand of the IXLCONN macro. The value of the SIZE parameter of the CFRM policy or the value of the STRSIZE operand of the IXLCONN macro is determined by the estimated target structure size.*

For estimation of the structure area size, the following calculation methods are provided:

- Estimation of structure size: The target structure size is estimated from the necessary object size. This estimation is useful when the object count of an already allocated structure is altered or the object count of a structure to be newly allocated is predefined.
- Estimation of object count: The object count is estimated from the (target) structure size. This estimation is useful when the size of an already allocated structure is altered or the size of a structure to be newly allocated is predefined.

An HCCF-LPAR can define expanded storage (ES) in addition to main storage (MS). As a result, allocation of structure(s) in a total size of more than 2 gigabytes is allowed. However, since the response time to an element in ES is larger than that to an element in MS, it is recommended to use only MS if the total structure size is 2 gigabytes or less.

If ES has been defined to the subject HCCF-LPAR and such HCCF ES has enough free space at the time of structure allocation, data area elements or list elements are predominantly allocated to ES and the other objects are allocated to MS. When ES is not defined, all the objects are allocated to MS.

If the HCCF-LPAR does not have enough free space in MS to allocate objects other than data area elements and list elements, structure allocation is possible but not the structure in the specified size. To work around the case, estimating a necessary minimum MS size by calculation is available.

The MS size of an HCCF-LPAR must be larger than the sum of the system area size, the dump area size, and the area size for structures allocated at a time. The ES size of an HCCF-LPAR must be larger than the sum of the necessary minimum MS size and the system area size.

The results of calculated estimations may slightly differ from actually allocated structure sizes or object counts because of the following reasons:

- Boundary alignment: HCCF allocates structures by rounding up the specified target structure size to the nearest storage increment.
- Possible shortage of MS: When the HCCF-LPAR does not have enough free space in MS to allocate a structure in the specified size, HCCF allocates a structure smaller than the specified one so that at least one or more objects can be reserved.
- Characteristic of structure allocation: Actually allocated ratio of entry (directory entry or list entry) to element (data area element or list element) may slightly differ from the specified input because HCCF allocates a structure so as to reserve as many objects as possible at the ratio nearest to the specified one.
- Simplification of calculation: Certain estimation formulas are simplified to escape from being excessively complicated.

5.5.2 Calculation Basics for Estimation of Structure Area Size

Estimation of cache structure size

An estimated structure size (STR_SIZE) of a cache structure is obtained from the parameters of directory entry count (DE_CNT) and data area element count (DAE_CNT).

Note: *Each STR_SIZE must be quadrupled when being specified as the SIZE parameter of the CFRM policy or as the STRSIZE operand of the IXLCONN macro.*

HCCF allocates each cache structure by rounding up the specified structure size to the nearest multiple of the storage increment. Hence, the actually allocated directory entry count and data area element count returned to MVS may slightly differ from the specified DE_CNT and DAE_CNT.

Note: *When the actual allocation falls short of the expected entry or element count, it is recommended that a larger value is specified as the SIZE parameter of the CFRM policy or the STRSIZE operand of the IXLCONN macro.*

Estimation of list structure size

Normally, an estimated structure size (STR_SIZE) of a list structure, whether keyed or not, is obtained from the parameters of list entry count (LE_CNT) and list element count (LEL_CNT). As an alternative method, an estimated structure size (STR_SIZE) of a keyed list structure can be obtained from the parameter of event monitor control count (EM_CNT).

Note: *Each STR_SIZE must be quadrupled when being specified as the SIZE parameter of the CFRM policy or as the STRSIZE operand of the IXLCONN macro.*

HCCF allocates each list structure by rounding up the specified structure size to the nearest multiple of the storage increment. Hence, the actually allocated list entry count, list element count, and event monitor control count returned to MVS may slightly differ from the specified LE_CNT, LEL_CNT, and EM_CNT.

Note: *When the actual allocation falls short of the expected entry or element count, it is recommended that a larger value is specified as the SIZE parameter of the CFRM policy or the STRSIZE operand of the IXLCONN macro.*

Estimation of cache structure object count

An estimated directory entry count (DE_CNT) of a cache structure is obtained from the parameter of data area element count (DAE_CNT) or structure size (STR_SIZE). Similarly, an estimated data area element count (DAE_CNT) of a cache structure is obtained from the parameter of directory entry count (DE_CNT) or structure size (STR_SIZE).

HCCF allocates each cache structure by rounding up the specified structure size to the nearest multiple of the storage increment. Hence, the actually allocated directory entry count and data area element count returned to MVS may slightly differ from the specified DE_CNT and DAE_CNT.

Note: *When the actual allocation falls short of the expected entry or element count, it is recommended that a larger value is specified as the SIZE parameter of the CFRM policy or the STRSIZE operand of the IXLCONN macro.*

Estimation of list structure object count

Normally, an estimated list entry count (LE_CNT) of a list structure, whether keyed or not, is obtained from the parameter of list element count (LEL_CNT) or structure size (STR_SIZE); similarly, an estimated list element count (LEL_CNT) of a list structure is obtained from the parameter of list entry count (LE_CNT) or structure size (STR_SIZE). As an alternative method, such estimations of a keyed list structure can be obtained from the parameter of event monitor control count (EM_CNT).

HCCF allocates each list structure by rounding up the specified structure size to the nearest multiple of the storage increment. Hence, the actually allocated list entry count and list element count returned to MVS may slightly differ from the specified LE_CNT and LEL_CNT.

Note: *When the actual allocation falls short of the expected entry or element count, it is recommended that a larger value is specified as the SIZE parameter of the CFRM policy or the STRSIZE operand of the IXL-CONN macro.*

5.5.3 Parameters and Variants in Structure Area Size Estimation Formulas

The estimation formulas for structure area allcation use the following parameters and variants as applicable depending on the HCCFA release level and the choice of estimation method.

Parameters in structure area size estimation formulas

- `ADJ_FLG`: Flag to identify whether any adjunct data of list entry is allocated (= 1) or not (= 0) to a structure (cache or list). This flag being “1” applies when a structure is allocated by the IXLCONN macro with `ADJUNCT=YES` specified. This flag being “0” applies when a structure is allocated by the IXLCONN macro with `ADJUNCT=NO` specified.
- `CO_CLASS`: The number of castout classes for a cache structure. When allocating a cache structure by the IXLCONN macro, this parameter assumes the value specified by the NUMCOCLASS operand of the IXLCONN macro.
- `DAE_CHAR`: Data area element characteristic used in calculating the data area element size for a cache structure. When allocating a cache structure by the IXLCONN macro, this input parameter assumes the value specified by the ELEMCHAR operand of the IXLCONN macro.
- `DAE_CNT`: The total number of data area elements reserved at cache structure allocation. Represented as `↓(DE_CNT * ELEM_RATIO / ENT_RATIO)↓.`
- `DE_CNT`: The total number of directory entries reserved at cache structure allocation. Represented as `↓(DAE_CNT * ENT_RATIO / ELEM_RATIO)↓.`
- `ELEM_RATIO`: Ratio of the target element count to the target entry count to be specified at structure allocation (cache or list). When allocating a structure by the IXLCONN macro, this parameter assumes the value specified by the ELEMENTRATIO operand of the IXLCONN macro.
- `EM_CNT`: The total number of event monitor controls reserved at list structure allocation. Only effective when a list structure is allocated by the IXLCONN macro with EFOPTION=KEY specified, in which case the EMCSTGPCT option setting must be equal to the result of the following formula: `em_size / obj_size * 10000.`
- `ENT_RATIO`: Ratio of the target entry count to the target element count to be specified at structure allocation (cache or list). When allocating a structure by the IXLCONN macro, this parameter assumes the value specified by the ENTRYRATIO operand of the IXLCONN macro.
- `KEY_FLG`: Flag to identify whether any key is allocated (= 1) or not (= 0) to a list entry of a list structure. This flag being “1” applies when a list structure is allocated by the IXLCONN macro with `REFOPTION=KEY` specified. This flag being “0” applies when a list structure is allocated by the IXLCONN macro with `REFOPTION=NOKEYNAME` or `REFOPTION=NAME` specified.
- `L_CNT`: The total number of lists for a list structure. When allocating a list structure by the IXLCONN macro, this parameter assumes the value specified by the LISTHEADERS operand of the IXLCONN macro.
- `LE_CNT`: The total number of list entries reserved at list structure allocation. Represented as `↓(LEL_CNT * ENT_RATIO / ELEM_RATIO)↓.`
- `LEL_CHAR`: List element characteristic used in calculating the list element size for a list structure. When allocating a list structure by the IXLCONN macro, this parameter assumes the value specified by the ELEMCHAR operand of the IXLCONN macro.
- `LEL_CNT`: The total number of list elements reserved at list structure allocation. Represented as `↓(LE_CNT * ELEM_RATIO / ENT_RATIO)↓.`
- `LEL_FLG`: Flag to identify whether any list element is allocated (= 1) or not (= 0) to a list structure. This flag being “1” applies when a list structure is allocated by the IXLCONN macro with `MAXELEMNUM=1` specified. This flag being “0” applies when a list structure is allocated by the IXLCONN macro with `MAXELEMNUM=0` specified.
- `LTE_CHAR`: Lock table entry characteristic used in calculating the lock table entry size for a list structure. When allocating a list structure by the IXLCONN macro, this parameter assumes the value specified by the NUMUSERS operand of the IXLCONN macro.

- `LTE_CNT`: The total number of lock table entries reserved at list structure allocation. When allocating a list structure by the IXLCONN macro, this parameter assumes the value specified by the LOCKENTRIES operand of the IXLCONN macro.
- `MAX_DA`: Maximum data area element size per directory entry for a cache structure. When allocating a cache structure by the IXLCONN macro, this parameter assumes the value specified by the MAXELEMNUM operand of the IXLCONN macro.
- `MAX_DLE`: Maximum number of list elements per list entry for a list structure. When allocating a list structure by the IXLCONN macro, this parameter assumes the value specified by the MAXELEMNUM operand of the IXLCONN macro.
- `MAX_STR`: Maximum structure size in units of 4KB.
- `MS_SIZE`: MS size of the target HCCF-LPAR in megabytes.
- `NAME_FLG`: Flag to identify whether any name is allocated (= 1) or not (= 0) to a list structure. This flag being “1” applies when a list structure is allocated by the IXLCONN macro with `REFOPTION=NAME` specified. This flag being “0” applies when a list structure is allocated by the IXLCONN macro with `REFOPTION=NOKEYNAME` or `REFOPTION=KEY` specified.
- `STG_CLASS`: Maximum number of storage classes for a cache structure. When allocating a cache structure by the IXLCONN macro, this parameter assumes the value specified by the NUMSTGCLASS operand of the IXLCONN macro.
- `STR_SIZE`: Structure size of the target structure. Such target structure size is obtained by quartering the value of the SIZE parameter of the CFRM policy or by quartering the value of the STRSIZE operand of the IXLCONN macro.

Variants in structure area size estimation formulas

- `ctl_size:` Control size.
- `data_size:` Data size.
- `de_page:` Directory entry page.
- `de_size:` Directory entry size.
- `de_unit:` Directory entry unit.
- `elem_ovh:` Element control overhead.
- `elem_size:` Element size.
- `elem_unit`: Element unit.
- `em_ovh`: Event monitor control overhead (for keyed list structure).
- `em_size`: Event monitor control size (for keyed list structure).
- `ent_ovh`: Entry control overhead.
- `ent_size`: Entry size.
- `le_page`: List entry page.
- `le_size`: List entry size.
- `le_unit`: List entry unit.
- `max_de`: Maximum directory entry.
- `max_le`: Maximum list entry.
- `max_ms`: Maximum MS size.
- `name_ovh`: Name allocation overhead.
- `obj_page`: Object page.
- `obj_size`: Object size.
- `robj_size`: Object size for allocation of event monitor control (for keyed list structure).

5.5.4 Structure Area Size Estimation Formulas

5.5.4.1 Cache Structure Size and Necessary Minimum MS Size

An estimated structure size (STR_SIZE) and a necessary minimun MS size (MIN_MS) are represented by the formulas shown below, depending on the case of allocation such as absence of data area element (indicated by the MAX_DA parameter being 0).

Case-1: MAX_DA = 0; max. structure size known

```
STR_SIZE
  = MIN_MS
  = ↑((ctl_size + DE_CNT * ent_size + ent_ovh) / 4096 * 1.03)↑
    [4KB]
```

- $\texttt{ctl\_size: } \uparrow(\texttt{40960 + CO\_CLASS * 20 + STG\_CLASS * 416})\uparrow_{4096}$
- `ent_size: 208`
- `ent_ovh:`
  $\downarrow((\texttt{MAX\_STR * 4096 - ctl\_size) / (ent\_size + 5)})\downarrow \texttt{ * 4}$
  $\texttt{+ } \uparrow((\texttt{MAX\_STR * 4096 - ctl\_size) / (ent\_size + 5)})\uparrow_{8192}$

Case-2: MAX_DA = 0; target = max. structure

```
STR_SIZE
  = MIN_MS
  = ↑((ctl_size + DE_CNT * ent_size + ent_ovh) / 4096 * 1.03)↑
    [4KB]
```

- $\texttt{ctl\_size: } \uparrow(\texttt{40960 + CO\_CLASS * 20 + STG\_CLASS * 416})\uparrow_{4096}$
- `ent_size: 208`
- $\texttt{ent\_ovh: DE\_CNT * 4 + } \uparrow(\texttt{DE\_CNT})\uparrow_{8192}$

Case-3: MAX_DA = 0; max. structure size unknown; target ≠ max. structure

Take the following two steps:

1. Calculate the maximum structure size by the formula of Case-2.
2. Calculate the target structure size (STR_SIZE) and the necessary minimum MS size (MIN_MS) by the formula of Case-1 assuming the value obtained above as MAX_STR.

Case-4: MAX_DA ≠ 0; max. structure size known

-
  ```
  STR_SIZE
  = ↑((ctl_size + DE_CNT * ent_size + DAE_CNT * elem_size
    + ent_ovh + elem_ovh) / 4096 * 1.03)↑ [4KB]
  ```

-
  ```
  MIN_MS
  = ↑((ctl_size + DE_CNT * ent_size + ent_ovh + elem_ovh)
    / 4096 * 1.03)↑ [4KB]
  ```

  -
    ```
    ctl_size: ↑(40960 + CO_CLASS * 20 + STG_CLASS * 416)↑4096
    ```

  -
    ```
    ent_size: 208 + ADJ_FLG * 64
    ```

  -
    ```
    elem_size: 256 *(2 ** DAE_CHAR)
    ```

  -
    ```
    ent_ovh:
    ↓((MAX_STR * 4096 - ctl_size)* elem_size
    / (5*elem_size + 12 * ent_size + ent_size * elem_size))↓ * 4
    + ↑((MAX_STR * 4096 - ctl_size)* elem_size / (5 * elem_size
    + 12 * ent_size + ent_size * elem_size))↑8192
    ```

  -
    ```
    elem_ovh:
    ↓((MAX_STR * 4096 - ctl_size)* ent_size / (5 * elem_size
    + 12 * ent_size + ent_size * elem_size))↓ * 12
    ```

Case-5: MAX_DA ≠ 0; target = max. structure

-
  ```
  STR_SIZE
  = ↑((ctl_size + DE_CNT * ent_size + DAE_CNT * elem_size
    + ent_ovh + elem_ovh) / 4096*1.03)↑ [4KB]
  ```

-
  ```
  MIN_MS
  = ↑((ctl_size + DE_CNT * ent_size + ent_ovh + elem_ovh)
    / 4096 * 1.03)↑ [4KB]
  ```

  -
    ```
    ctl_size: ↑(40960 + CO_CLASS * 20 + STG_CLASS * 416)↑4096
    ```

  -
    ```
    ent_size: 208 + ADJ_FLG * 64
    ```

  -
    ```
    elem_size: 256 * (2 ** DAE_CHAR)
    ```

  -
    ```
    ent_ovh:
    ↓(obj_size / ent_size)↓ * 4 + ↑(obj_size / ent_size)↑8192
    ```

  -
    ```
    obj_size: ↑(DE_CNT * ent_size + DAE_CNT * elem_size)↑4096
    ```

  -
    ```
    elem_ovh: ↓(obj_size / elem_size)↓ * 12
    ```

Case-6: MAX_DA ≠ 0; max. structure size unknown; target ≠ max. structure

Take the following two steps:

1. Calculate the maximum structure size by the formula of Case-5.
2. Calculate the target structure size (STR_SIZE) and the necessary minimum MS size (MIN_MS) by the formulas of Case-4 assuming the value obtained above as MAX_STR.

5.5.4.2 List Structure Size and Necessary Minimum MS Size

An estimated structure size (STR_SIZE) and a necessary minimun MS size (MIN_MS) are represented by the formulas shown below, depending on the case of allocation such as absence of list element (indicated by the LEL_FLG parameter being 0).

Case-1: ADJ_FLG = LEL_FLG = 0

Normally, STR_SIZE and MIN_MS are obtained by the following equations:

- STR_SIZE
= ↑((5120 + LTE_CNT * (2 ** LTE_CHAR)) / 4096 * 1.03)↑ [4KB]
- MIN_MS = STR_SIZE [4KB]

As an alternative with HCCFA 3.1, STR_SIZE and MIN_MS can be obtained by the following equations:

- STR_SIZE
= ↑((5120 + em_ovh + LTE_CNT * (2 ** LTE_CHAR)) / 4096
* 1.03)↑ [4KB] + em_size
- MIN_MS = STR_SIZE [4KB]
  - em_ovh: KEY_FLG * 2048
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

Case-2: LEL_FLG = 0; max. structure size known

Normally, STR_SIZE and MIN_MS are obtained by the following equations:

- STR_SIZE
= ↑((ctl_size + LE_CNT * ent_size + name_ovh + ent_ovh)
/ 4096 * 1.03)↑ [4KB]
- MIN_MS = STR_SIZE [4KB]
  - ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑4096
  - ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32
  - name_ovh:
NAME_FLG * ↑((MAX_STR * 4096 - ctl_size) / (ent_size + 5))↑8192
  - ent_ovh:
↓((MAX_STR * 4096 - ctl_size) / (ent_size + 4 + NAME_FLG * 1))↓ * 4

As an alternative, STR_SIZE and MIN_MS can be obtained by the following equations:

- STR_SIZE
= ↑ + em_ovh + LE_CNT * ent_size + name_ovh + ent_ovh)
/ 4096 * 1.03)↑ [4KB] + em_size
- MIN_MS = STR_SIZE [4KB]
  - ctl_size: Same as defined above.
  - em_ovh: KEY_FLG * 2048
  - ent_size: Same as defined above.
  - name_ovh: Same as defined above.
  - ent_ovh: Same as defined above.
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

Case-3: LEL_FLG = 0; target = max. structure

Normally, STR_SIZE and MIN_MS are obtained by the following equations:

- STR_SIZE
  = ↑((ctl_size + LE_CNT * ent_size + name_ovh + ent_ovh)
  / 4096 * 1.03)↑ [4KB]
- MIN_MS = STR_SIZE [4KB]
  - ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑$_{4096}$
  - ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32
  - name_ovh: NAME_FLG * ↑(LE_CNT)↑$_{8192}$
  - ent_ovh: LE_CNT * 4

As an alternative, STR_SIZE and MIN_MS can be obtained by the following equations:

- STR_SIZE
  = ↑ + em_ovh + LE_CNT * ent_size + name_ovh + ent_ovh)
  / 4096 * 1.03)↑ [4KB] + em_size
- MIN_MS = STR_SIZE [4KB]
  - ctl_size: Same as defined above.
  - em_ovh: KEY_FLG * 2048
  - ent_size: Same as defined above.
  - name_ovh: Same as defined above.
  - ent_ovh: Same as defined above.
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

Case-4: LEL_FLG = 0; max. structure size unknown; target ≠ max. structure

Take the following two steps:

1. Calculate the maximum structure size by the formula of Case-3.
2. Calculate the target structure size (STR_SIZE) and the necessary minimum MS size (MIN_MS) by the formula of Case-2 assuming the value obtained above as MAX_STR.

Case-5: LEL_FLG ≠ 0; max. structure size known

Normally, STR_SIZE and MIN_MS are obtained by the following equations:

- STR_SIZE
  = ↑((ctl_size + LE_CNT * ent_size + LEL_CNT * elem_size
  #+ name_ovh + ent_ovh + elem_ovh) / 4096 * 1.03)↑ [4KB]
- MIN_MS
  = ↑((ctl_size + LE_CNT * ent_size + name_ovh + ent_ovh
  + elem_ovh) / 4096 * 1.03)↑ [4KB]
  - ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑4096
  - ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32
  - elem_size: 256 * (2 ** LEL_CHAR)
  - name_ovh:
    NAME_FLG * ↑((MAX_STR * 4096 - ctl_size) * elem_size
    / (5 * elem_size + 12 * ent_size + ent_size * elem_size))↑8192
  - ent_ovh:
    ↓((MAX_STR * 4096 - ctl_size) * elem_size / ((4 + NAME_FLG * 1)
    * elem_size + 12 * ent_size + ent_size * elem_size))↓ * 4
  - elem_ovh:
    ↓((MAX_STR * 4096 - ctl_size) * ent_size / ((4 + NAME_FLG * 1)
    * elem_size + 12 * ent_size + ent_size * elem_size))↓ * 12

As an alternative, STR_SIZE and MIN_MS can be obtained by the following equations:

- STR_SIZE
  = ↑((ctl_size + em_ovh + LE_CNT * ent_size + LEL_CNT * elem_size
  + name_ovh + ent_ovh + elem_ovh) / 4096 * 1.03)↑ [4KB]
  + em_size
- MIN_MS
  = ↑((ctl_size + em_ovh + LE_CNT * ent_size + name_ovh + ent_ovh
  + elem_ovh) / 4096 * 1.03)↑ [4KB]
  - ctl_size: Same as defined above.
  - em_ovh: KEY_FLG * 2048
  - ent_size: Same as defined above.
  - elem_size: Same as defined above.
  - name_ovh: Same as defined above
  - ent_ovh: Same as defined above.
  - elem_ovh: Same as defined above.
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

Case-6: LEL_FLG ≠ 0; target = max. structure

Normally, STR_SIZE and MIN_MS are obtained by the following equations:

- STR_SIZE
  = ↑((ctl_size + LE_CNT * ent_size + LEL_CNT * elem_size
  + name_ovh + ent_ovh + elem_ovh) / 4096 * 1.03)↑ [4KB]
- MIN_MS
  = ↑((ctl_size + LE_CNT * ent_size + name_ovh + ent_ovh
  + elem_ovh) / 4096 * 1.03)↑ [4KB]
  - ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑$_{4096}$
  - ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32
  - elem_size: 256 * (2 ** LEL_CHAR)
  - name_ovh: NAME_FLG * ↑(obj_size / ent_size)↑$_{8192}$
  - obj_size: ↑(LE_CNT * ent_size + LEL_CNT * elem_size)↑$_{4096}$
  - ent_ovh: ↓(obj_size / ent_size)↓ * 4
  - elem_ovh: ↓(obj_size / elem_size)↓ * 12

As an alternative, STR_SIZE and MIN_MS can be obtained by the following equations:

- STR_SIZE
  = ↑((ctl_size + em_ovh + LE_CNT * ent_size + LEL_CNT * elem_size
  + name_ovh + ent_ovh + elem_ovh) / 4096 * 1.03)↑ [4KB] + em_size
- MIN_MS
  = ↑((ctl_size + em_ovh + LE_CNT * ent_size + name_ovh + ent_ovh
  + elem_ovh) / 4096 * 1.03)↑ [4KB]
  - ctl_size: Same as defined above.
  - em_ovh: KEY_FLG * 2048
  - ent_size: Same as defined above.
  - elem_size: Same as defined above.
  - name_ovh: Same as defined above.
  - obj_size: Same as defined above.
  - ent_ovh: Same as defined above.
  - elem_ovh: Same as defined above.
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

Case-7: LEL_FLG ≠ 0; max. structure size unknown; target structure ≠ max. structure

Take the following two steps:

1. Calculate the maximum structure size by the formula of Case-6.
2. Calculate the target structure size (STR_SIZE) and necessary minimum MS size (MIN_MS) by the formulas of Case-5 assuming the value obtained above as MAX_STR.

5.5.4.3 Cache Structure Object Count

An estimated directory entry count (DE_CNT) and/or data area element count (DAE_CNT) are represented by the formulas shown below, depending on the case of allocation such as absence of data area element (indicated by the MAX_DA parameter being 0).

Case-1: Structure size unknown

- DE_CNT = ↓(DAE_CNT * ENT_RATIO / ELEM_RATIO)↓ [number]
- DAE_CNT = ↓(DE_CNT * ELEM_RATIO / ENT_RATIO)↓ [number]

Case-2: Structure size known; MAX_DA = 0

DE_CNT = ↓(obj_size / ent_size)↓ [number]

- obj_size:
  ↑(↓(STG_SIZE * 4096 * 0.97)↓ - ctl_size - ent_ovh)↑$_{4096}$
- ctl_size: ↑(40960 + CO_CLASS * 20 + STG_CLASS * 416)↑$_{4096}$
- ent_ovh:
  ↓((MAX_STR * 4096 - ctl_size) / (ent_size + 5))↓ * 4
  + ↑((MAX_STR * 4096 - ctl_size) / (ent_size + 5))↑$_{8192}$
- ent_size: 208

Case-3: Structure size known; MAX_DA ≠ 0

- DE_CNT
  = ↓(ENT_RATIO * obj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
- DAE_CNT
  = ↓(ELEM_RATIO * obj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
  - obj_size:
    ↑(↓(STR_SIZE * 4096 * 0.97)↓ - ctl_size - ent_ovh
    - elem_ovh)↑$_{4096}$
  - ctl_size: ↑(40960 + CO_CLASS * 20 + STG_CLASS * 416)↑$_{4096}$
  - ent_ovh:
    ↓((max_ms - ctl_size) * elem_size
    / (5 * elem_size + 12* ent_size + ent_size * elem_size))↓ * 4
    + ↑((max_ms - ctl_size) * elem_size
    / (5 * elem_size + 12 * ent_size + ent_size * elem_size))↑$_{8192}$
  - max_ms: min(MAX_STR * 4096,MS_SIZE * 1048576)
  - elem_ovh:
    ↓((MAX_STR * 4096 - ctl_size) * ent_size / (5 * elem_size
    + 12 * ent_size + ent_size * elem_size))↓ * 12
  - ent_size: 208 + ADJ_FLG * 64
  - elem_size: 256 * (2 ** DAE_CHAR)

5.5.4.4 List Structure Object Count

An estimated list entry count (LE_CNT) and/or list element count (LEL_CNT) are represented by the formulas shown below, depending on the case of allocation such as absence of data area element (indicated by the LEL_FLG parameter being 0).

Case-1: Structure size unknown

- LE_CNT = ↓(LEL_CNT * ENT_RATIO / ELEM_RATIO)↓ [number]
- LEL_CNT = ↓(LE_CNT * ELEM_RATIO / ENT_RATIO)↓ [number]

Case-2: Structure size known; LEL_FLG = 0

Normally, LE_CNT is obtained by the following equation:

LE_CNT = Ø(obj_size / ent_size)Ø [number]

- obj_size:
  ↑(↓(STR_SIZE * 4096 * 0.97)↓ - ctl_size - name_ovh
  - ent_ovh)↑$_{4096}$
- ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑$_{4096}$
- name_ovh:
  NAME_FLG * ↑((MAX_STR * 4096 - ctl_size) / (ent_size + 5))↑$_{8192}$
- ent_ovh:
  ↓((MAX_STR * 4096 - ctl_size) / (ent_size + 4 + NAME_FLG
  * 1))↓ * 4
- ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32

As an alternative, LE_CNT can be obtained by the following equation:

LE_CNT = ↓(robj_size / ent_size)↓ [number]

- robj_size: obj_size - em_size
- obj_size:
  ↑(↓(STR_SIZE * 4096 * 0.97)↓ - ctl_size - name_ovh
  - ent_ovh)↑4096
- ctl_size:
  ↑(5120 + em_ovh + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR))↑$_{4096}$
- em_ovh: KEY_FLG * 2048
- name_ovh: Same as defined above.
- ent_ovh: Same as defined above.
- em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K
- ent_size: Same as defined above.

Case-3: Structure size known; LEL_FLG ≠ 0

Normally, LE_CNT and LEL_CNT are obtained by the following equation:

- LE_CNT
  =↓(ENT_RATIO * obj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
- LEL_CNT
  =↓(ELEM_RATIO * obj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
  - obj_size:
    ↑(↓(STR_SIZE * 4096 * 0.97)↓ - ctl_size - name_ovh - ent_ovh
    - elem_ovh)N4096
  - ctl_size: ↑(5120 + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR)↑4096
  - name_ovh:
    NAME_FLG * ↑((max_ms - ctl_size) * elem_size
    / (5 * elem_size + 12 * ent_size + ent_size * elem_size))↑8192
  - max_ms: min(MAX_STR * 4096,MS_SIZE * 1048576)
  - ent_ovh:
    ↓((max_ms - ctl_size) * elem_size / ((4 + NAME_FLG * 1)
    * elem_size + 12 * ent_size + ent_size * elem_size))↓ * 4
  - elem_ovh:
    ↓((MAX_STR * 4096 - ctl_size) * ent_size/ ((4 + NAME_FLG * 1)
    * elem_size + 12 * ent_size + ent_size * elem_size))↓ * 12
  - ent_size: 64 + ADJ_FLG * 64 + KEY_FLG * 32 + NAME_FLG * 32
  - elem_size: 256 * (2 ** LEL_CHAR)

As an alternative, LE_CNT and LEL_CNT can be obtained by the following equations:

- LE_CNT
  = ↓(ENT_RATIO * robj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
- LEL_CNT
  = ↓(ELEM_RATIO * robj_size / (ENT_RATIO * ent_size
  + ELEM_RATIO * elem_size))↓ [number]
  - robj_size: obj_size - em_size
  - obj_size:
    ↑(↓(STR_SIZE * 4096 * 0.97)↓ - ctl_size - name_ovh - ent_ovh
    - elem_ovh)↑4096
  - ctl_size:
    ↑(5120 + em_ovh + L_CNT * 608 + LTE_CNT * (2 ** LTE_CHAR)↑4096
  - em_ovh: KEY_FLG * 2048
  - name_ovh: Same as defined above.
  - max_ms: Same as defined above.
  - ent_ovh: Same as defined above.
  - elem_ovh: Same as defined above.
  - ent_size: Same as defined above.
  - elem_size: Same as defined above.
  - em_size: KEY_FLG * (EM_CNT + 3317 - 1) / 3317 * 256K

6 HCCF OPERATIONAL LEVEL SUPPORTED BY HCCFA

6.1 HCCF Levels Supported by HCCFA 4.0 and 5.0

Table 6-1 shows the HCCF level (equivalent to IBM's CF level) supported by HCCFA 4.0 and 5.0. The user's HCCFA version can be confirmed with the CLA201I message displayed at HCCF activation.

HCCF level 4 supports Parallel Sysplex functions of IBM's CF level 4; HCCF level 5 supports Parallel Sysplex functions of IBM's CF level 5. However, the following features supported in IBM's CF level 4 or 5 are not supported in HCCF level 4 or 5.

- Battery Backup feature
- VM/ESA guest coupling simulation

This HCCF level is supported by MVS/ESA SP 5.2 or above.

Table 6-1 HCCF Level Supported by HCCFA 4.0 and 5.0

HCCFA Version	**HCCF Level**
04-00 (4.0)	4
05-00 (5.0)	5

6.2 Deviations in Coupling Facility Limits

Table 6-2 shows coupling facility limits for HCCF level 4, which may differ from those supported by IBM's CF level 4.

Table 6-2 Deviations in coupling Facility Limits

Coupling Facility Limit	**HCCF Level 4**	**HCCF Level 5**
Maximum number of logical IPs	10	10
Facility information size	64 bytes	64 bytes
Storage increment	256 KB	256 KB
Storage class limit	63	63
Castout class limit	1024	1024
Local cache identifier limit	255	255
Retry buffer limit	151	263
Structure ID limit	1023	1023
User identifier limit	32	32
List number limit	4873645	4472286
Maximum data area element characteristic	4	4
Maximum list element characteristic	4	4
Maximum lock table entry characteristic	5	5