IBM Z

SNMP Application Programming Interfaces

SB10-7171-06

Note:

Before you use this information and the product it supports, read the information in "Safety" on page vii, Appendix G, "Notices," on page 219, and *IBM Systems Environmental Notices and User Guide*, Z125–5823.

This edition, SB10-7171-06, applies to the IBM Z and IBM LinuxONE servers. This edition replaces *SNMP Application Programming Interfaces*, SB10-7171-05.

There might be a newer version of this document in a **PDF** file available on **Resource Link**. Go to http://www.ibm.com/servers/resourcelink and click **Library** on the navigation bar.

Contents

Safety ..... **vii**
Safety notices vii
World trade safety information vii
Laser safety information vii
Laser compliance vii

About this publication ..... **ix**
Accessibility ix
Accessibility features x
Keyboard navigation x
Consult assistive technologies x
IBM and accessibility x
Revisions x
How to send your comments x
Summary of SNMP API support updates x

Chapter 1. APIs objectives **1**

Chapter 2. Overview **3**

Chapter 3. Console application APIs **5**
Management APIs 5
Data exchange APIs 5
Commands API 23
Command arguments 26
Data exchange APIs and commands API structures and definitions 50
Constant definitions 50
Data exchange APIs SNMP target structure (HWMCA_SNMP_TARGET_T) 61
Data exchange APIs initialize structure (HWMCA_INITIALIZE_T) 62
Data exchange APIs datatype structure (HWMCA_DATATYPE_T) 63
Data exchange APIs variable binding structure (HWMCA_VARBIND_T) 63
Data exchange APIs event qualifier structure (HWMCA_EVENT_QUALIFIER_T) 63
Function prototypes 63
Data exchange APIs and commands API example 66

Chapter 4. Console application managed objects **75**
Console application object identifier conventions 75
prefix 75
attribute 76
group 76
object 77
Console application object 77
Console application name bindings 77
Console attributes 77
Console application commands 78
Console application notifications 78
Group 79
Group name bindings 79
Group attributes 79
Group commands 81

Group notifications....81
Defined CPC....81
Defined CPC name bindings....81
Defined CPC attributes....81
Defined CPC relationships....96
Defined CPC commands....96
Defined CPC notifications....97
CPC image....98
CPC image name bindings....99
CPC image attributes....99
CPC image relationships....125
CPC image commands....125
CPC image notifications....126
Coupling facility....127
Coupling facility name bindings....127
Coupling facility attributes....127
Coupling facility relationships....138
Coupling facility commands....138
Coupling facility notifications....138
Reset activation profile object....139
Reset activation profile name bindings....139
Reset activation profile attributes....140
Reset activation profile notifications....141
Image activation profile object....141
Image activation profile name bindings....141
Image activation profile attributes....141
Image activation profile notifications....167
Load activation profile object....167
Load activation profile name bindings....167
Load activation profile attributes....167
Load activation profile notifications....170
Group profile object....170
Group profile name bindings....170
Group profile attributes....170
Group profile notifications....173
LPAR capacity group object....173
LPAR capacity group name bindings....173
LPAR capacity group attributes....173
Capacity record object....175
Capacity record name bindings....175
Capacity record attributes....175

Chapter 5. Configuring for the data exchange APIs....179
Configuring the console for API....179
Configuring BCPii Security Controls (for consoles 2.14.0 or later)....180

Appendix A. Building an application....181

Appendix B. HWMCA_EVENT_COMMAND_RESPONSE return codes....183

Appendix C. API return codes....189
Data exchange API call return codes....189
Command API call return codes....192
HWMCA_EVENT_COMMAND_RESPONSE return codes....195
Data exchange and command API (REXX version) return codes....200

Appendix D. APIs for Java (com.ibm.hwmca.api)....201

Appendix E. Object Attribute Availability.....203

Appendix F. XML descriptions..... 209
Add capacity command.....209
Remove capacity command.....209
Capacity record query..... 210
Engineering Change (EC)/Microcode Level (MCL) query..... 211
STP configuration information.....212
XML schema.....212

Appendix G. Notices..... 219
Trademarks.....219
Class A Notices..... 220

Safety

Safety notices

Safety notices may be printed throughout this guide. **DANGER** notices warn you of conditions or procedures that can result in death or severe personal injury. **CAUTION** notices warn you of conditions or procedures that can cause personal injury that is neither lethal nor extremely hazardous. **Attention** notices warn you of conditions or procedures that can cause damage to machines, equipment, or programs.

There are no **DANGER** notices in this guide.

World trade safety information

Several countries require the safety information contained in product publications to be presented in their translation. If this requirement applies to your country, a safety information booklet is included in the publications package shipped with the product. The booklet contains the translated safety information with references to the US English source. Before using a US English publication to install, operate, or service this product, you must first become familiar with the related safety information in the *Systems Safety Notices*, G229-9054. You should also refer to the booklet any time you do not clearly understand any safety information in the US English publications.

Laser safety information

All IBM Z® (Z) and IBM® LinuxONE (LinuxONE) models can use I/O cards such as FICON®, Open Systems Adapter (OSA), InterSystem Channel-3 (ISC-3), RoCE Express, Integrated Coupling Adapter (ICA SR), zHyperLink Express, or other I/O features which are fiber optic based and utilize lasers (short wavelength or long wavelength lasers).

Laser compliance

All lasers are certified in the US to conform to the requirements of DHHS 21 CFR Subchapter J for Class 1 or Class 1M laser products. Outside the US, they are certified to be in compliance with IEC 60825 as a Class 1 or Class 1M laser product. Consult the label on each part for laser certification numbers and approval information.

Laser Notice: U.S. FDA CDRH NOTICE if low power lasers are utilized, integrated, or offered with end product systems as applicable. Complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

CAUTION: Data processing environments can contain equipment transmitting on system links with laser modules that operate at greater than Class 1 power levels. For this reason, never look into the end of an optical fiber cable or open receptacle. (C027)

CAUTION: This product contains a Class 1M laser. Do not view directly with optical instruments. (C028)

About this publication

This document is intended to assist system management independent software vendors, customers, and system programmers in developing system management applications that provide integrated hardware and software system management solutions using the Console programming interfaces. A knowledge of the console and the C and/or Rexx language is recommended.

Note: Throughout this book, the term "Console" refers to the Hardware Management Console or the Support Element.

The Console is a direct-manipulation object-oriented graphical user interface that provides single point of control and single-system image for hardware elements. The Console provides the customer grouping support, aggregated and individual real-time system status by colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a customer group of systems. Also, the Console is exception based through customizable acceptable statuses per object. The objects the Console currently manages are:

- Central Processing Complexes (CPCs)
- Central Processing Complex Processor Resource/Systems Manager (PR/SM) partitions and/or native mode images (CPC Images)
- Central Processing Complex Coupling Facilities (Coupling Facility CPC Images)
- Customer defined groups of Central Processing Complexes, PR/SM partitions, native mode images, and/or Coupling Facilities.

In addition to providing an end user with the ability to view and manipulate managed objects, the Console also provides **management application programming interfaces (APIs)**. The management APIs provide the ability to get/set the attributes of a Console managed object, issue commands to be performed on a managed object from a local or remote application, receive asynchronous notifications, and generate Simple Network Management Protocol enterprise-specific traps.

In the following pages, the Console programming interfaces are detailed. The four areas to be covered are:

- Console APIs objectives
- Overview of the Console APIs architecture
- Console APIs definition, data structures, and usage
- Console managed object definitions and identifications.

Figures included in this document illustrate concepts and are not necessarily accurate in content, appearance, or specific behavior.

Accessibility

Accessible publications for this product are offered in EPUB format and can be downloaded from Resource Link® at http://www.ibm.com/servers/resourcelink.

If you experience any difficulty with the accessibility of any IBM Z and IBM LinuxONE information, go to Resource Link at http://www.ibm.com/servers/resourcelink and click **Feedback** from the navigation bar on the left. In the **Comments** input area, state your question or comment, the publication title and number, choose **General comment** as the category and click **Submit**. You can also send an email to reslink@us.ibm.com providing the same information.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Accessibility features

The following list includes the major accessibility features in IBM Z and IBM LinuxONE documentation, and on the Hardware Management Console and Support Element console:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Customizable display attributes such as color, contrast, and font size
- Communication of information independent of color
- Interfaces commonly used by screen magnifiers
- Interfaces that are free of flashing lights that could induce seizures due to photo-sensitivity.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

Consult assistive technologies

Assistive technology products such as screen readers function with our publications, the Hardware Management Console, and the Support Element console. Consult the product information for the specific assistive technology product that is used to access the EPUB format publication or console.

IBM and accessibility

See http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

Revisions

A technical change from the previous edition of this document is indicated by a thick vertical line to the left of the change.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. Send your comments by using Resource Link at http://www.ibm.com/servers/resourcelink. Click **Feedback** on the navigation bar on the left. You can also send an email to reslink@us.ibm.com. Be sure to include the name of the book, the form number of the book, the version of the book, if applicable, and the specific location of the text you are commenting on (for example, a page number, table number, or a heading).

Summary of SNMP API support updates

The following table describes updates to the SNMP API support, in chronological order by Console version. To locate the version level installed on your Console, look at the title bar on the workplace window.

Table 1. Summary of SNMP API support updates

Update	Description	Available since console:
Shutdown/Restart command support	“Commands API” on page 23 describes how to use the Commands API to shutdown/restart the Console.	2.9.0

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
On/Off Capacity on Demand (On/Off CoD) support	On/Off Capacity on Demand support provides the ability to activate, undo, or query information about a On/Off CoD record for a Defined CPC. “Commands API” on page 23 describes how to use the Commands API to perform an Activation or Undo of an On/Off CoD record for a Defined CPC, while “Defined CPC” on page 81 describes the On/Off CoD related attributes for the Defined CPC object. Important planning information for On/Off CoD API activation can be found in the *Capacity on Demand User's Guide* (available on **Resource Link** at http://www.ibm.com/servers/resourcelink).	2.9.1
Integrated Facility for Applications and z Integrated Information Processors weight support	Support for the processing weight value and processing weight capped attributes for Integrated Facility for Applications (IFA) processors was added to the CPC Image and Image Activation Profile objects on all Consoles version 2.9.0 or later. Support for the processing weight value and processing weight capped attributes for z Integrated Information Processors (zIIP) was added to the CPC Image and Image Activation Profile objects on all Consoles version 2.9.1 or later.	2.9.0 / 2.9.1
Processor running time support	Support for the processor running attributes was added to the Defined CPC and Reset Activation Profile objects.	2.9.1
Group profile support	Group Profile Object, in Chapter 4, “Console application managed objects,” on page 75, describes the new support for the Group Profile managed object. An additional attribute used to determine the list of Group Profile objects has also been added to the Defined CPC object as well.	2.9.2

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
Additional image activation profile attributes	Support for the following attributes was added to the Image Activation Profile objects on all Consoles version 2.9.2 or later: • Load at activation • Central storage • Reserved central storage • Expanded storage • Reserved expanded storage • Number of dedicated general-purpose processors • Number of reserved dedicated general-purpose processors • Number of dedicated Integrated Facility for Applications (IFA) processors • Number of reserved dedicated Integrated Facility for Applications (IFA) processors • Number of dedicated Integrated Facility for Linux® (IFL) processors • Number of reserved dedicated Integrated Facility for Linux (IFL) processors • Number of dedicated Internal Coupling Facility (ICF) processors • Number of reserved dedicated Internal Coupling Facility (ICF) processors • Number of dedicated z Integrated Information Processors (zIIP) processors • Number of reserved dedicated z Integrated Information Processors (zIIP) processors • Number of shared general-purpose processors • Number of reserved shared general-purpose processors • Number of shared Integrated Facility for Applications (IFA) processors • Number of reserved shared Integrated Facility for Applications (IFA) processors • Number of shared Integrated Facility for Linux (IFL) processors • Number of reserved shared Integrated Facility for Linux (IFL) processors • Number of shared Internal Coupling Facility (ICF) processors • Number of reserved shared Internal Coupling Facility (ICF) processors • Number of shared z Integrated Information Processors (zIIP) processors • Number of reserved shared z Integrated Information Processors (zIIP) processors	2.9.2

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
HwmcaGetBulk API	"HwmcaGetBulk" on page 11 describes the new HwmcaGetBulk application programming interface. This new API allows the application program to use the SNMP GetBulk request, which provides a mechanism for getting multiple attributes with a single request.	2.9.2
SNMP over TCP support	Prior to version 2.9.2, the Data Exchange APIs exclusively used the User Datagram Protocol (UDP) of TCP/IP for the sending of SNMP requests and the receiving of SNMP responses. Consoles version 2.9.2 or later now have support for flowing SNMP requests/responses using the Transmission Control Protocol (TCP) of TCP/IP. Since TCP guarantees reliable delivery, the Data Exchange APIs will automatically attempt to use the TCP protocol first and then fall back to UDP if it is unavailable. Support for using TCP for SNMP is also being made available for earlier Console versions as well. Contact your IBM support representative for details on what microcode levels are needed for this support.	2.9.2
Version support	Support for a new version attribute has been added to the Defined CPC and Console Application objects.	2.10.0
Engineering Change (EC)/ Microcode Level (MCL) support	Support for a new attribute that describes the Engineering Change and Microcode levels has been added to the Defined CPC and Console Application objects.	2.10.0
Internet Protocol (IP) addresses support	Support for a new attribute that describes all of the internal protocol (IP) addresses being used has been added to the Defined CPC and Console Application objects.	2.10.0
z/VM® IML/partition activation mode	The IML/Partition Activation mode attribute for CPC Image object supports a new value for when a CPC Image is activated in this newly supported mode.	2.10.0
Disabled wait event support	"HWMCA_EVENT_DISABLED_WAIT" on page 21 describes the data provided in the newly supported HWMCA_EVENT_DISABLED_WAIT event.	2.10.0
No command response event support	"HwmcaWaitEvent" on page 14 describes the capabilities available for the receipt of asynchronous event notifications. While command response event notifications are provided by all levels of Consoles, not all Consoles provide support for the new event mask, HWMCA_EVENT_NO_COMMAND_RESPONSE, which is used to indicate the registering application does not want to receive HWMCA_EVENT_COMMAND_RESPONSE events.	2.10.0

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
Temporary capacity support	New support in the form of a new object, new attributes, and new events has been added for temporary capacity support for Defined CPC objects. Capacity Record Object, in Chapter 4, “Console application managed objects,” on page 75 describes the new Capacity Record object and the object's associated attributes. Two new commands, HWMCA_ADD_CAPACITY_COMMAND and HWMCA_REMOVE_CAPACITY_COMMAND are also provided to allow for the addition and removal of temporary capacity for Defined CPC objects. Lastly, two new events are defined, HWMCA_EVENT_CAPACITY_CHANGE and HWMCA_EVENT_CAPACITY_RECORD_CHANGE, to allow for registered applications to be notified about temporary capacity changes for Defined CPC objects, as well as changes in Capacity Record objects.	2.10.0
IPv6 support	Support was added for Internet Protocol Version 6 (IPv6). To take advantage of this new support, new versions of the build and run-time files are available for platforms that also support IPv6.	2.10.0
Additional data added to HWMCA_EVENT_DATA event	Additional event data is now provided in “HWMCA_EVENT_ENDED” on page 18 which consists of: • the reason the console was ended, • the name of the Console application component that caused the Console to end, and • the type of shutdown that caused the Console to end.	2.10.0
Integrated Facility for Applications (IFA) are Application Assist Processor (AAP) in newer consoles	Integrated Facility for Applications (IFA) processors are now called Application Assist Processor (AAP) processors.	2.10.0
Additional image activation profile attributes	Support for the following CPU counter and CPU sampling related attributes were added to the Image Activation Profile objects: • Basic CPU counter authorization control • Problem state CPU counter authorization control • Crypto activity CPU counter authorization control • Extended CPU counter authorization control • Coprocessor group CPU counter authorization control • Basic CPU sampling authorization control	2.10.1
IPL Token attribute for CPC Image object	Support for the IPL token attribute was added to the CPC Image object on all Consoles version 2.10.1 or later. The returned value for the IPL token was not correct on Hardware Management Consoles before version 2.14.0.	2.10.1 / 2.14.0

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
Server Time Protocol (STP) configuration support	Support for a new attribute that describes the STP configuration has been added to the Defined CPC object. Also, the following STP commands were added to the Defined CPC object: • Swap Current Time Server • Set STP Configuration • Change STP-only CTN • Join STP-only CTN • Leave STP-only CTN	2.10.1
Additional temporary capacity support	Prior to version 2.10.1, only the total number of processors pending activation could be queried via the Data Exchange APIs. Support has been added to be able to query the number of processors pending activation by type as well.	2.10.1
Additional image activation profile attributes	Support for the following crypto related attributes were added to the Image Activation Profile objects: • Permit DEA key import functions • Permit AES key import functions	2.10.2
Group Profile capacity support	Support for a new attribute that provides the current capacity value for a group profile has been added to the Image object.	2.11.0
Alternate subchannel IPL	Specifying an alternate subchannel IPL address to the Load command is supported.	2.11.1
Absolute capping	Absolute capping is supported.	2.12.1
IBM zEnterprise® Application Assist Processor (zAAP) support	Support for IBM zEnterprise Application Assist Processor (zAAP) processors has been dropped for the 2.13.0 console. It is still available in consoles version 2.10.0 through 2.12.1.	2.13.0
HwmcaEnhancedGet and HwmcaEnhancedSet APIs	“HwmcaEnhancedGet” on page 12 and “HwmcaEnhancedSet” on page 14 describe the new HwmcaEnhancedGet and HwmcaEnhancedSet application programming interfaces. These new APIs allow the application program to get/set multiple attributes with a single request, as well as supporting the new category based attributes. While this API is introduced with version 2.13.1, most earlier versions of Consoles already support this new request.	2.13.1
Server Time Protocol (STP) daylight savings configuration support	Support has been added to the Defined CPC that sets the Daylight Savings Time setting in an STP-only Coordinated Timing Network (CTN).	2.13.1
Group absolute capping	Support has been added for getting/setting the group absolute capping settings for a CPC Image object and a group profile object.	2.13.1
Storage support for CPC Image object	Support has been added to allow for the querying of storage related information for the Defined CPC and CPC Image objects.	2.14.0

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
IBM Virtual Flash Memory support	Support has been added for the CPC Image and Image Activation Profile objects for IBM Virtual Flash Memory (VFM).	2.14.0
Removal of support for End Time Slice	Support has been removed for the "End timeslice if CPC image enters a wait state" attribute of the Defined CPC and Reset Activation Profile objects starting via Consoles version 2.14.0 or later. These attributes are still supported for Defined CPCs at a version earlier than 2.14.0.	2.14.0
IBM Secure Service Container Image Profile Support	Support has been added for the Image Activation Profile object for IBM Secure Service Container configuration settings The IBM Secure Service Container (SSC) was formerly known as the IBM z Appliance Container Infrastructure (zACI). Depending on the EC MCLs that have been applied to your system, the IBM zAware partition has been renamed to the IBM Secure Service Container (SSC) or z Appliance Container Infrastructure (zACI), and the IBM zAware firmware now runs in one of these partitions.	2.14.0
Base Control Program internal interface (BCPii) security controls	It has been possible to send API requests to the Support Element console using the Base Control Program internal interface (BCPii) for quite some time. Additional controls are now available for configuring additional security for these types of request. A summary of these security controls can be found in “Configuring BCPii Security Controls (for consoles 2.14.0 or later)” on page 180.	2.14.0
Additional image activation profile attributes	Support for the following attributes was added to the Image Activation Profile objects on all Consoles version 2.14.0 or later: • Number of dedicated Container Based Processors (CBP) • Number of reserved dedicated Container Based Processors (CBP) • Number of shared Container Based Processors (CBP) • Number of reserved shared Container Based Processors (CBP) • Initial Container Based Processor processing weight • Initial Container Based Processor processing weight capped • Minimum Container Based Processor processing weight • Minimum Container Based Processor processing weight capped • Maximum Container Based Processor processing weight • Maximum Container Based Processor processing weight capped • Current Container Based Processor processing weight • Current Container Based Processor processing weight capped • Container Based Processor absolute capping type • Container Based Processor absolute capping value	2.14.0

Table 1. Summary of SNMP API support updates (continued)		
Update	**Description**	**Available since console:**
Additional CPC image attributes	Support for the following attributes was added to the CPC Image objects on all Consoles version 2.14.0 or later: • Initial Container Based Processor processing weight • Initial Container Based Processor processing weight capped • Minimum Container Based Processor processing weight • Minimum Container Based Processor processing weight capped • Maximum Container Based Processor processing weight • Maximum Container Based Processor processing weight capped • Current Container Based Processor processing weight • Current Container Based Processor processing weight capped • Container Based Processor absolute capping type • Container Based Processor absolute capping value	2.14.0
New LPAR capacity group object	Support for a new LPAR capacity group object to allow for query and updates of the LPAR capacity group run time values. In addition, a new members attribute is added to the Group profile object.	2.14.1
Log event support	“HWMCA_EVENT_LOG_EVENT” on page 21 describes the data provided in the newly supported HWMCA_EVENT_LOG_EVENT event.	2.15.0
Sub-capacity boost support	Operating systems can opt into this support to boost the GP speed during the IPL window. Any Z operating system that runs on GPs could choose to exploit this.	2.15.0
zIIP-capacity boost support	The zIIP capacity boost provides additional capacity to z/OS LPARs during periods of service restoration from both planned and unplanned outages at no additional software cost.	2.15.0
Secure execution support	Secure execution protects secure guest data from other guests, hypervisors, and machine administrators.	2.15.0
Software signature support	Software signature support ensures secure boot for the load task and for Load and Image profiles.	2.15.0
ECC key	The ECC key is a new bit that controls the enablement of digital signatures (KDSA instruction). HMC/SE UI and API users are able to set or unset this bit, called the "ECC Key" bit in partitions.	2.15.0
NVMe Load and NVMe Dump support	“Commands API” on page 23 describes how to use the Commands API to perform an NVMe Load and an NVMe Dump.	2.15.0

Chapter 1. APIs objectives

The purpose of the Console application programming interfaces is to provide an open set of interfaces and a workstation platform for system management application providers. The interfaces provide the capability to use object-based industry-standard programming interfaces instead of building home-grown release specific programs for collecting the hardware information needed to provide an integrated hardware and software system management solution. Figure 1 on page 1 illustrates the integration of system management applications using the Console application open programming interfaces to provide a single-system image (SSI) and a single point of control (SPOC).

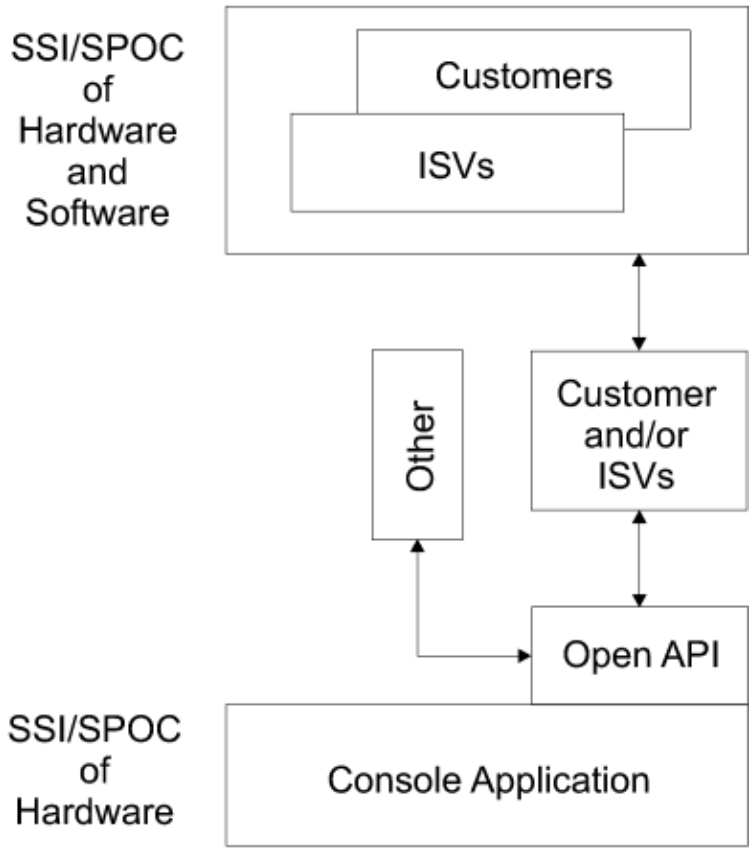


Figure 1. Console APIs Objectives

Chapter 2. Overview

This chapter contains a high-level diagram that illustrates how the Console application accomplishes the purpose of the application programming interfaces, shown in Figure 1 on page 1.

Figure 2 on page 3 shows a high-level architecture and flow of information for the Console application management programming interfaces. The Console application APIs are implemented using the Simple Network Management Protocol (SNMP) agent. The objects managed by the Console application described in Chapter 4, "Console application managed objects," on page 75 are stored in the Simple Network Management Protocol management information base (MIB). For more information about using the management application programming interfaces, see "Management APIs" on page 5.

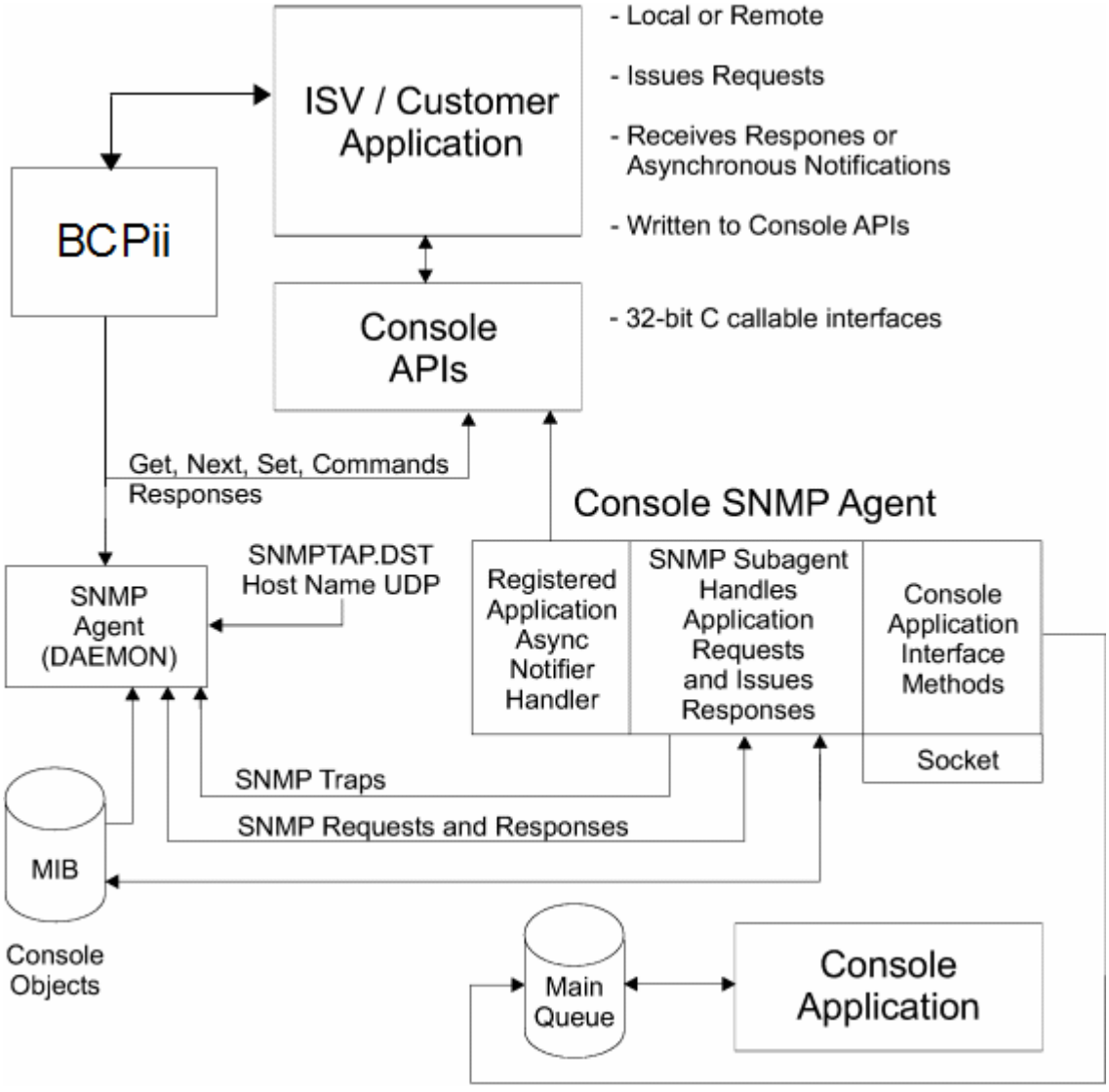


Figure 2. Console Application Data Exchange and Commands APIs

Note: It is important to note that while many simultaneous requests can be sent to the console, the SNMP Agent services these requests one at a time in the order in which they are received. The ISVs/Customer applications will have to handle the cases where the request execution might be delayed due to the other requests and implement an appropriate timeout value.

Chapter 3. Console application APIs

Management APIs

Data exchange APIs

The purpose of the Data Exchange APIs is to allow other applications, local or remote, the ability to exchange data related to the objects that the Console application manages. Specifically, this support allows other applications to request the Console application to:

- Query (Get/Get-Next) the attributes of objects,
- Change (Set) certain attributes of objects,
- Receive notification of significant events occurring to objects, and
- Generate enterprise-specific Simple Network Management Protocol traps for significant events occurring to objects.

The Data Exchange APIs use the Simple Network Management Protocol (SNMP) as the transport mechanism. The attributes of objects can be queried/changed through the underlying SNMP Set, Get, Get-Next requests, while event notification is accomplished through the use of the enterprise-specific SNMP Trap message.

Prior to version 2.9.2, the Data Exchange APIs exclusively used the User Datagram Protocol (UDP) of TCP/IP for the sending of SNMP requests and the receiving of SNMP responses. Consoles version 2.9.2 or later now have support for flowing SNMP requests/responses using the Transmission Control Protocol (TCP) of TCP/IP. Since TCP guarantees reliable delivery, the Data Exchange APIs automatically attempt to use the TCP protocol first and then fall back to UDP if it is unavailable.

The underlying SNMP protocol is encapsulated in several APIs in order to reduce the complexities for the application programmer. Specifically, the set of Data Exchange APIs consists of:

HwmcaInitialize
: Used to perform some initialization tasks necessary for the remainder of the Data Exchange APIs set and the Commands API.

HwmcaGet
: Used to perform a query or Get request for a specified object or object attribute.

HwmcaGetNext
: Used to perform a query-next or Get-next request for an object or object attributes that occurs next in the lexical sequence of objects managed by the Console application.

HwmcaGetBulk
: Used to minimize the number of requests required to retrieve large amounts of object or object attribute data in a manner similar to what could be obtained with a series of HwmcaGetNext calls.

HwmcaEnhancedGet
: Used to perform a query or Get request for a set of objects or object attributes, as well as for category based attributes.

HwmcaSet
: Used to perform a change or Set request for a specified object or object attribute.

HwmcaEnhancedSet
: Used to perform a change or Set request for a set of objects or object attributes.

HwmcaWaitEvent
: Used to wait for a specified period (or forever) for an event notification from the Console application.

HwmcaTerminate
Used to perform any cleanup tasks required by any of the other APIs in the set.

HwmcaBuildId
A convenience routine that can be used to construct an object identifier for any object supported by the Console application.

HwmcaBuildAttributeId
A convenience routine that can be used to construct an attribute object identifier for any object supported by the Console applications, based on the object identifier of the object itself.

Note: It is possible that some of these APIs might encounter problems if the Console that they are targeting has been configured to use the *Lockup/Screen saver mode* capability. It is recommended that Consoles used as targets for these APIs not have this feature of OS/2 enabled.

The following pages describe each of these APIs in greater detail.

HwmcaInitialize

Use this API to perform any initialization tasks required in order for the remainder of the API set to function correctly. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to an **HWMCA_INITIALIZE_T** structure. This structure defines all the information that is required for the Console application to perform the initialization request. The fields of the **HWMCA_INITIALIZE_T** structure meaningful are:

pTarget
A pointer to data specifying the target Console application for the initialization request.

This is a pointer to an **HWMCA_SNMP_TARGET_T** structure. The fields of this structure are:

pHost
A pointer to a null terminated string specifying the host name or internet address for the target Console application.

szCommunity
A null terminated string specifying the community name that is to be used for the SNMP request made to the target Console application. (Refer to Chapter 5, "Configuring for the data exchange APIs," on page 179 for more information regarding the community name used in SNMP requests.)

ulSecurityVersion
Used to specify the desired authentication method. Use the value **HWMCA_SECURITY_VERSION2** for community name based SNMPv2c authentication. Use the value **HWMCA_SECURITY_VERSION3** for username and password based SNMPv3 authentication.

szUsername
Username to be used for SNMPv3 authentication.

szPassword
Password to be used for SNMPv3 authentication.

ulEventMask
Used to specify the types of event notifications that the application program would like to be registered for. Any combination of the **HWMCA_EVENT_*** constants logically ORed together can be specified. This event mask is used for all events emitted by Console applications managed objects, such as:

- HWMCA_EVENT_COMMAND_RESPONSE
- HWMCA_EVENT_MESSAGE
- HWMCA_EVENT_STATUS_CHANGE
- HWMCA_EVENT_NAME_CHANGE

- HWMCA_EVENT_ACTIVATE_PROF_CHANGE
- HWMCA_EVENT_CREATED
- HWMCA_EVENT_DESTROYED
- HWMCA_EVENT_EXCEPTION_STATE
- HWMCA_EVENT_ENDED
- HWMCA_EVENT_HARDWARE_MESSAGE
- HWMCA_EVENT_OPSYS_MESSAGE
- HWMCA_EVENT_NO_REFRESH_MESSAGE
- HWMCA_EVENT_STARTED
- HWMCA_EVENT_HARDWARE_MESSAGE_ DELETE
- HWMCA_EVENT_SECURITY_EVENT
- HWMCA_EVENT_CAPACITY_CHANGE
- HWMCA_EVENT_CAPACITY_RECORD_ CHANGE
- HWMCA_EVENT_DISABLED_WAIT
- HWMCA_EVENT_LOG_EVENT

These event notifications are sent to all registered applications, independent of whether an application originated the request.

In addition to specifying the types of events that the application program wants to be registered for, this field can also be used to specify some additional options for the Data Exchange APIs. These additional options are:

- HWMCA_DIRECT_INITIALIZE

  By default, the Data Exchange APIs and the Commands API use SNMP when performing the HwmcaInitialize. This flag can be specified to instruct the HwmcaInitialize call to use a proprietary TCP/IP sockets level protocol to perform the HwmcaInitialize, rather than using the SNMP protocol. When this flag is specified it is possible for the HwmcaInitialize to be successful when using a community name that has read only address. When this flag is not used it is required that the community name used for the HwmcaInitialize call has read/write access.

  Note: Specifying this flag is highly recommended when a firewall exists between the Console and the API application. This is because the socket used for the HwmcaInitialize call is also used to send event to the API application. Since this socket connection targets a specific port on the Console (port 3161), it is very straight forward to define a rule in the firewall that allows connections to this port on the Console. If this flag is not specified, the Console attempts to establish a socket connection to a socket created when the API application called the HwmcaInitialize routine. Since the port number for this socket is not fixed, it is very difficult to define a firewall rule to allow this connection from the Console back to the API application.
- HWMCA_FORCE_CLIENT_PATH

  When using the Data Exchange APIs to target a Console with multiple LAN interfaces (for example, a token ring and ethernet interface), this flag can be used to instruct the Console to ensure that all Data Exchange APIs and the Commands API use the targeted internet address when sending and receiving data.
- HWMCA_SNMP_VERSION_2

  By default, the Data Exchange APIs and the Commands API use SNMP version 1. By specifying this flag, the Data Exchange APIs are instructed to used SNMP version 2 as the underlying protocol. The major reason a Data Exchange APIs application would specify this, is so that it can receive more detailed error return codes that are provided by SNMP version 2.
- HWMCA_TOLERATE_LOST_EVENTS

  By default, the HwmcaWaitEvent call terminates the connection to the target console if the API application is unable to process events as fast or faster than the target console is able to send

them. By specifying this event mask flag, the connection will not be terminated in this case. Instead, events will not be sent to the API application while it is unable to receive them.

- HWMCA_QUALIFIER_SPECIFIED

  By default event notifications from all Console application managed objects that match the event masks specified in this field will be sent to the API application. By specifying this event mask flag, additional qualification information can be provided to further limit the event notifications that will be sent to the API application. When this event mask flag is specified, the calling API application should also provide additional qualification information in the *ulReserved* field. Refer to the description of the *ulReserved* field for details on how this additional qualification information is specified.

- HWMCA_EVENT_NO_COMMAND_RESPONSE

  By default, all HWMCA_EVENT_COMMAND_RESPONSE events are sent to each registered application. This event mask flag can be used to indicate that the registering application does not want to receive these events.

- HWMCA_EVENT_KEEP_ALIVE

  By default, there is no network traffic sent from the target console to the API application unless there is an event to be delivered that the application is interested in. In some configurations involving firewalls this can result in the firewall dropping the underlying communication path without either the console or the API application realizing it, resulting in the potential loss of events being delivered to the application. This event mask flag can be used to enable keep alive events to be sent from the console to the API application to prevent the underlying communications path from being dropped by the firewall. The default frequency for the sending of these keep alive events is every 5 minutes if no other event has been sent. This frequency can be controlled by specifying an event qualifier for this event type to number of seconds for the desired frequency.

Note: Care should be used when trying to use the same **HWMCA_INITIALIZE_T** structure for HwmcaWaitEvent calls in addition to the rest of the APIs in the set. Events associated with a **HWMCA_INITIALIZE_T** structure will be queued until retrieved with the HwmcaWaitEvent or until another API, such as HwmcaGet, is called. Therefore, making calls, such as HwmcaGet, will cause any queued events to be discarded and lost.

When both HwmcaWaitEvent and other calls need to be made, an application should perform two HwmcaInitialize calls using two distinct **HWMCA_INITIALIZE_T** structures. The application can then use one of the **HWMCA_INITIALIZE_T** structures for only HwmcaWaitEvent calls and the other **HWMCA_INITIALIZE_T** structure for the other API calls.

ulReserved
This is a reserved field and must be set to zero for the Data Exchange APIs if the **HWMCA_QUALIFIER_SPECIFIED** event mask flag is not specified. If the **HWMCA_QUALIFIER_SPECIFIED** event mask flag is specified, then this field should contain a pointer to an **HWMCA_EVENT_QUALIFIER_T** structure, which is the first of a linked list of additional event qualification information. The fields of the **HWMCA_EVENT_QUALIFIER_T** structures in the list are:

ulEventMask
This field should be set to the event mask flag that is being qualified. Only one event mask flag should be specified in this field. For example, **HWMCA_EVENT_OPSYS_MESSAGE** should be specified when qualifying operating system message event notifications.

ulType
This field is used to indicate the type of event qualification information being provided. The following event qualification types are currently supported.

HWMCA_QUALIFIER_TYPE_NAME
This value is used to indicate that the event qualification data is the null terminated name of the managed object, which is specified in the *type.szName* field of this structure. An **HWMCA_EVENT_QUALIFIER_T** structure that specifies this event qualification type can

be used to limit event notifications for the specified event mask to those associated with a managed object with the specified name.

pNext
A pointer to the next **HWMCA_EVENT_QUALIFIER_T** structure. A **NULL** is used to indicate that there are no more structures in the linked list.

Once the **HWMCA_INITIALIZE_T** is used on a successful *HwmcaInitialize*, this field should not be altered in any way.

The remainder of the **HWMCA_INITIALIZE_T** structure should be left alone and will be filled in by the *HwmcaInitialize* API. It is important that this structure be left intact and accessible, since it must be passed as a parameter on each of the remaining Data Exchange APIs and Commands API.

In addition to using the **HWMCA_INITIALIZE_T** for any subsequent Data Exchange APIs, it can also be reused on another *HwmcaInitialize* call. The only field that can be changed when doing this is the *ulEventMask* field. By changing this value, an application can change the events notifications that it is registered to receive.

Refer to "Data exchange APIs initialize structure (HWMCA_INITIALIZE_T)" on page 62 for the C declaration of this structure.

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for the *HwmcaInitialize* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaInitialize* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the initialization request was successfully delivered and processed by the Hardware Management Console Application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

Note: Upon successful completion of the HwmcaInitialize call, the ulEventMask field of the HWMCA_INITIALIZE_T can be checked for the HWMCA_SNMP_USING_TCP flag to determine if the initialized session is using UDP or TCP for the flow of SNMP data.

HwmcaRegister

Use this API to alter the event mask and/or event qualifiers used on a previous HwmcaInitialize call. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

ulEventMask
Used to specify the new types of event notifications that the application program would like to be registered for. Any combination of the **HWMCA_EVENT_*** constants logically ORed together can be specified.

pQualifiers
If the **HWMCA_QUALIFIER_SPECIFIED** event mask flag is specified, then this field should contain a pointer to an **HWMCA_EVENT_QUALIFIER_T** structure, which is the first of a linked list of additional event qualification information.

ulTimeout
Used to specify the amount of time that the calling application wants to wait for the *HwmcaRegister* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaRegister* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the register request was successfully delivered and processed by the Hardware Management Console Application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

Note: The event mask and event qualifiers specified on the *HwmcaRegister* call will completely replace those in effect from the previous *HwmcaRegister* call.

HwmcaGet

Used to retrieve or Get the data associated with a specific object attribute. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pszObjectID
A pointer to a null terminated object identifier string for which the data is to be retrieved. Refer to Chapter 4, "Console application managed objects," on page 75 for more information about the object identifiers that the Console application manages.

pOutput
A pointer to an output buffer for the data of the returned object.

uILength
The size of the output buffer specified by the *pOutput* argument.

pulBytesNeeded
A pointer to an unsigned long integer where the number of total bytes needed for this Get request is returned. If the returned value is greater than that specified in the *ulLength* argument, then the call should be made again, with a larger buffer in order to Get all the object data. If the buffer specified by *pOutput* is too small, then the retrieved object data should not be used, since it is incomplete.

ulTimeout
Used to specify the amount of time that the calling application wants to wait for the *HwmcaGet* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaGet* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the retrieve/Get request was successfully delivered and processed by the Console application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

Upon successful completion of the *HwmcaGet* API, the output buffer specified by *pOutput* is populated with a series of one or more **HWMCA_DATATYPE_T** structures along with their associated data. The fields of the **HWMCA_DATATYPE_T** structure are:

ucType
Defines the type of data represented by this **HWMCA_DATATYPE_T** structure. Possible values are:

HWMCA_TYPE_INTEGER
Represents a signed number value in host byte order.

HWMCA_TYPE_OCTETSTRING
Represents a null terminated string value.

HWMCA_TYPE_NULL
Used to denote that no value is present.

HWMCA_TYPE_IPADDRESS
Represents a 32- bit internet address in host byte order.

ulLength
Used to specify the length of the data represented by this **HWMCA_DATATYPE_T** structure.

pData
A pointer to the actual data that this **HWMCA_DATATYPE_T** structure represents.

pNext
A pointer to the next **HWMCA_DATATYPE_T** structure. A **NULL** is used to indicate that there are no more structures in the linked list.

Note: The value stored in the *pulBytesNeeded* field represents the total amount of data returned, while the *ulLength* field of each **HWMCA_DATATYPE_T** structure represents the length of each individual data element in the series.

HwmcaGetNext

Used to retrieve or Get the data associated with the object attribute that occurs next in the lexical sequence of objects, based on a specified object identifier. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.)

The arguments specified for this API are identical to those specified for the HwmcaGet API with two subtle differences.

1. The meaning of the *pszObjectID* argument is used as the base for the Get-Next operation, as opposed to having its object data retrieved.
2. Two **HWMCA_DATATYPE_T** structures and their associated data are returned. The first is the object identifier string for the object whose data is being returned and the second is for the data itself.

HwmcaGetBulk

Used to retrieve or Get data associated with a series of object attributes with a single request. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) This call can be viewed as performing a series of HwmcaGetNext calls with a single request. For additional details about the underlying SNMP GetBulkRequest used by this function refer to Request for Comments (RFC) 3416.

The arguments specified for this API are:

pInitialize
A pointer to the HWMCA_INITIALIZE_T structure that was used on the HwmcaInitialize API.

pszObjectIDs
A pointer to a linked list of HWMCA_DATATYPE_T structures used to specify the object identifiers to use for the GetBulk request. Refer to Chapter 4, "Console application managed objects," on page 75 for more information about the object identifiers that the Console application manages.

nonRepeaters
The number of object identifiers specified in the pszObjectIds argument that are to produce only one HWMCA_DATATYPE_T structure in the output buffer.

maxRepititions
The maximum number of HWMCA_DATATYPE_T fields to be placed in the output buffer for the remaining object identifiers specified in the pszObjectIDs argument.

pOutput
A pointer to an output buffer for the data of the returned object.

uILength
The size of the output buffer specified by the pOutput argument.

pulBytesNeeded
A pointer to an unsigned long integer where the number of total bytes needed for this GetBulk request is returned. If the returned value is greater than that specified in the ulLength argument, then the call should be made again, with a larger buffer in order to get the complete set of object data. If the buffer specified by pOutput is too small, then the retrieved object data should not be used, since it is incomplete.

ulTimeout
Used to specify the amount of time that the calling application wants to wait for the HwmcaGetBulk to complete. This value is specified in milliseconds and the value of HWMCA_INFINITE_WAIT can be used to cause the application to wait forever.

The HwmcaGetBulk API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the request was successfully delivered and processed by the Console application. A value of HWMCA_DE_NO_ERROR indicates successful completion. Upon successful completion of the HwmcaGetBulk API, the output buffer specified by pOutput is populated

with a series of one or more HWMCA_DATATYPE_T structures along with their associated data. The fields of the HWMCA_DATATYPE_T structure are:

ucType
Defines the type of data represented by this HWMCA_DATATYPE_T structure. Possible values are:

HWMCA_TYPE_INTEGER
Represents a signed number value in host byte order.

HWMCA_TYPE_OCTETSTRING
Represents a null terminated string value.

HWMCA_TYPE_NULL
Used to denote that no value is present.

HWMCA_TYPE_IPADDRESS
Represents a 32- bit internet address in host byte order.

ulLength
Used to specify the length of the data represented by this HWMCA_DATATYPE_T structure.

pData
A pointer to the actual data that this HWMCA_DATATYPE_T structure represents.

pNext
A pointer to the next HWMCA_DATATYPE_T structure. A NULL is used to indicate that there are no more structures in the linked list.

Note: The value stored in the pulBytesNeeded field represents the total amount of data returned, while the ulLength field of each HWMCA_DATATYPE_T structure represents the length of each individual data element in the series.

HwmcaEnhancedGet

Used to retrieve or Get the data associated with multiple object attributes and/or category based attributes. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pVarbinds
A pointer to a linked list of **HWMCA_VARBIND_T** structures used to specify the object identifiers to use for the request.

pOutput
A pointer to an output buffer for the data of the returned object(s).

uILength
The size of the output buffer specified by the *pOutput* argument.

pulBytesNeeded
A pointer to an unsigned long integer where the number of total bytes needed for this request is returned. If the returned value is greater than that specified in the *ulLength* argument, then the call should be made again, with a larger buffer in order to Get all the object data. If the buffer specified by *pOutput* is too small, then the retrieved object data should not be used, since it is incomplete.

ulTimeout
Used to specify the amount of time that the calling application wants to wait for the *HwmcaEnhancedGet* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaEnhancedGet* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the retrieve/Get request was successfully delivered and processed by the Console application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

Upon successful completion of the *HwmcaEnhancedGet* API, the output buffer specified by *pOutput* is populated with a series of one or more **HWMCA_VARBIND_T** structures along with their associated data. The fields of the **HWMCA_VARBIND_T** structure are:

ucType
Defines the type of data represented by this **HWMCA_VARBIND_T** structure. Possible values are:

HWMCA_TYPE_INTEGER
Represents a signed number value in host byte order.

HWMCA_TYPE_OCTETSTRING
Represents a null terminated string value.

HWMCA_TYPE_NULL
Used to denote that no value is present.

HWMCA_TYPE_IPADDRESS
Represents a 32- bit internet address in host byte order.

status
The error status for this variable binding.

ulLength
Used to specify the length of the data represented by this **HWMCA_VARBIND_T** structure.

pData
A pointer to the actual data that this **HWMCA_VARBIND_T** structure represents.

pNext
A pointer to the next **HWMCA_VARBIND_T** structure. A **NULL** is used to indicate that there are no more structures in the linked list.

The number of returned **HWMCA_VARBIND_T** structures will not necessarily equal the number of input variable bindings specified. This can occur if one or more of the input variable bindings is for a category based attribute since multiple object identifiers are returned for each input category based attribute. If there was an error getting the data for one or more of the input variable bindings the return code will be **HWMCA_DE_PARTIAL_SUCCESS**. The output data will contain the entries for the failing object identifiers and the status field for those entries will contain the specific error code.

Note: The value stored in the *pulBytesNeeded* field represents the total amount of data returned, while the *ulLength* field of each **HWMCA_VARBIND_T** structure represents the length of each individual data element in the series.

HwmcaSet

Used to change or Set the data associated with a specific object attribute. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pszObjectID
A pointer to a null terminated object identifier string for which the data is to be changed or Set. Refer to Chapter 4, "Console application managed objects," on page 75 for more information about the object identifiers that the Console application manages.

pDataType
A pointer to an **HWMCA_DATATYPE_T** structure that specifies the data to be used for the Set request. The fields of the **HWMCA_DATATYPE_T** structure are:

ucType
Defines the type of data represented by this **HWMCA_DATATYPE_T** structure. Possible values are:

HWMCA_TYPE_INTEGER
Represents a signed number value in host byte order.

Note: The Data Exchange APIs currently only support lengths of 2 bytes or 4 bytes for the HWMCA_TYPE_INTEGER data type when using the HwmcaSet.

HWMCA_TYPE_OCTETSTRING
Represents a null terminated string value.

ulLength
Used to specify the length of the data represented by this **HWMCA_DATATYPE_T** structure.

pData
A pointer to the actual data that this **HWMCA_DATATYPE_T** structure represents.

pNext
This should be set to NULL for the *HwmcaSet* API and is ignored.

Refer to Chapter 4, "Console application managed objects," on page 75 for a description of the data types, data lengths, and valid data values of the data associated with each type of object managed by the Console application.

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for the *HwmcaSet* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaSet* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the change/Set request was successfully delivered and processed by the Console application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

HwmcaEnhancedSet

Used to update or Set the data associated with multiple object attributes. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pVarbinds
A pointer to a linked list of **HWMCA_VARBIND_T** structures used to specify the object identifiers and their associated data values to use for the request. Each **HWMCA_VARBIND_T** structure in the list specifies the object identifier and data for an individual update

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for the *HwmcaEnhancedGet* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaEnhancedSet* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the update/Set request was successfully delivered and processed by the Console application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion. Any other return value indicates a failure and the error status for the first failing variable binding will be populated with the value for the type of error encountered.

HwmcaWaitEvent

Used to wait for event notifications for objects managed by the Console application. The application specifies the types of events that it wants to receive through the use of the *ulEventMask* field of the **HWMCA_INITIALIZE_T** structure that is used on the *HwmcaInitialize* API. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pOutput
A pointer to an output buffer for the returned event notification data.

ulLength
The size of the output buffer specified by the *pOutput* argument.

pulBytesNeeded
A pointer to an unsigned long integer where the number of total bytes needed for this event notification is returned. If the returned value is greater than that specified in the *ulLength* argument, then the event notification data should not be used, since it is incomplete.

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for an event notification. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaWaitEvent* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if any errors occurred while waiting for the event notification. A value of **HWMCA_DE_NO_ERROR** indicates successful completion. A value of **HWMCA_DE_TIMEOUT** indicates that no event notifications were present in the specified timeout period.

Upon successful completion of the *HwmcaWaitEvent* API, the output buffer specified by *pOutput* is populated with a series of one or more **HWMCA_DATATYPE_T** structures along with their associated data. The fields of the **HWMCA_DATATYPE_T** structure are:

ucType
Defines the type of data represented by this **HWMCA_DATATYPE_T** structure. Possible values are:

HWMCA_TYPE_INTEGER
Represents a signed number value in host byte order.

HWMCA_TYPE_OCTETSTRING
Represents a null terminated string value.

HWMCA_TYPE_OBJECTID
Represents a null terminated object identifier string.

ulLength
Used to specify the length of the data represented by this **HWMCA_DATATYPE_T** structure.

pData
A pointer to the actual data that this **HWMCA_DATATYPE_T** structure represents.

pNext
A pointer to the next **HWMCA_DATATYPE_T** structure. A **NULL** is used to indicate that there are no more structures in the linked list.

Note: The value stored in the *pulBytesNeeded* field represents the total amount of data returned, while the *ulLength* field of each **HWMCA_DATATYPE_T** structure represents the length of each individual data element in the series.

The series of **HWMCA_DATATYPE_T** structures returned from the *HwmcaWaitEvent* API are used to specify:

- An HWMCA_TYPE_OBJECTID that specifies the object identifier of the object that the event notification pertains to
- An HWMCA_TYPE_INTEGER that specifies the event notification type for this event
- Any additional data for the event notification type, as specified below.

The additional data for each of the event notification types are:

HWMCA_EVENT_COMMAND_RESPONSE

Used to notify the application of completion information for a command that has been initiated through the use of the Commands API.

The additional data for this event consists of three object identifier/value pairs that describe the following:

1. An HWMCA_TYPE_OBJECTID that specifies the object identifier of the command for which this command response event has been generated.
2. An HWMCA_TYPE_INTEGER that specifies the return code value to be used to determine the success or failure of the command request that is associated with this command response event.

   Note: Refer to Appendix B, "HWMCA_EVENT_COMMAND_RESPONSE return codes," on page 183 for a list of possible values that can be returned.
3. An HWMCA_TYPE_INTEGER that specifies whether this is the last **HWMCA_EVENT_COMMAND_RESPONSE** event that will be issued for this command. A value of HWMCA_TRUE indicates this event as the last, while a value of HWMCA_FALSE indicates that more **HWMCA_EVENT_COMMAND_RESPONSE** events will be forthcoming.
4. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.
5. An HWMCA_TYPE_OCTETSTRING that specifies the command correlator.

   Note: This field will only be present if the command was invoked using the HwmcaCorrelatedCommand API call.

HWMCA_EVENT_MESSAGE

Used to notify the application that an object managed by the Console application or the Console application itself has a new or refreshed message. This event is generated only for the base objects and not for copies of objects within user-defined groups.

This event is returned to the application when any combination of the following values is used in the *ulEventMask* field of the **HWMCA_INITIALIZE_T** structure:

- HWMCA_EVENT_MESSAGE
- HWMCA_EVENT_HARDWARE_MESSAGE
- HWMCA_EVENT_OPSYS_MESSAGE

If the **HWMCA_EVENT_MESSAGE** value is specified in the *ulEventMask* field of the **HWMCA_INITIALIZE_T** structure, then the application will be notified of both hardware and operating system message events.

If only the **HWMCA_EVENT_HARDWARE_MESSAGE** or **HWMCA_EVENT_OPSYS_MESSAGE** value is specified in the *ulEventMask* field of the **HWMCA_INITIALIZE_T** structure, then the application will be notified only of hardware or operating system message events, respectively.

In addition, the **HWMCA_EVENT_NO_REFRESH_MESSAGE** value can be specified with the above values to control whether the application should be notified of **HWMCA_EVENT_MESSAGE** events for refreshed messages. If the **HWMCA_EVENT_NO_REFRESH_MESSAGE** value is specified in the *ulEventMask* field of the **HWMCA_INITIALIZE_T** structure, then the application will not be notified of **HWMCA_EVENT_MESSAGE** events for refreshed messages.

The additional data for this event can take on two different formats. The format being received can be determined through examining the first object identifier/value pair. The object identifier/value pairs for each of the two formats follows:

An HWMCA_TYPE_INTEGER that specifies whether the message is a hardware or operating system message (HWMCA_HARDWARE_MESSAGE or HWMCA_OPSYS_MESSAGE).

1. The remaining object identifier/value pair for hardware messages is:
   a. An HWMCA_TYPE_OCTETSTRING that specifies the new or refreshed hardware message text.
   b. An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).
   c. An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the new or refreshed hardware message.
   d. An HWMCA_TYPE_OCTETSTRING that specifies the names of the CPC Image object(s) associated with the object that generated the new or refreshed hardware message. This HWMCA_TYPE_OCTETSTRING is a null terminated, blank delimited list of the CPC Image name(s).

When receiving this event from a Support Element Console, this value contains the name(s) of the CPC Images that are running on the CPC that the Support Element Console is controlling.

When receiving this event from a Hardware Management Console, this value:

- Contains no CPC Image names for hardware messages for the Hardware Management Console itself
- Contains no CPC Image names for Optical Network related hardware messages
- Contains the name(s) of the CPC Images that are running on the CPC that the hardware message pertains to.

e. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

2. The remaining object identifier/value pairs for operating system messages are:
   a. An HWMCA_TYPE_OCTETSTRING that specifies the new or refreshed operating system message text.

      Note: If the operating system message text contains multiple lines, then each additional line is delimited from the next line with the character sequence of a carriage return (\r) and a new line (\n).
   b. An HWMCA_TYPE_OCTETSTRING that specifies the message identifier of the new operating system message.
   c. An HWMCA_TYPE_OCTETSTRING that specifies the date of the new operating system message or an HWMCA_TYPE_NULL indicating that there is no date value for this new operating system message.
   d. An HWMCA_TYPE_OCTETSTRING that specifies the time of the new operating system message or an HWMCA_TYPE_NULL indicating that there is no time value for this new operating system message.
   e. An HWMCA_TYPE_INTEGER that specifies whether the new operating system message should cause the alarm to be sounded (HWMCA_TRUE or HWMCA_FALSE).
   f. An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a priority message or not (HWMCA_TRUE or HWMCA_FALSE).
   g. An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a held message or not (HWMCA_TRUE or HWMCA_FALSE).
   h. An HWMCA_TYPE_OCTETSTRING that specifies the prompt text that should be associated with the new operating system message or an HWMCA_TYPE_NULL indicating that there is no prompt text for this new operating system message.
   i. An HWMCA_TYPE_OCTETSTRING that specifies the name of the operating system that generated this new operating system message or an HWMCA_TYPE_NULL indicating that there is no operating system name associated with this new operating system message.
   j. An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).
   k. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_STATUS_CHANGE

Used to notify the application that an object managed by the Console application has changed status. This event is generated only for the base objects and not for copies of objects within user-defined groups.

The additional data for this event consists of two object identifier/value pairs that describe the following:

1. An HWMCA_TYPE_INTEGER that specifies the new status value
2. An HWMCA_TYPE_INTEGER that specifies the old status value.
3. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_NAME_CHANGE

Used to notify the application that an object managed by the Console application has had a name change. This event notification can be useful when an application retains the object identifiers for objects it is interested in, since the name of an object is used to build the unique portion of the object identifier. This event is generated only for the base objects and not for copies of objects within user-defined groups.

The additional data for this event consists of two object identifier/value pairs that describe the following:

1. An HWMCA_TYPE_OCTETSTRING that specifies the new object name
2. An HWMCA_TYPE_OCTETSTRING that specifies the old object name.

HWMCA_EVENT_ACTIVATE_PROF_CHANGE

Used to notify the application that an object managed by the Console application has changed which activation profile is associated with it.

The additional data for this event consists of two object identifier/value pairs that describe the following:

1. An HWMCA_TYPE_OCTETSTRING that specifies the new activation profile name
2. An HWMCA_TYPE_OCTETSTRING that specifies the old activation profile name.
3. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_CREATED

Used to notify the application that a new object managed by the Console application has been defined or instantiated.

The additional data for this event consists of an object identifier/value pair for an HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_DESTROYED

Used to notify the application that an object managed by the Console application has been undefined.

The additional data for this event consists of an object identifier/value pair for an HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_EXCEPTION_STATE

Used to notify the application that an object managed by the Console application has either entered into or out of an exception state. An object is considered in an exception state when its status is not considered acceptable as defined by the acceptable status attribute of the object. This event is generated only for the base objects and not for copies of objects within user-defined groups.

The additional data for this event consists of two object identifier/value pairs that describe the following:

1. An HWMCA_TYPE_INTEGER that specifies whether the object is entering into an exception state (HWMCA_TRUE) or leaving an exception state (HWMCA_FALSE).
2. An HWMCA_TYPE_INTEGER that specifies the status value for the object.
3. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_STARTED

Used to notify the application that the Console application has started and is now ready to handle Data Exchange APIs and Commands API request.

The additional data for this event consists of an object identifier/value pair for an HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_ENDED

Used to notify the application that the Console application is ending.

The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_INTEGER that specifies the reason for the event. The possible values are:
   - HWMCA_ENDED_USER - the event was initiated by a user,
   - HWMCA_ENDED_AUTOMATION - the event was initiated by automation, or
   - HWMCA_ENDED_OTHER - the event was initiated by the Console application itself (for example, recovery action, change management, etc.)
2. An HWMCA_TYPE_OCTETSTRING that specifies the name of the Console application component that caused the event.
3. An HWMCA_TYPE_INTEGER that specifies the shutdown type for the event. The possible values are:
   - HWMCA_SHUTDOWN_CONSOLE - the console has been shut down and will take manual intervention to be restarted,
   - HWMCA_RESTART_APPLICATION - the console application has been stopped and will automatically be restarted, or
   - HWMCA_RESTART_CONSOLE - the console has been stopped and will automatically be restarted.
4. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

HWMCA_EVENT_HARDWARE_MESSAGE_DELETE

Used to notify the application that a hardware message associated with an object managed by the Console application or the Console application itself, has been deleted. This event is generated only for the base objects and not for copies of objects within user-defined groups.

The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_INTEGER that specifies that the message being deleted is a hardware message (HWMCA_HARDWARE_MESSAGE).
2. An HWMCA_TYPE_OCTETSTRING that specifies the message text for the hardware message being deleted.
3. An HWMCA_TYPE_INTEGER that is always set to HWMCA_FALSE for this event.
4. An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the hardware message being deleted.
5. An HWMCA_TYPE_OCTETSTRING that specifies the names of the CPC Image object(s) associated with the object for which the hardware message is being deleted. This HWMCA_TYPE_OCTETSTRING is a null terminated, blank delimited list of the CPC Image name(s).

   When receiving this event from a Support Element Console, this value contains the name(s) of the CPC Images that are running on the CPC that the Support Element Console is controlling.

   When receiving this event from a Hardware Management Console, this value:
   - Contains no CPC Image names for hardware messages for the Hardware Management Console itself
   - Contains no CPC Image names for Optical Network related hardware messages
   - Contains the name(s) of the CPC Images that are running on the CPC that the hardware message pertains to.
6. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

Note: The application should ensure that it provides a buffer that is at least large enough to hold the **HWMCA_DATATYPE_T** structures and associated data for the event notification object identifier and type. A constant, **HWMCA_MIN_EVENT_BUF_SIZE** is provided to the application for this purpose. In addition, another constant, **HWMCA_MAX_EVENT_BUF_SIZE** is provided to the application. This constant can be used to allocate a buffer large enough to hold any event notification. It is important to note that although the **HWMCA_MAX_EVENT_BUF_SIZE** constant can be used to allocate a buffer large enough for any event, it is not intended to indicate a buffer of this size is large enough for all *HwmcaGet* requests.

HWMCA_EVENT_SECURITY_EVENT

Used to notify the application that a security event has been logged.

The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the security log.
2. An HWMCA_TYPE_OCTETSTRING that specifies the text of the security log.
3. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case the console itself).

HWMCA_EVENT_CAPACITY_CHANGE

Used to notify the application that the processing capacity for a Defined CPC object has changed in some manner. The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_INTEGER that specifies the type of capacity change that occurred, using one of the following constants:
   - HWMCA_CAPACITY_FENCED_BOOK A processor book has been fenced and is not longer usable.
   - HWMCA_CAPACITY_DEFECTIVE_PROCESSOR A processor has become defective.
   - HWMCA_CAPACITY_CONCURRENT_BOOK_REPLACE A concurrent processor book replacement has been performed.
   - HWMCA_CAPACITY_CONCURRENT_BOOK_ADD A concurrent processor book addition has been performed.
   - HWMCA_CAPACITY_CHECK_STOP A processor has gone into a check stopped state.
   - HWMCA_CAPACITY_CHANGES_ALLOWED A user has configured the APIs to be allowed to perform capacity changes.
   - HWMCA_CAPACITY_CHANGES_NOT_ALLOWED A user has configured the APIs to no longer be allowed to perform capacity changes.
2. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a Defined CPC object).

HWMCA_EVENT_CAPACITY_RECORD_CHANGE

Used to notify the application that a change has occurred to a temporary capacity record. The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_INTEGER that specifies the type of capacity record change that occurred, using one of the following constants:
   - HWMCA_CAPACITY_RECORD_ADD The capacity record has been added to the machine.
   - HWMCA_CAPACITY_RECORD_DELTA The capacity record has been modified.
   - HWMCA_CAPACITY_RECORD_DELETE The capacity record has been deleted.
   - HWMCA_CAPACITY_RECORD_ACCOUNTING
   - HWMCA_CAPACITY_ACTIVATION_LEVEL The capacity record has changed it's level of activation (either more resources from this record have been added or removed from the machine).
   - HWMCA_CAPACITY_PRIORITY_PENDING Additional capacity has been added for the capacity record, with priority, but not enough resources were available to allow for all the capacity specified to be put into effect. As resources become available they will be added for this record in order to completely satisfy the original request for additional capacity.
   - HWMCA_CAPACITY_RECORD_OTHER The capacity record has changed in some other manner.
2. An HWMCA_TYPE_OCTETSTRING for the temporary capacity record identifier that has changed.
3. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a Defined CPC object).

HWMCA_EVENT_DISABLED_WAIT

Used to notify the application that a CPC Image object has entered a disabled wait state. The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_OCTETSTRING for the name of the Defined CPC that is associated with the CPC Image that entered a disabled wait state.
2. An HWMCA_TYPE_OCTETSTRING for the disabled wait PSW value.
3. An HWMCA_TYPE_INTEGER for the partition identifier of the CPC Image that entered a disabled wait state.
4. An HWMCA_TYPE_INTEGER for the number of the processor that entered a disabled wait state.
5. An HWMCA_TYPE_OCTETSTRING for the serial number of the Defined CPC that is associated with the CPC Image that entered a disabled wait state.
6. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a CPC Image object).
7. An HWMCA_TYPE_INTEGER that specifies if the disabled wait event was due to an SCP initiated reset (HWMCA_TRUE) or not (HWMCA_FALSE).

HWMCA_EVENT_LOG_EVENT

Used to notify the application that a log event has occurred. This includes log events for security, audit and console logs.

The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the security log.
2. An HWMCA_TYPE_OCTETSTRING that specifies the text of the security log.
3. An HWMCA_TYPE_INTEGER that specifies the event identifier for the log.
4. An HWMCA_TYPE_INTEGER that specifies the type of log:
   - HWMCA_CONSOLE_LOG for a console log event.
   - HWMCA_SECURITY_LOG for a security log event.
   - HWMAC_AUDIT_LOG for an audit log event.
5. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case the console itself).

HwmcaTerminate

Used to perform any cleanup tasks required by any of the other APIs. An application should always perform an *HwmcaTerminate* whenever a successful *HwmcaInitialize* has been done after the application has completed all the activities that are required using the Data Exchange APIs and Commands API. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for the *HwmcaTerminate* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

The *HwmcaTerminate* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the terminate request was successfully delivered and processed by the Console application. A value of **HWMCA_DE_NO_ERROR** indicates successful completion.

Once the *HwmcaTerminate* has been successfully called, the **HWMCA_INITIALIZE_T** structure can then be used for another purpose or freed, depending on the needs of the application.

HwmcaBuildId

A convenience routine to aid the application program in constructing an object identifier for any object supported by the Console. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pszBuffer
A pointer to a buffer where the built object identifier string is to be placed. It is recommended that this buffer be at least **HWMCA_MAX_ID_LEN** bytes in length.

pszPrefix
A pointer to the prefix string to be used for the object identifier to be built. Any of the valid prefixes defined in the Data Exchange APIs include file can be used, such as:

- HWMCA_CONSOLE_ID
- HWMCA_CFG_CPC_GROUP_ID
- HWMCA_CFG_CPC_ID
- HWMCA_CPC_IMAGE_GROUP_ID
- HWMCA_CPC_IMAGE_ID
- HWMCA_GROUPS_GROUP_ID
- HWMCA_GROUPS_OBJECT_ID
- HWMCA_COMMAND_PREFIX
- HWMCA_ACT_RESET_OBJECT_ID
- HWMCA_ACT_IMAGE_OBJECT_ID
- HWMCA_ACT_LOAD_OBJECT_ID
- HWMCA_ACT_GROUP_OBJECT_ID
- HWMCA_CAPACITY_RECORD_OBJECT_ID
- HWMCA_LPAR_GROUP_OBJECT_ID

pszAttribute
A pointer to the attribute suffix string to be used for the object identifier to be built. This can be specified as NULL, when building an identifier for an object itself, as opposed to an attribute object identifier. (Any of the **HWMCA_*_SUFFIX** constants can be specified in this argument.)

pszGroupName
A pointer to the group name to be used for building the object identifier. This can be specified as NULL, when building an object identifier for a predefined group or an object contained within a predefined group.

pszObjectName
A pointer to the object name to be used for building the object identifier. This can be specified as NULL, when building an object identifier for a predefined or user-defined group object.

Note: Refer to "Console application object identifier conventions" on page 75 for more information on the conventions used for the object identifiers for objects managed by the Console.

HwmcaBuildAttributeId

A convenience routine to aid the application program in constructing an attribute object identifier for any object supported by the Console, based on the object identifier of the object itself. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pszBuffer
A pointer to a buffer where the built object identifier string is to be placed. It is recommended that this buffer be at least **HWMCA_MAX_ID_LEN** bytes in length.

pszObjectID
A pointer to the object identifier of the object for which the attribute identifier is to be built.

pszAttribute
A pointer to the attribute suffix string to be used for the object identifier to be built. (Any of the **HWMCA_*_SUFFIX** constants can be specified in this argument.)

Note: Refer to "Console application object identifier conventions" on page 75 for more information on the conventions used for the object identifiers for objects managed by the Console.

Commands API

Allows other applications, local or remote, the ability to execute commands against the objects that the Console application manages. Specifically, this support will allow other applications to request the Console applications to perform the following commands:

- Activate
- Reset Normal
- Reset Clear
- Deactivate
- Send Operating System command
- Start
- Stop
- PSW Restart
- Load
- Hardware Message Refresh
- Hardware Message Delete
- Activate CBU
- Undo CBU
- Import Profile
- Export Profile
- Reserve
- External Interrupt
- SCSI Load
- SCSI Dump
- NVMe Load
- NVMe Dump
- Shutdown/Restart
- Activate On/Off CoD
- Undo On/Off CoD
- Add Temporary Capacity
- Remove Temporary Capacity
- Swap Current Time Server
- Set STP Configuration
- Change STP-only CTN
- Join STP-only CTN
- Leave STP-only CTN
- STP Change Daylight Savings Time

The Commands API uses the Simple Network Management Protocol (SNMP) as the transport mechanism. The underlying SNMP protocol is encapsulated in the *HwmcaCommand* API in order to reduce the

complexities for the application programmer. Refer to following pages for additional information about the *HwmcaCommand*.

HwmcaCommand

Used to perform a command against a specific object managed by the Console. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) The arguments specified for this API are:

pInitialize
A pointer to the **HWMCA_INITIALIZE_T** structure that was used on the *HwmcaInitialize* API.

pszObjectID
A pointer to a null terminated object identifier string for the target object of the command. Refer to Chapter 4, "Console application managed objects," on page 75 for more information about the object identifiers that the Console manages.

pszCommandID
A pointer to a null terminated object identifier string for the object identifier of the command that is to be executed. Valid values for this argument are:

- HWMCA_ACTIVATE_COMMAND
- HWMCA_DEACTIVATE_COMMAND
- HWMCA_RESETNORMAL_COMMAND
- HWMCA_START_COMMAND
- HWMCA_STOP_COMMAND
- HWMCA_PSWRESTART_COMMAND
- HWMCA_SEND_OPSYS_COMMAND
- HWMCA_LOAD_COMMAND
- HWMCA_HW_MESSAGE_REFRESH_COMMAND
- HWMCA_RESETCLEAR_COMMAND
- HWMCA_HW_MESSAGE_DELETE_COMMAND
- HWMCA_ACTIVATE_CBU_COMMAND
- HWMCA_UNDO_CBU_COMMAND
- HWMCA_IMPORT_PROFILE_COMMAND
- HWMCA_EXPORT_PROFILE_COMMAND
- HWMCA_RESERVE_COMMAND
- HWMCA_EXTERNAL_INTERRUPT_COMMAND
- HWMCA_SCSI_LOAD_COMMAND
- HWMCA_SCSI_DUMP_COMMAND
- HWMCA_NVME_LOAD_COMMAND
- HWMCA_NVME_DUMP_COMMAND
- HWMCA_SHUTDOWN_RESTART_COMMAND
- HWMCA_ACTIVATE_OOCOD_COMMAND
- HWMCA_UNDO_OOCOD_COMMAND
- HWMCA_ADD_CAPACITY_COMMAND
- HWMCA_REMOVE_CAPACITY_COMMAND
- HWMCA_SYSPLEX_TIME_SWAP_CTS_COMMAND
- HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND
- HWMCA_SYSPLEX_TIME_CHANGE_STP_ONLY_CTN_ COMMAND
- HWMCA_SYSPLEX_TIME_JOIN_STP_ONLY_CTN_COMMAND

- HWMCA_SYSPLEX_TIME_LEAVE_STP_ONLY_CTN_ COMMAND
- HWMCA_SYSPLEX_TIME_SET_DST_COMMAND

pDatatype
A pointer to a linked list of **HWMCA_DATATYPE_T** structures used to represent the arguments to be passed to the specified command.

The *HwmcaCommand* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the command request was successfully delivered for execution to the Console application. A value of **HWMCA_CMD_NO_ERROR** indicates successful completion.

Once the application determines that the command request has been successfully delivered to the Console, it must wait for one or more **HWMCA_EVENT_COMMAND_RESPONSE** event notification(s) for this command request. This is accomplished through the use of the *HwmcaWaitEvent*. All applications are implicitly registered for this event type. The **HWMCA_EVENT_COMMAND_RESPONSE** event notification will contain:

- Object identifier of the object for which command request was targeted,
- Object identifier for the command that was requested to be executed,
- Return code value that can be used to determine the success or failure of the command request, and
- An indication of whether this event is the last **HWMCA_EVENT_COMMAND_RESPONSE** event notification that should be expected for this command.

Refer to "HwmcaWaitEvent" on page 14 for more details regarding the data returned from the *HwmcaWaitEvent* for the **HWMCA_EVENT_COMMAND_RESPONSE** event notification.

The exceptions to this rule are **HWMCA_HW_MESSAGE_REFRESH_COMMAND** and **HWMCA_HW_MESSAGE_DELETE_COMMAND** commands. There is no need to wait for a **HWMCA_EVENT_COMMAND_RESPONSE** event notification for these commands. These commands are finished once the HwmcaCommand has completed.

HwmcaCorrelatedCommand

Used to perform a command against a specific object managed by the Console. (Refer to "Function prototypes" on page 63 for the C function prototype for this API.) While similar to the *HwmcaCommand* API, this API call is intended to be used to allow the caller to specify some unique correlator data that will then be provided back to the caller as part of the **HWMCA_EVENT_COMMAND_RESPONSE** event, so that the caller can be sure that the event was a result of the command that it requested to be executed. The arguments specified for this API are:

pInitialize
A pointer to the HWMCA_INITIALIZE_T structure that was used on the HwmcaInitialize API.

pszObjectId
A pointer to a null terminated object identifier string for the target object of the command. Refer to Chapter 4, "Console application managed objects," on page 75 for more information about the object identifiers that the Console manages.

pszCommandId
A pointer to a null terminated object identifier string for the object identifier of the command that is to be executed.

pDataType
A pointer to a linked list of **HWMCA_DATATYPE_T** structures used to represent the arguments to be passed to the specified command.

ulTimeout
Used to specify the amount of time that the calling application wants to wait for the *HwmcaCorrelatedCommand* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

pCorrelator
A pointer to the data to be used as a correlator for the specified command.

correlatorSize
The length of the correlator data.

The *HwmcaCorrelatedCommand* API returns an unsigned long integer return code value to the calling application. This return code lets the calling application know if the command request was successfully delivered for execution to the Console application. A value of **HWMCA_CMD_NO_ERROR** indicates successful completion. Once the application determines that the command request has been successfully delivered to the Console, it must wait for one or more **HWMCA_EVENT_COMMAND_RESPONSE** event notification(s) for this command request. This is accomplished through the use of the *HwmcaWaitEvent*. All applications are implicitly registered for this event type. The **HWMCA_EVENT_COMMAND_RESPONSE** event notification will contain:

- Object identifier of the object for which command request was targeted,
- Object identifier for the command that was requested to be executed,
- Return code value that can be used to determine the success or failure of the command request, and
- An indication of whether this event is the last **HWMCA_EVENT_COMMAND_RESPONSE** event notification that should be expected for this command.
- The command correlator specified when the command was invoked.

Refer to "HwmcaWaitEvent" on page 14 for more details regarding the data returned from the *HwmcaWaitEvent* for the **HWMCA_EVENT_COMMAND_RESPONSE** event notification.

Command arguments

The acceptable and/or required arguments for each command are as follows.

HWMCA_ACTIVATE_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Activation profile name
Name of the activation profile to be used for the Activate command. The default is to use the profile name specified in the Activation profile name attribute for the specified object.

Force indicator
An indicator used to request conditional processing of the Activate command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE=TRUE) no matter what the state of the target object is.

Either one or both of these arguments can be specified, but they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the **HWMCA_DATATYPE_T** structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Activation profile name

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the activation profile name (including the null terminator).

pData
A pointer to the activation profile name itself.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Force Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_DEACTIVATE_COMMAND
No arguments are required, but optionally a Force indicator can be specified for the Deactivate command. If this argument is not specified, then the default is to unconditionally perform the command (that is, FORCE=TRUE) no matter what the state of the target object is. The fields of the HWMCA_DATATYPE_T structure used to describe the optional Force indicator are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_RESETNORMAL_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Force indicator
An indicator used to request conditional processing of the Reset Normal command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE-TRUE) no matter what the state of the target object is.

IPL Token
An IPL token to associate with the Reset Normal command. The default is to not associate an IPL token with the command.

Either one or both of these arguments can be specified, but they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Force Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

IPL Token

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the IPL token.

pData
A pointer to the IPL token itself.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_RESETCLEAR_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Force indicator
An indicator used to request conditional processing of the Reset Clear command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE-TRUE) no matter what the state of the target object is.

IPL Token
An IPL token to associate with the Reset Clear command. The default is to not associate an IPL token with the command.

Either one or both of these arguments can be specified, but they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Force Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

IPL Token

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the IPL token.

pData
A pointer to the IPL token itself.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_START_COMMAND
No arguments are accepted or required.

HWMCA_STOP_COMMAND
No arguments are accepted or required.

HWMCA_PSWRESTART_COMMAND
No arguments are accepted or required.

HWMCA_SEND_OPSYS_COMMAND
This command requires the following two arguments:

- An indication of whether this is a priority operating system command
- The text of the operating system command.

The fields of the **HWMCA_DATATYPE_T** structures used to describe these two arguments are:

Priority Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for priority operating system commands or HWMCA_FALSE for nonpriority operating system commands.

pNext
Should be set to the address of the **HWMCA_DATATYPE_T** structure used to describe the text for the operating system command itself.

Operating System Command Text

ucType
Should be set to HWMCA_TYPE_OCTETSTRING

ulLength
Should be set to the length of the operating system command (including the null terminator).

Note: The operating system command itself should have a length of at least one byte, not including the null terminator.

pData
Should be a pointer to the operating system command itself.

pNext
Should be set to NULL, since this is the last argument expected for this command.

HWMCA_LOAD_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Load address
Hexadecimal address to be used when performing the Load. The default will be to use the Load address last used when a Load was performed for the object.

Load parameter
Parameter string to be used when performing the Load. The default will be to use the Load parameter last used when a Load was performed for the object.

Clear indicator
Whether or not memory should be cleared before performing the Load. The default is to clear memory before performing the Load.

Timeout
Amount of time (in seconds) to wait for the Load to complete. The default timeout is 60 seconds.

Store status indicator
Whether or not status should be stored before performing the Load. The default is not to store status before performing the Load.

Force indicator
An indicator used to request conditional processing of the Load command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE=TRUE) no matter what the state of the target object is.

IPL Token
An IPL token to associate with the Load command. The default is to not associate an IPL token with the command.

Any number of arguments can be specified; however, they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the **HWMCA_DATATYPE_T** structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

The default for any argument can be overridden by specifying the **HWMCA_DATATYPE_T** structure used to describe the argument as follows:

Load address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the address string to be used when performing the Load (including the null terminator). This string (including the null terminator) must be less than or equal to 6 characters.

pData
Should be a pointer to a field containing the address string to be used when performing the Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

Load parameter

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the parameter string to be used when performing the Load (including the null terminator). This string (including the null terminator) must be less than or equal to nine characters.

pData
Should be a pointer to a field containing the parameter string to be used when performing the Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

Clear indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for memory to be cleared before performing the Load or HWMCA_FALSE to bypass the clearing of memory before performing the Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

Timeout

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the timeout value that is to be used when performing the Load. This value must be between 60 seconds and 600 seconds.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

Store status indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for status to be stored before performing the Load or HWMCA_FALSE to bypass the storing of status before performing the Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

Force indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

IPL Token

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the IPL token.

pData
A pointer to the IPL token itself.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the **HWMCA_DATATYPE_T** structure used to describe the next argument.

HWMCA_HW_MESSAGE_REFRESH_COMMAND
No arguments are accepted or required.

HWMCA_HW_MESSAGE_DELETE_COMMAND
This command requires the following argument:

- The time stamp of the hardware message.

The fields of the **HWMCA_DATATYPE_T** structure used to describe the time stamp value are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the time stamp (including the null terminator).

pData
A pointer to the time stamp string itself.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_ACTIVATE_CBU_COMMAND
This command has one required and one optional argument:

- An indicator of whether a real or test CBU activation should be performed is required.
- The password used to validate the CBU activation is optional. If not specified, the password will be obtained automatically from the IBM support system.

The fields of the **HWMCA_DATATYPE_T** structure used to describe these arguments are:

Real/Test Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for a real CBU activation or HWMCA_FALSE for a test CBU activation.

pNext
Should be set to NULL, if this is the last argument to being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Password

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the password (including the null terminator).

pData
A pointer to the password string itself.

pNext
Should be set to NULL, if this is the last argument expected for this command.

HWMCA_UNDO_CBU_COMMAND
No arguments are accepted or required.

HWMCA_IMPORT_PROFILE_COMMAND
This command requires the following argument:

- The profile area to be imported.

The fields of the **HWMCA_DATATYPE_T** structure used to describe the profile area are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
Should be an integer value greater than or equal to 1 and less than or equal to 4, indicating the profile area to be imported.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_EXPORT_PROFILE_COMMAND
This command requires the following argument:

- The profile area to be exported.

The fields of the **HWMCA_DATATYPE_T** structure used to describe the profile area are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
Should be an integer value greater than or equal to 1 and less than or equal to 4, indicating the profile area to be exported.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_RESERVE_COMMAND

Note: This command is available only on a Support Element console. After successfully issuing this command to request the reserve, all API command requests and the majority of other API requests will be blocked, including those from the issuer of the reserve request, until the reserve is released.

This command requires the following arguments:

- An indicator of whether the reserve is being requested or released.
- The name of the application requesting/releasing the reserve (exclusive control).

The fields of the **HWMCA_DATATYPE_T** structure used to describe these two arguments are:

Request/Release Indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE when requesting the reserve or HWMCA_FALSE when releasing the reserve.

pNext
Should be set to the address of the HWMCA_DATATYPE_T structure used to describe the text for the application name.

Application Name

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the application name (including the null terminator). The length of this field including the null terminator must be less than or equal to 9 characters.

pData
A pointer to the application itself.

pNext
Should be set to NULL, since this is the last argument expected for this command.

HWMCA_EXTERNAL_INTERRUPT_COMMAND

This command requires the following argument:

- The number of the processor that is the target of the external interrupt command. This is a number between zero and the maximum number of processors for the target CPC Image object.

The fields of the **HWMCA_DATATYPE_T** structure used to describe the application name are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to the processor number.

pNext
Should be set to NULL, since this command only accepts one argument.

ulTimeOut
Used to specify the amount of time that the calling application wants to wait for the *HwmcaCommand* to complete. This value is specified in milliseconds and the value of **HWMCA_INFINITE_WAIT** can be used to cause the application to wait forever.

HWMCA_SCSI_LOAD_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Load address
Hexadecimal address to be used when performing the SCSI Load. The default will be to use the Load address last used when a SCSI Load was performed for the object.

Load parameter
Parameter string to be used when performing the SCSI Load. The default will be to use the Load parameter last used when a SCSI Load was performed for the object.

Worldwide port name
The worldwide port name (WWPN) to be used for the SCSI Load. The default will be to use the worldwide port name last used when a SCSI Load was performed for the object.

Logical unit number
The logical unit number (LUN) to be used for the SCSI Load. The default will be to use the logical unit number last used when a SCSI Load was performed for the object.

Boot program selector
The boot program selector to be used for the SCSI Load. The default will be to use the boot program selector last used when a SCSI Load was performed for the object.

Operating system specific load parameters
The operating system specific load parameters to be used for the SCSI Load. The default will be to use the operating system specific load parameters last used when a SCSI Load was performed for the object.

Boot record logical block address
The boot record logical block address to be used for the SCSI Load. The default will be to use the boot record logical block address last used when a SCSI Load was performed for the object.

Force indicator
An indicator used to request conditional processing of the SCSI Load command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE=TRUE) no matter what the state of the target object is.

Clear indicator
Whether or not memory should be cleared before performing the Load. The default is to clear memory before performing the Load.

Any number of arguments can be specified; however, they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Load address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the address string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 6 characters.

pData
Should be a pointer to a field containing the address string to be used when performing the SCSI Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Load parameter

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the parameter string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 9 characters.

pData
Should be a pointer to a field containing the parameter string to be used when performing the SCSI Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Worldwide port name

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the worldwide port name string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the worldwide port name string to be used when performing the SCSI Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Logical unit number

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the logical unit number string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the logical unit number string to be used when performing the SCSI Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Disk Partition Identifer

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the boot program selector value, which can be in the range 0 - 30, inclusive.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Operating system specific load parameters

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the operating system specific parameters string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 257 characters.

pData
Should be a pointer to a field containing the operating system specific parameters string to be used when performing the SCSI Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot record logical block address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the boot record logical block address string to be used when performing the SCSI Load (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the boot record logical block address string to be used when performing the SCSI Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Force indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Clear indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for memory to be cleared before performing the Load or HWMCA_FALSE to bypass the clearing of memory before performing the Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_SCSI_DUMP_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Load address
Hexadecimal address to be used when performing the SCSI Dump. The default will be to use the Load address last used when a SCSI Dump was performed for the object.

Load parameter
Parameter string to be used when performing the SCSI Dump. The default will be to use the Load parameter last used when a SCSI Dump was performed for the object.

Worldwide port name
The worldwide port name (WWPN) to be used for the SCSI Dump. The default will be to use the worldwide port name last used when a SCSI Dump was performed for the object.

Logical unit number
The logical unit number (LUN) to be used for the SCSI Dump. The default will be to use the logical unit number last used when a SCSI Dump was performed for the object.

Boot program selector
The boot program selector to be used for the SCSI Dump. The default will be to use the boot program selector last used when a SCSI Dump was performed for the object.

Operating system specific load parameters
The operating system specific load parameters to be used for the SCSI Dump. The default will be to use the operating system specific load parameters last used when a SCSI Dump was performed for the object.

Boot record logical block address
The boot record logical block address to be used for the SCSI Dump. The default will be to use the boot record logical block address last used when a SCSI Dump was performed for the object.

Force indicator
An indicator used to request conditional processing of the SCSI Dump command depending on the state of the target object. The default is to unconditionally perform the command (that is, FORCE=TRUE) no matter what the state of the target object is.

Any number of arguments can be specified; however, they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order

to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Load address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the address string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 5 characters.

pData
Should be a pointer to a field containing the address string to be used when performing the SCSI Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Load parameter

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the parameter string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 9 characters.

pData
Should be a pointer to a field containing the parameter string to be used when performing the SCSI Dump.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Worldwide port name

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the worldwide port name string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the worldwide port name string to be used when performing the SCSI Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Logical unit number

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the logical unit number string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the logical unit number string to be used when performing the SCSI Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Disk Partition Identifer

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the boot program selector value, which can be in the range 0 to 30, inclusive.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Operating system specific load parameters

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the operating system specific parameters string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 257 characters.

pData
Should be a pointer to a field containing the operating system specific parameters string to be used when performing the SCSI Dump.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot record logical block address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the boot record logical block address string to be used when performing the SCSI Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the boot record logical block address string to be used when performing the SCSI Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Force indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_NVME_LOAD_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Load address
Hexadecimal address to be used when performing the NVMe Load. The default will be to use the Load address last used when an NVMe Load was performed for the object.

Load parameter
Parameter string to be used when performing the NVMe Load. The default will be to use the Load parameter last used when an NVMe Load was performed for the object.

Boot program selector
The boot program selector to be used for the NVMe Load. The default will be to use the boot program selector last used when an NVMe Load was performed for the object.

Operating system specific load parameters
The operating system specific load parameters to be used for the NVMe Load. The default will be to use the operating system specific load parameters last used when an NVMe Load was performed for the object.

Boot record logical block address
The boot record logical block address to be used for the NVMe Load. The default will be to use the boot record logical block address last used when an NVMe Load was performed for the object.

Clear indicator
Whether or not memory should be cleared before performing the NVMe Load. The default will be to use the last clear indicator last used when an NVMe Load was performed for the object.

Secure load type
Whether or not the loaded software signature is verified. The default will be to use the secure load type last used when an NVMe Load was performed for the object.

Any number of arguments can be specified; however, they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Load address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the address string to be used when performing the NVMe Load (including the null terminator). This string (including the null terminator) must be less than or equal to 6 characters.

pData
Should be a pointer to a field containing the address string to be used when performing the NVMe Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Load parameter

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the parameter string to be used when performing the NVMe Load (including the null terminator). This string (including the null terminator) must be less than or equal to 9 characters.

pData
Should be a pointer to a field containing the parameter string to be used when performing the NVMe Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot program selector (Disk partition identifier)

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
Should be a pointer to a field containing the boot program selector value, which can be in the range 0 - 30, inclusive.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Operating system specific load parameters

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the operating system specific parameters string to be used when performing the NVMe Load (including the null terminator). This string (including the null terminator) must be less than or equal to 257 characters.

pData
Should be a pointer to a field containing the operating system specific parameters string to be used when performing the NVMe Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot record logical block address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the boot record logical block address string to be used when performing the NVMe Load (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the boot record logical block address string to be used when performing the NVMe Load. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Clear indicator

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for memory to be cleared before performing the Load or HWMCA_FALSE to bypass the clearing of memory before performing the Load.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Verify software signature

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE to indicate that the loaded software signature is verified or HWMCA_FALSE to indicate that the loaded software signature is not verified.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_NVME_DUMP_COMMAND
No arguments are required, but the following arguments can optionally be specified:

Load address
Hexadecimal address to be used when performing the NVMe Dump. The default will be to use the Load address last used when an NVMe Dump was performed for the object.

Load parameter
Parameter string to be used when performing the NVMe Dump. The default will be to use the Load parameter last used when an NVMe Dump was performed for the object.

Boot program selector
The boot program selector to be used for the NVMe Dump. The default will be to use the boot program selector last used when an NVMe Dump was performed for the object.

Operating system specific load parameters
The operating system specific load parameters to be used for the NVMe Dump. The default will be to use the operating system specific load parameters last used when an NVMe Dump was performed for the object.

Boot record logical block address
The boot record logical block address to be used for the NVMe Dump. The default will be to use the boot record logical block address last used when an NVMe Dump was performed for the object.

Secure load type
Whether or not the loaded software signature is verified. The default will be to use the secure load type last used when an NVMe Dump was performed for the object.

Any number of arguments can be specified; however, they must be specified in the order shown by the preceding list. If an argument is not specified, then the default for that argument is used. In order to specify an argument, such that the default will be used, the HWMCA_DATATYPE_T structure used to describe the argument should be specified as follows:

ucType
Should be set to HWMCA_TYPE_NULL.

ulLength
Should be set to zero.

pData
A pointer value of zero.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

The default for any argument can be overridden by specifying the HWMCA_DATATYPE_T structure used to describe the argument as follows:

Load address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the address string to be used when performing the NVMe Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 5 characters.

pData
Should be a pointer to a field containing the address string to be used when performing the NVMe Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Load parameter

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the parameter string to be used when performing the NVMe Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 9 characters.

pData
Should be a pointer to a field containing the parameter string to be used when performing the NVMe Dump.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot program selector (Disk partition identifier)

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
Should be a pointer to a field containing the boot program selector value, which can be in the range 0 - 30, inclusive.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Operating system specific load parameters

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the operating system specific parameters string to be used when performing the NVMe Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 257 characters.

pData
Should be a pointer to a field containing the operating system specific parameters string to be used when performing the NVMe Dump.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Boot record logical block address

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the boot record logical block address string to be used when performing the NVMe Dump (including the null terminator). This string (including the null terminator) must be less than or equal to 17 characters.

pData
Should be a pointer to a field containing the boot record logical block address string to be used when performing the NVMe Dump. This string must consist of only hexadecimal characters.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

Verify software signature

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE to indicate that the loaded software signature is verified or HWMCA_FALSE to indicate that the loaded software signature is not verified.

pNext
Should be set to NULL if this is the last argument being specified, or this should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

HWMCA_SHUTDOWN_RESTART_COMMAND
This command requires the following argument:

- An indicator of the type of shutdown or restart to be performed.

The fields of the **HWMCA_DATATYPE_T** structure used to describe this shutdown/restart type are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing one of the following values:

- HWMCA_RESTART_APPLICATION - Used to indicate the Console application is to be restarted.

  Note: For Support Element consoles, this value will implicitly cause the Console to be restarted.
- HWMCA_RESTART_CONSOLE - Used to indicate the Console is to be restarted.
- HWMCA_SHUTDOWN_CONSOLE - Used to indicate the Console is to be shutdown/powered off.
- HWMCA_RESTART_APPLICATION_ALTERNAT E - Used to indicate the Alternate Support Element Console application is to be restarted. This option is only valid for the Support Element Console.
- HWMCA_RESTART_CONSOLE_ALTERNATE -Used to indicate the Alternate Support Element Console is to be restarted. This option is only valid for the Support Element Console.

  Note: This value will implicitly cause the Alternate Console to be restarted.
- HWMCA_SHUTDOWN_CONSOLE_ALTERNATE - Used to indicate the Alternate Support Element Console is to be shutdown/powered off. This option is only valid for the Support Element Console.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_ACTIVATE_OOCOD_COMMAND
This command requires the following argument:

- The order number of the On/Off Capacity on Demand (On/Off CoD) record to be activated.

  The fields of the **HWMCA_DATATYPE_T** structure used to describe the order number are:

  ucType
  Should be set to HWMCA_TYPE_OCTETSTRING.

  ulLength
  Should be set to length of the order number string (including the null terminator).

  pData
  A pointer to the string itself.

  pNext
  Should be set to NULL, since this command only accepts one argument.

HWMCA_UNDO_OOCOD_COMMAND
No arguments are accepted or required.

HWMCA_ADD_CAPACITY_COMMAND
This command, which is used to add temporary capacity to a Defined CPC object, requires the following argument:

- An XML fragment describing the temporary capacity to be added. This XML is used to describe:
  - the identifier of the capacity record to be used,
  - the software model to be used for the capacity addition (optional),
  - the delta processor information to be used for the capacity addition (optional),
  - an indicator for whether the capacity addition is a priority request, (optional, default false), and
  - an indicator for whether the additional capacity is to be added as test or real.

  Note: Refer to Appendix F, "XML descriptions," on page 209 for a detailed description of this XML data.

The fields of the **HWMCA_DATATYPE_T** structure used to describe the capacity information XML are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to length of the capacity information XML string.

pData
A pointer to a the capacity information XML string.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_REMOVE_CAPACITY_COMMAND
This command, which is used to remove temporary capacity from a Defined CPC object, requires the following argument:

- An XML fragment describing the temporary capacity to be removed. This XML is used to describe:
  - the identifier of the capacity record to be used,
  - the software model to be used for the capacity removal (optional), and
  - the delta processor information to be used for the capacity removal (optional).

  Note: Refer to Appendix F, "XML descriptions," on page 209 for a detailed description of this XML data.

The fields of the HWMCA_DATATYPE_T structure used to describe the capacity information XML are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to length of the capacity information XML string.

pData
A pointer to a the capacity information XML string.

pNext
Should be set to NULL, since this command only accepts one argument.

HWMCA_SYSPLEX_TIME_SWAP_CTS_COMMAND
In a configured STP-only Coordinated Timing Network (CTN), one CPC has the role of Current Time Server (CTS). If the CTN has both a Preferred Time Server and a Backup Time Server configured, either one can be the CTS. This command swaps the role of CTS from Preferred Time Server to Backup Time Server or vice versa. The target system must be the system that will become the CTS.

This command requires the following argument:

STP ID
A string representing the current STP identifier for the Defined CPC object.

The fields of the HWMCA_DATATYPE_T structure used to describe the STP ID are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP ID. This string (including the null terminator) must be less than or equal to nine characters.

pData
A pointer to a field containing the STP ID string.

pNext
Should be set to NULL since this command only accepts one argument.

HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND
This command sets the configuration for an STP-only Coordinated Timing Network (CTN). The target system must be the system that will become the Current Time Server (CTS).

This command requires the following arguments:

STP ID
A string representing the current STP identifier for the Defined CPC object. This is used to verify that the CPC is a member of correct CTN.

The fields of the HWMCA_DATATYPE_T structure used to describe the STP ID are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP ID. This string (including the null terminator) must be less than or equal to nine characters.

pData
A pointer to a field containing the STP ID string.

pNext
Should be set to the HWMCA_DATATYPE_T structure used to describe the next argument.

Force Indicator
An indicator used to request conditional processing of the command depending on the state of the target object.

The fields of the HWMCA_DATATYPE_T structure used to describe the Force Indicator are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE for the command to be performed unconditionally or HWMCA_FALSE for the command to be performed conditionally based on the state of the target object.

pNext
Should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

STP Config XML
An XML fragment describing the configuration for the STP-only CTN. This XML describes:

- the identifier for the STP-only CTN (optional)
- the identity of the CPC to act as Preferred Time Server for the CTN
- the identity of the CPC to act as Backup Time Server for the CTN (optional)

- the identity of the CPC to act as Arbiter for the CTN (optional)
- an indicator of which CPC has the role of Current Time Server (Preferred Time Server or Backup Time Server)

Note: Refer to Appendix F, "XML descriptions," on page 209 for a detailed description of this XML data.

The fields of the HWMCA_DATATYPE_T structure used to describe the STP-only CTN configuration are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP Configuration data XML string.

pData
A pointer to the STP Configuration data XML string.

pNext
Should be set to NULL since this is the last argument accepted by this command.

HWMCA_SYSPLEX_TIME_CHANGE_STP_ONLY_CTN_COMMAND
This command, sent to the Defined CPC with the role of Current Time Server (CTS) in an STP-only Coordinated Timing Network (CTN), changes the STP ID portion of the CTN ID for the entire STP-only CTN.

This command requires the following argument:

STP ID
A string representing the desired STP identifier for the Defined CPC object and all CPCs that are members of the same STP-only CTN.

The fields of the HWMCA_DATATYPE_T structure used to describe the STP ID are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP ID. This string (including the null terminator) must be less than or equal to nine characters.

pData
A pointer to a field containing the STP ID string.

pNext
Should be set to NULL since this command only accepts one argument.

HWMCA_SYSPLEX_TIME_JOIN_STP_ONLY_CTN_COMMAND
This command allows a CPC to join an STP-only Coordinated Timing Network (CTN). The target system cannot be the Current Time Server. If the CPC is already participating in an STP-only CTN, it will be removed from that CTN and join the specified one. If the CPC has an ETR ID, it will be removed.

This command requires the following argument:

STP ID
A string representing the STP identifier of the CTN that the Defined CPC object is joining.

The fields of the HWMCA_DATATYPE_T structure used to describe the STP ID are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP ID. This string (including the null terminator) must be less than or equal to nine characters.

pData
A pointer to a field containing the STP ID string.

pNext
Should be set to NULL since this command only accepts one argument.

HWMCA_SYSPLEX_TIME_LEAVE_STP_ONLY_CTN_COMMAND
This command removes a CPC from an STP-only Coordinated Timing Network (CTN). The target system cannot be the Current Time Server.

No arguments are accepted or required.

HWMCA_SYSPLEX_TIME_SET_DST_COMMAND

This command sets the Daylight Savings Time setting in an STP-only Coordinated Timing Network (CTN).

This command requires the following arguments:

STP ID
A string representing the desired STP identifier for the Defined CPC object and all CPCs that are members of the same STP-only CTN. The fields of the HWMCA_DATATYPE_T structure used to describe the STP ID are:

ucType
Should be set to HWMCA_TYPE_OCTETSTRING.

ulLength
Should be set to the length of the STP ID. This string (including the null terminator) must be less than or equal to nine characters

pData
A pointer to a field containing the STP ID string.

pNext
Should point to the HWMCA_DATATYPE_T structure used to describe the next argument.

DST Enabled
An indicator for the desired Daylight Savings Time setting. The fields of the HWMCA_DATATYPE_T structure used to describe the DST Enabled parameter are:

ucType
Should be set to HWMCA_TYPE_INTEGER.

ulLength
Should be set to 2.

pData
A pointer to a field containing the value HWMCA_TRUE to enable DST or HWMCA_FALSE to disable it.

pNext
Should be set to NULL since this is the last argument accepted by this command.

Data exchange APIs and commands API structures and definitions

The following structure and constant definitions can be found in the Data Exchange APIs. The most up to date copy of this code is available on Resource Link at http://www.ibm.com/servers/resourcelink. Click **Services**, and then Click **API**.

Constant definitions

```
/*****************************************************************************/
/* Defines for the Console Data Exchange                                     */
/* Return Code Values.                                                       */
/*****************************************************************************/
#define HWMCA_DE_NO_ERROR                0
#define HWMCA_DE_NO_SUCH_OBJECT          1
#define HWMCA_DE_INVALID_DATA_TYPE       2
#define HWMCA_DE_INVALID_DATA_LENGTH     3
#define HWMCA_DE_INVALID_DATA_PTR        4
```

```
#define HWMCA_DE_INVALID_DATA_VALUE        5
#define HWMCA_DE_INVALID_INIT_PTR          6
#define HWMCA_DE_INVALID_ID_PTR            7
#define HWMCA_DE_INVALID_BUF_PTR           8
#define HWMCA_DE_INVALID_BUF_SIZE          9
#define HWMCA_DE_INVALID_DATATYPE_PTR      10
#define HWMCA_DE_INVALID_TARGET            11
#define HWMCA_DE_INVALID_EVENT_MASK        12
#define HWMCA_DE_INVALID_PARAMETER         13
#define HWMCA_DE_READ_ONLY_OBJECT          14
#define HWMCA_DE_SNMP_INIT_ERROR           15
#define HWMCA_DE_INVALID_OBJECT_ID         16
#define HWMCA_DE_REQUEST_ALLOC_ERROR       17
#define HWMCA_DE_REQUEST_SEND_ERROR        18
#define HWMCA_DE_TIMEOUT                   19
#define HWMCA_DE_REQUEST_RECV_ERROR        20
#define HWMCA_DE_SNMP_ERROR                21
#define HWMCA_DE_INVALID_TIMEOUT           22
#define HWMCA_DE_OBJECT_BUSY               24
#define HWMCA_DE_INVALID_HOST              28
#define HWMCA_DE_INVALID_COMMUNITY         29
#define HWMCA_DE_INVALID_QUALIFIER         30
#define HWMCA_DE_PROTOCOL_ERROR            31
#define HWMCA_DE_INVALID_EVENT_ERROR       32
#define HWMCA_DE_INVALID_VARBIND_PTR       33
#define HWMCA_DE_PARTIAL_SUCCESS           34
#define HWMCA_DE_AUTHORIZATION_ERROR       74
#define HWMCA_DE_STATS_NOT_IN_TIME_WINDOWS 94
#define HWMCA_DE_UNKNOWN_USERNAME          95
#define HWMCA_DE_INCORRECT_PASSWORD        96
#define HWMCA_DE_INVALID_STACKNAME         97
#define HWMCA_DE_REQUIRES_QUALIFIER        98
#define HWMCA_DE_TRANSPORT_ERROR           99
```

```
/*****************************************************************************/
/* Defines for the Console Command Return Code Values                        */
/*****************************************************************************/
#define HWMCA_CMD_NO_ERROR                          0
#define HWMCA_CMD_NO_SUCH_OBJECT                    1
#define HWMCA_CMD_INVALID_DATA_TYPE                 2
#define HWMCA_CMD_INVALID_DATA_LENGTH               3
#define HWMCA_CMD_INVALID_DATA_PTR                  4
#define HWMCA_CMD_INVALID_DATA_VALUE                5
#define HWMCA_CMD_INVALID_INIT_PTR                  6
#define HWMCA_CMD_INVALID_ID_PTR                    7
#define HWMCA_CMD_INVALID_DATATYPE_PTR              10
#define HWMCA_CMD_INVALID_PARAMETER                 13
#define HWMCA_CMD_REQUEST_ALLOC_ERROR               17
#define HWMCA_CMD_REQUEST_SEND_ERROR                18
#define HWMCA_CMD_TIMEOUT                           19
#define HWMCA_CMD_REQUEST_RECV_ERROR                20
#define HWMCA_CMD_SNMP_ERROR                        21
#define HWMCA_CMD_INVALID_TIMEOUT                   22
#define HWMCA_CMD_INVALID_CMD                       23
#define HWMCA_CMD_OBJECT_BUSY                       24
#define HWMCA_CMD_INVALID_OBJECT                    25
#define HWMCA_CMD_COMMAND_FAILED                    26
#define HWMCA_CMD_INITTERM_OK                       27
#define HWMCA_CMD_CBU_DISRUPTIVE_OK                 28
#define HWMCA_CMD_CBU_PARTIAL_HW                    29
#define HWMCA_CMD_CBU_NO_SPARES                     30
#define HWMCA_CMD_CBU_TEMPORARY                     31
#define HWMCA_CMD_CBU_NOT_ENABLED                   32
#define HWMCA_CMD_CBU_NOT AUTHORIZED                33
#define HWMCA_CMD_CBU_FAILED                        34
#define HWMCA_CMD_CBU_ALREADY_ACTIVE                35
#define HWMCA_CMD_CBU_INPROGRESS                    36
#define HWMCA_CMD_CBU_CPSAP_SPLIT_CHG               37
#define HWMCA_CMD_INVALID_MACHINE_STATE             38
#define HWMCA_CMD_NO_RECORDID                       39
#define HWMCA_CMD_NO_SW_MODEL                       40
#define HWMCA_CMD_NOT_ENOUGH_RESOURCES              41
#define HWMCA_CMD_NOT_ENOUGH_ACTIVE_RESOURCES       42
#define HWMCA_CMD_ACT_LESS_RESOURCES                43
#define HWMCA_CMD_DEACT_MORE_RESOURCES              44
#define HWMCA_CMD_ACT_TYPE_MISMATCH                 45
#define HWMCA_CMD_API_NOT_ALLOWED                   46
#define HWMCA_CMD_CDU_IN_PROGRESS                   47
#define HWMCA_CMD_MIRRORING_RUNNING                 48
#define HWMCA_CMD_COMMUNICATIONS_NOT_ACTIVE         49
#define HWMCA_CMD_RECORD_EXPIRED                    50
```

```
#define HWMCA_CMD_PARTIAL_CAPACITY                      51
#define HWMCA_CMD_INVALID_REQUEST                       52
#define HWMCA_CMD_ALREADY_ACTIVE                        53
#define HWMCA_CMD_RESERVE_HELD                          54
#define HWMCA_CMD_GENERAL_XML_PARSING_ERROR             55
#define HWMCA_CMD_STP_NOT_ENABLED                       56
#define HWMCA_CMD_STP_MUST_TARGET_CTS                   57
#define HWMCA_CMD_STP_INVALID_CONFIG_SPECIFIED          58
#define HWMCA_CMD_STP_WRONG_CTN                         59
#define HWMCA_CMD_STP_NOT_VALID_FOR_CTS                 60
#define HWMCA_CMD_STP_IN_ETR_MIGRATION                  61
#define HWMCA_CMD_STP_NODE_NOT_FOUND_IN_SYSTEM_LIST     62
#define HWMCA_CMD_STP_CTNID_TAG_ERROR                   63
#define HWMCA_CMD_STP_NODE_TAG_ERROR                    64
#define HWMCA_CMD_STP_CONFIG_TAG_NOT_FOUND              65
#define HWMCA_CMD_STP_ACTIVE_CTS_TAG_ERROR              66
#define HWMCA_CMD_STP_INITIALIZE_INCOMPLETE             67
#define HWMCA_CMD_STP_INVALID_STP_ID                    68
#define HWMCA_CMD_STP_LINKS_ERROR                       69
#define HWMCA_CMD_STP_REQUIRES_FORCE_TO_CONFIGURE       70
#define HWMCA_CMD_PROCESSOR_POWER_MODE_NOT_ENTITLED     71
#define HWMCA_CMD_PROCESSOR_POWER_MODE_NOT_ALLOWED      72
#define HWMCA_CMD_PROCESSOR_POWER_MODE_GROUP_CONTROLLED 73
#define HWMCA_CMD_AUTHORIZATION_ERROR                   74
```

```
/*****************************************************************************/
/* Defines for the Console Rexx I/F Return Code Value                        */
/*****************************************************************************/
#define HWMCA_RX_INVALID_STEM_VAR      1000
```

```
/*****************************************************************************/
/* Miscellaneous defines for the Console APIs.                               */
/*****************************************************************************/
#define HWMCA_INFINITE_WAIT        -1
#define HWMCA_MAX_ID_LEN           80
#define HWMCA_MAX_COMMUNITY_LEN    16
#define HWMCA_MIN_EVENT_BUF_SIZE   ((sizeof(HWMCA_DATATYPE_T)*2)+4+HWMCA_MAX_ID_LEN)
#define HWMCA_MAX_EVENT_BUF_SIZE   (HWMCA_MIN_EVENT_BUF_SIZE+4+9+8+9+4+4+4+9+4+4096+\
                                    (((sizeof(HWMCA_DATATYPE_T)*2)+HWMCA_MAX_ID_LEN)*11))
#define HWMCA_TRUE                 1
#define HWMCA_FALSE                0
#define HWMCA_API_PORT             3161
```

```
/*****************************************************************************/
/* Defines for the Console Object Data Types.                                */
/*****************************************************************************/
#define HWMCA_TYPE_SEQUENCE        0x30
#define HWMCA_TYPE_OPAQUE_SEQUENCE 0x44
#define HWMCA_TYPE_INTEGER         0x02
#define HWMCA_TYPE_OCTETSTRING     0x04
#define HWMCA_TYPE_NULL            0x05
#define HWMCA_TYPE_OBJECTID        0x06
#define HWMCA_TYPE_IPADDRESS       0x40
#define HWMCA_TYPE_COUNTER         0x41
#define HWMCA_TYPE_GAUGE           0x42
#define HWMCA_TYPE_TIMETICKS       0x43
```

```
/*****************************************************************************/
/* Defines for the Console Event Notification Types.                         */
/*****************************************************************************/
#define HWMCA_EVENT_COMMAND_RESPONSE              0x00000000
#define HWMCA_EVENT_MESSAGE                       0x00000001
#define HWMCA_EVENT_STATUS_CHANGE                 0x00000002
#define HWMCA_EVENT_NAME_CHANGE                   0x00000004
#define HWMCA_EVENT_ACTIVATE_PROF_CHANGE          0x00000008
#define HWMCA_EVENT_CREATED                       0x00000010
#define HWMCA_EVENT_DESTROYED                     0x00000020
#define HWMCA_EVENT_EXCEPTION_STATE               0x00000040
#define HWMCA_EVENT_ENDED                         0x00000080
#define HWMCA_EVENT_HARDWARE_MESSAGE              0x00000100
#define HWMCA_EVENT_OPSYS_MESSAGE                 0x00000200
#define HWMCA_EVENT_NO_REFRESH_MESSAGE            0x00000400
#define HWMCA_EVENT_STARTED                       0x00000800
#define HWMCA_EVENT_HARDWARE_MESSAGE_DELETE       0x00001000
#define HWMCA_EVENT_SECURITY_EVENT                0x00004000
#define HWMCA_EVENT_CAPACITY_CHANGE               0x00008000
#define HWMCA_EVENT_CAPACITY_RECORD_CHANGE        0x00010000
#define HWMCA_EVENT_DISABLED_WAIT                 0x00040000
```

```
#define HWMCA_EVENT_POWER_CHANGE                    0x00080000
#define HWMCA_EVENT_LOG_EVENT                       0X00100000
#define HWMCA_EVENT_ALL_EVENTS                      0x000DFFFFF
#define HWMCA_DIRECT_INITIALIZE                     0x20000000
#define HWMCA_FORCE_CLIENT_PATH                     0x10000000
#define HWMCA_SNMP_VERSION_2                        0X08000000
#define HWMCA_TOLERATE_LOST_EVENTS                  0X02000000
#define HWMCA_QUALIFIER_SPECIFIED                   0x00800000
#define HWMCA_SNMP_USING_TCP                        0x00400000
#define HWMCA_EVENT_NO_COMMAND_RESPONSE             0x00020000
#define HWMCA_EVENT_KEEP_ALIVE                      0x00200000
```

```
/*****************************************************************************/
/* Defines for the Console Static Object IDs.                                */
/*****************************************************************************/
#define HWMCA_OBJECT_PREFIX             "1.3.6.1.4.1.2.6.42."
#define HWMCA_CONSOLE_ID                "1.3.6.1.4.1.2.6.42.0"   /* .x.x      */
#define HWMCA_CFG_CPC_GROUP_ID          "1.3.6.1.4.1.2.6.42.1"   /* .x.x      */
#define HWMCA_CFG_CPC_ID                "1.3.6.1.4.1.2.6.42.1.0" /* .x.x.*    */
#define HWMCA_CPC_IMAGE_GROUP_ID        "1.3.6.1.4.1.2.6.42.2"   /* .x.x      */
#define HWMCA_CPC_IMAGE_ID              "1.3.6.1.4.1.2.6.42.2.0" /* .x.x.*    */
#define HWMCA_GROUPS_GROUP_ID           "1.3.6.1.4.1.2.6.42.3"   /* .x.x.*    */
#define HWMCA_GROUPS_OBJECT_ID          "1.3.6.1.4.1.2.6.42.3.0" /* .x.x.*.*  */
#define HWMCA_COMMAND_PREFIX            "1.3.6.1.4.1.2.6.42.4."
#define HWMCA_ACT_RESET_OBJECT_ID       "1.3.6.1.4.1.2.6.42.5.0" /* .x.x.*.*  */
#define HWMCA_ACT_IMAGE_OBJECT_ID       "1.3.6.1.4.1.2.6.42.6.0" /* .x.x.*.*  */
#define HWMCA_ACT_LOAD_OBJECT_ID        "1.3.6.1.4.1.2.6.42.7.0" /* .x.x.*.*  */
#define HWMCA_ACT_GROUP_OBJECT_ID       "1.3.6.1.4.1.2.6.42.8.0" /* .x.x.*.*  */
#define HWMCA_CAPACITY_RECORD_OBJECT_ID "1.3.6.1.4.1.2.6.42.9.0"  /* .x.x.*.* */
#define HWMCA_LPAR_GROUP_OBJECT_ID      "1.3.6.1.4.1.2.6.42.13.0" /* x.x.*.*  */
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Object Attribute ID suffix    */
/*****************************************************************************/
#define HWMCA_COMMAND_OBJECT_ID_SUFFIX      "0.1"
#define HWMCA_COMMAND_CONDITION_CODE_SUFFIX "0.2"
#define HWMCA_COMMAND_LAST_INDICATOR_SUFFIX "0.3"
#define HWMCA_ENDED_REASON_SUFFIX           "0.4"
#define HWMCA_ENDED_COMPONENT_SUFFIX        "0.5"
#define HWMCA_ENDED_TYPE_SUFFIX             "0.6"
#define HWMCA_COMMAND_CORRELATOR_SUFFIX     "0.7"
#define HWMCA_LOG_TYPE_SUFFIX               "0.8"
#define HWMCA_EVENT_ID_SUFFIX               "0.9"
#define HWMCA_NAME_SUFFIX                   "1.0"
#define HWMCA_PARENT_NAME_SUFFIX            "2.0"
#define HWMCA_OPSYS_NAME_SUFFIX             "3.0"
#define HWMCA_OPSYS_TYPE_SUFFIX             "4.0"
#define HWMCA_OPSYS_LEVEL_SUFFIX            "5.0"
#define HWMCA_SYSPLEX_NAME_SUFFIX           "6.0"
#define HWMCA_STATUS_ERROR_SUFFIX           "7.0"
#define HWMCA_BUSY_SUFFIX                   "8.0"
#define HWMCA_MESSAGE_SUFFIX                "9.0"
#define HWMCA_MESSAGE_TYPE_SUFFIX           "9.1"
#define HWMCA_MESSAGE_TEXT_SUFFIX           "9.2"
#define HWMCA_MESSAGE_MSG_ID_SUFFIX         "9.3"
#define HWMCA_MESSAGE_DATE_SUFFIX           "9.4"
#define HWMCA_MESSAGE_TIME_SUFFIX           "9.5"
#define HWMCA_MESSAGE_ALARM_SUFFIX          "9.6"
#define HWMCA_MESSAGE_PRIORITY_SUFFIX       "9.7"
#define HWMCA_MESSAGE_HELD_SUFFIX           "9.8"
#define HWMCA_MESSAGE_PROMPT_TEXT_SUFFIX    "9.9"
#define HWMCA_MESSAGE_OSNAME_TEXT_SUFFIX    "9.10"
#define HWMCA_MESSAGE_REFRESH_SUFFIX        "9.11"
#define HWMCA_MESSAGE_TIMESTAMP             "9.12"
#define HWMCA_MESSAGE_IMAGE_LIST            "9.13"
#define HWMCA_STATUS_SUFFIX                 "10.0"
#define HWMCA_EXPECTED_STATUS_SUFFIX        "11.0"
#define HWMCA_IMLMODE_SUFFIX                "12.0"
#define HWMCA_ACTIVATION_PROFILE_SUFFIX     "13.0"
#define HWMCA_LAST_ACT_PROFILE_SUFFIX       "14.0"
#define HWMCA_IP_ADDRESS_SUFFIX             "15.0"
#define HWMCA_SNA_ADDRESS_SUFFIX            "16.0"
#define HWMCA_MODEL_SUFFIX                  "17.0"
#define HWMCA_TYPE_SUFFIX                   "18.0"
#define HWMCA_MACHINE_SERIAL_SUFFIX         "19.0"
#define HWMCA_CPC_SERIAL_SUFFIX             "20.0"
#define HWMCA_CPC_ID_SUFFIX                 "21.0"
#define HWMCA_OBJECT_TYPE_SUFFIX            "22.0"
#define HWMCA_GROUP_CONTENTS_SUFFIX         "23.0"
#define HWMCA_ACT_RESET_LIST_SUFFIX         "24.0"
```

```
#define HWMCA_ACT_IMAGE_LIST_SUFFIX         "25.0"
#define HWMCA_ACT_LOAD_LIST_SUFFIX          "26.0"
#define HWMCA_ACT_PROFILE_IOCDS_SUFFIX      "27.0"
#define HWMCA_ACT_PROFILE_IPLADDR_SUFFIX    "28.0"
#define HWMCA_ACT_PROFILE_IPLPARM_SUFFIX    "29.0"
#define HWMCA_WEIGHT_SUFFIX                 "30.0"
#define HWMCA_CAPPED_SUFFIX                 "31.0"
#define HWMCA_CBU_INSTALLED                 "32.0"
#define HWMCA_CBU_ACTIVATED                 "33.0"
#define HWMCA_CBU_ACTIVATION_DATE           "34.0"
#define HWMCA_CBU_EXPIRATION_DATE           "35.0"
#define HWMCA_NUMBER_CBU_TEST_LEFT          "36.0"
#define HWMCA_REAL_CBU_ACTIVATION_AVAILABLE "37.0"
#define HWMCA_MINIMUM_WEIGHT_SUFFIX         "38.0"
#define HWMCA_MAXIMUM_WEIGHT_SUFFIX         "39.0"
#define HWMCA_WLM_MANAGED_SUFFIX            "40.0"
#define HWMCA_CURRENT_WEIGHT_SUFFIX         "41.0"
#define HWMCA_CURRENT_CAPPED_SUFFIX         "42.0"
#define HWMCA_WORK_LOAD_UNITS_SUFFIX        "43.0"
#define HWMCA_RESERVE_ID_SUFFIX             "44.0"
#define HWMCA_ALERT_SUFFIX                  "45.0"
#define HWMCA_SERVICE_REQUIRED_SUFFIX       "46.0"
#define HWMCA_DEGRADED_SUFFIX               "47.0"
#define HWMCA_CBU_ENABLED_SUFFIX            "48.0"
#define HWMCA_CLUSTER_NAME_SUFFIX           "49.0"
#define HWMCA_CLUSTER_LIST_SUFFIX           "50.0"
#define HWMCA_PARTITION_ID_SUFFIX           "51.0"
#define HWMCA_ACT_PROFILE_IPLTYPE_SUFFIX    "52.0"
#define HWMCA_ACT_PROFILE_WWPN_SUFFIX       "53.0"
#define HWMCA_ACT_PROFILE_BPS_SUFFIX        "54.0"
#define HWMCA_ACT_PROFILE_LUN_SUFFIX        "55.0"
#define HWMCA_ACT_PROFILE_BRLBA_SUFFIX      "56.0"
#define HWMCA_ACT_PROFILE_OSLOADPARM_SUFFIX "57.0"
#define HWMCA_EVENT_TEXT_SUFFIX             "58.0"
#define HWMCA_EVENT_TIMESTAMP_SUFFIX        "59.0"
#define HWMCA_IFA_WEIGHT_SUFFIX             "60.0"
#define HWMCA_IFA_CAPPED_SUFFIX             "61.0"
#define HWMCA_IFA_MINIMUM_WEIGHT_SUFFIX     "62.0"
#define HWMCA_IFA_MAXIMUM_WEIGHT_SUFFIX     "63.0"
#define HWMCA_IFA_CURRENT_WEIGHT_SUFFIX     "64.0"
#define HWMCA_IFA_CURRENT_CAPPED_SUFFIX     "65.0"
#define HWMCA_IFL_WEIGHT_SUFFIX             "66.0"
#define HWMCA_IFL_CAPPED_SUFFIX             "67.0"
#define HWMCA_IFL_MINIMUM_WEIGHT_SUFFIX     "68.0"
#define HWMCA_IFL_MAXIMUM_WEIGHT_SUFFIX     "69.0"
#define HWMCA_IFL_CURRENT_WEIGHT_SUFFIX     "70.0"
#define HWMCA_IFL_CURRENT_CAPPED_SUFFIX     "71.0"
#define HWMCA_ICF_WEIGHT_SUFFIX             "72.0"
#define HWMCA_ICF_CAPPED_SUFFIX             "73.0"
#define HWMCA_ICF_MINIMUM_WEIGHT_SUFFIX     "74.0"
#define HWMCA_ICF_MAXIMUM_WEIGHT_SUFFIX     "75.0"
#define HWMCA_ICF_CURRENT_WEIGHT_SUFFIX     "76.0"
#define HWMCA_ICF_CURRENT_CAPPED_SUFFIX     "77.0"
#define HWMCA_PROCESSOR_RUNNING_TIME_TYPE   "78.0"
#define HWMCA_PROCESSOR_RUNNING_TIME        "79.0"
#define HWMCA_END_TIMESLICE_IF_WAITSTATE    "80.0"
#define HWMCA_IIP_WEIGHT_SUFFIX             "81.0"
#define HWMCA_IIP_CAPPED_SUFFIX             "82.0"
#define HWMCA_IIP_MINIMUM_WEIGHT_SUFFIX     "83.0"
#define HWMCA_IIP_MAXIMUM_WEIGHT_SUFFIX     "84.0"
#define HWMCA_IIP_CURRENT_WEIGHT_SUFFIX     "85.0"
#define HWMCA_IIP_CURRENT_CAPPED_SUFFIX     "86.0"
#define HWMCA_OOCOD_INSTALLED_SUFFIX        "87.0"
#define HWMCA_OOCOD_ACTIVATED_SUFFIX        "88.0"
#define HWMCA_OOCOD_ENABLED_SUFFIX          "89.0"
#define HWMCA_OOCOD_ACTIVATION_DATE_SUFFIX  "90.0"
#define HWMCA_ACT_GROUP_LIST_SUFFIX         "91.0"
#define HWMCA_ACT_PROFILE_CAPACITY_SUFFIX   "92.0"
#define HWMCA_GROUP_PROFILE_NAME_SUFFIX     "93.0"
```

```
#define HWMCA_ACT_PROFILE_LOAD_AT_ACTIVATION_SUFFIX         "94.0"
#define HWMCA_ACT_PROFILE_CENTRAL_STORAGE_SUFFIX            "95.0"
#define HWMCA_ACT_PROFILE_CENTRAL_STORAGE_RESERVED_SUFFIX   "96.0"
#define HWMCA_ACT_PROFILE_EXPANDED_STORAGE_SUFFIX           "97.0"
#define HWMCA_ACT_PROFILE_EXPANDED_STORAGE_RESERVED_SUFFIX  "98.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_CP_SUFFIX           "99.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_CP_RESERVED_SUFFIX  "100.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IFA_SUFFIX          "101.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IFA_RESERVED_SUFFIX "102.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IFL_SUFFIX          "103.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IFL_RESERVED_SUFFIX "104.0"
```

```
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_ICF_SUFFIX          "105.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_ICF_RESERVED_SUFFIX "106.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IIP_SUFFIX          "107.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_IIP_RESERVED_SUFFIX "108.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_CP_SUFFIX              "109.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_CP_RESERVED_SUFFIX     "110.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IFA_SUFFIX             "111.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IFA_RESERVED_SUFFIX    "112.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IFL_SUFFIX             "113.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IFL_RESERVED_SUFFIX    "114.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_ICF_SUFFIX             "115.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_ICF_RESERVED_SUFFIX    "116.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IIP_SUFFIX             "117.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_IIP_RESERVED_SUFFIX    "118.0"
#define HWMCA_CAPACITY_RECORD_LIST_SUFFIX                   "119.0"
#define HWMCA_PERM_SOFTWARE_MODEL_SUFFIX                    "120.0"
#define HWMCA_PERMBILL_SOFTWARE_MODEL_SUFFIX                "121.0"
#define HWMCA_PERMALL_SOFTWARE_MODEL_SUFFIX                 "122.0"
#define HWMCA_PERM_MSU_SUFFIX                               "123.0"
#define HWMCA_PERMBILL_MSU_SUFFIX                           "124.0"
#define HWMCA_PERMALL_MSU_SUFFIX                            "125.0"
#define HWMCA_GEN_PROCESSOR_NUM_SUFFIX                      "126.0"
#define HWMCA_SAP_PROCESSOR_NUM_SUFFIX                      "127.0"
#define HWMCA_IFA_PROCESSOR_NUM_SUFFIX                      "128.0"
#define HWMCA_IFL_PROCESSOR_NUM_SUFFIX                      "129.0"
#define HWMCA_ICF_PROCESSOR_NUM_SUFFIX                      "130.0"
#define HWMCA_IIP_PROCESSOR_NUM_SUFFIX                      "131.0"
#define HWMCA_DEFECTIVE_PROCESSOR_NUM_SUFFIX                "132.0"
#define HWMCA_SPARE_PROCESSOR_NUM_SUFFIX                    "133.0"
#define HWMCA_PENDING_PROCESSOR_NUM_SUFFIX                  "134.0"
#define HWMCA_RECORD_ID_SUFFIX                              "135.0"
#define HWMCA_RECORD_TYPE_SUFFIX                            "136.0"
#define HWMCA_RECORD_ACTIVATION_STATUS_SUFFIX               "137.0"
#define HWMCA_RECORD_ACTIVATION_DATE_SUFFIX                 "138.0"
#define HWMCA_RECORD_EXPIRE_DATE_SUFFIX                     "139.0"
#define HWMCA_RECORD_ACT_EXPIRE_DATE_SUFFIX                 "140.0"
#define HWMCA_RECORD_MAX_REAL_ACT_DAYS_SUFFIX               "141.0"
#define HWMCA_RECORD_MAX_TEST_ACT_DAYS_SUFFIX               "142.0"
#define HWMCA_RECORD_REM_REAL_ACT_DAYS_SUFFIX               "143.0"
#define HWMCA_RECORD_REM_TEST_ACT_DAYS_SUFFIX               "144.0"
#define HWMCA_CAPACITY_CHANGE_TYPE_SUFFIX                   "145.0"
#define HWMCA_RECORD_CHANGE_TYPE_SUFFIX                     "146.0"
#define HWMCA_RECORD_REM_REAL_COUNT_SUFFIX                  "147.0"
#define HWMCA_RECORD_REM_TEST_COUNT_SUFFIX                  "148.0"
#define HWMCA_CAPACITY_CHANGE_ALLOWED_SUFFIX                "149.0"
#define HWMCA_PSW_SUFFIX                                    "150.0"
#define HWMCA_PROCESSOR_SUFFIX                              "150.1"
#define HWMCA_SCP_INITIATE_RESET_SUFFIX                     "150.2"
#define HWMCA_VERSION_SUFFIX                                "151.0"
#define HWMCA_POWER_VERSION_INFO_SUFFIX                     "152.0"
#define HWMCA_POWER_BUFFER_TAG_SUFFIX                       "153.0"
#define HWMCA_POWER_STATUS_REGISTER_SUFFIX                  "154.0"
#define HWMCA_POWER_EVENT_REGISTER_SUFFIX                   "155.0"
#define HWMCA_POWER_ERROR_REGISTER_SUFFIX                   "156.0"
#define HWMCA_POWER_EXHAUST_HEAT_INDEX_SUFFIX               "157.0"
#define HWMCA_POWER_INLET_TEMP_SUFFIX                       "158.0"
#define HWMCA_POWER_AVG_POWER_SAMPLES_SUFFIX                "159.0"
#define HWMCA_POWER_PEAK_POWER_SAMPLES_SUFFIX               "160.0"
#define HWMCA_ALL_IP_ADDRESSES_SUFFIX                       "161.0"
#define HWMCA_EC_MCL_INFO_SUFFIX                            "162.0"
#define HWMCA_AUTO_SWITCH_ENABLED_SUFFIX                    "163.0"
#define HWMCA_IPL_TOKEN_SUFFIX                              "164.0"
#define HWMCA_SYSPLEX_TIME_STP_INFO_SUFFIX                  "165.0"
#define HWMCA_ACT_PROFILE_STORESTATUS_SUFFIX                "166.0"
#define HWMCA_ACT_PROFILE_LOADTYPE_SUFFIX                   "167.0"
#define HWMCA_CPU_COUNTER_BASIC_ENABLED_SUFFIX              "168.0"
#define HWMCA_CPU_COUNTER_PROBLEMSTATE_ENABLED_SUFFIX       "169.0"
#define HWMCA_CPU_COUNTER_CRYPTOACTIVITY_ENABLED_SUFFIX     "170.0"
#define HWMCA_CPU_COUNTER_EXTENDED_ENABLED_SUFFIX           "171.0"
#define HWMCA_CPU_COUNTER_COPROCGROUP_ENABLED_SUFFIX        "172.0"
#define HWMCA_CPU_SAMPLING_BASIC_ENABLED_SUFFIX             "173.0"
#define HWMCA_CPU_SAMPLING_DIAGNOSIS_ENABLED_SUFFIX         "174.0"
#define HWMCA_PENDING_GEN_PROCESSOR_NUM_SUFFIX              "175.0"
#define HWMCA_PENDING_SAP_PROCESSOR_NUM_SUFFIX              "176.0"
#define HWMCA_PENDING_IFA_PROCESSOR_NUM_SUFFIX              "177.0"
#define HWMCA_PENDING_IFL_PROCESSOR_NUM_SUFFIX              "178.0"
#define HWMCA_PENDING_ICF_PROCESSOR_NUM_SUFFIX              "179.0"
#define HWMCA_PENDING_IIP_PROCESSOR_NUM_SUFFIX              "180.0"
#define HWMCA_ZBX_CHASSIS_LIST_SUFFIX                       "181.0"
#define HWMCA_POWER_BUFFER_SIZE_SUFFIX                      "182.0"
#define HWMCA_ENCRYPT_AES_FUNCTIONS_SUFFIX                  "183.0"
#define HWMCA_ENCRYPT_DEA_FUNCTIONS_SUFFIX                  "184.0"
```

```
#define HWMCA_LABEL_POWER_SUFFIX                          "185.0"
#define HWMCA_POWER_SAMPLE_RATE_SUFFIX                    "186.0"
#define HWMCA_GROUP_PROFILE_CAPACITY_SUFFIX               "192.0"
#define HWMCA_LAST_USED_LOAD_ADDR_SUFFIX                  "201.0"
#define HWMCA_LAST_USED_LOAD_PARM_SUFFIX                  "202.0"
#define HWMCA_DESCRIPTION_SUFFIX                          "203.0"
#define HWMCA_OPERATING_MODE_SUFFIX                       "204.0"
#define HWMCA_CLOCK_TYPE_SUFFIX                           "205.0"
#define HWMCA_TIME_OFFSET_DAYS_SUFFIX                     "206.0"
#define HWMCA_TIME_OFFSET_HOURS_SUFFIX                    "207.0"
#define HWMCA_TIME_OFFSET_MINUTES_SUFFIX                  "208.0"
#define HWMCA_TIME_OFFSET_INCREASE_SUFFIX                 "209.0"
#define HWMCA_LICCC_VALIDATION_ENABLED_SUFFIX             "210.0"
#define HWMCA_GLOBAL_PERFORMANCE_DATA_CONTROL_SUFFIX      "211.0"
#define HWMCA_IO_CONFIGURATION_CONTROL_SUFFIX             "212.0"
#define HWMCA_CROSS_PARTITION_AUTHORITY_SUFFIX            "213.0"
#define HWMCA_LOGICAL_PARTITION_ISOLATION_SUFFIX          "214.0"
#define HWMCA_ABS_CAPPED_SUFFIX                           "217.0"
#define HWMCA_ABS_CAP_VALUE_SUFFIX                        "218.0"
#define HWMCA_IFA_ABS_CAPPED_SUFFIX                       "219.0"
#define HWMCA_IFA_ABS_CAP_VALUE_SUFFIX                    "220.0"
#define HWMCA_IFL_ABS_CAPPED_SUFFIX                       "221.0"
#define HWMCA_IFL_ABS_CAP_VALUE_SUFFIX                    "222.0"
#define HWMCA_ICF_ABS_CAPPED_SUFFIX                       "223.0"
#define HWMCA_ICF_ABS_CAP_VALUE_SUFFIX                    "224.0"
#define HWMCA_IIP_ABS_CAPPED_SUFFIX                       "225.0"
#define HWMCA_IIP_ABS_CAP_VALUE_SUFFIX                    "226.0"
#define HWMCA_GROUP_PROFILE_ABS_CAPPED_SUFFIX             "227.0"
#define HWMCA_GROUP_PROFILE_ABS_CAP_VALUE_SUFFIX          "228.0"
#define HWMCA_GROUP_PROFILE_ICF_ABS_CAPPED_SUFFIX         "229.0"
#define HWMCA_GROUP_PROFILE_ICF_ABS_CAP_VALUE_SUFFIX      "230.0"
#define HWMCA_GROUP_PROFILE_IFL_ABS_CAPPED_SUFFIX         "231.0"
#define HWMCA_GROUP_PROFILE_IFL_ABS_CAP_VALUE_SUFFIX      "232.0"
#define HWMCA_GROUP_PROFILE_IIP_ABS_CAPPED_SUFFIX         "233.0"
#define HWMCA_GROUP_PROFILE_IIP_ABS_CAP_VALUE_SUFFIX      "234.0"
#define HWMCA_PROCESSOR_INFORMATION_SUFFIX                "235.0"
#define HWMCA_LPAR_CONTROLS_SUFFIX                        "236.0"
#define HWMCA_GROUP_CONTROLS_SUFFIX                       "237.0"
#define HWMCA_STORAGE_TOTAL_INSTALLED_SUFFIX              "238.0"
#define HWMCA_STORAGE_HARDWARESYSTEM_AREA_SUFFIX          "239.0"
#define HWMCA_STORAGE_CUSTOMER_SUFFIX                     "240.0"
#define HWMCA_STORAGE_CUSTOMERCENTRAL_SUFFIX              "241.0"
#define HWMCA_STORAGE_CUSTOMEREXPANDED_SUFFIX             "242.0"
#define HWMCA_STORAGE_CUSTOMERAVAILABLE_SUFFIX            "243.0"
#define HWMCA_STORAGE_VFM_TOTAL_SUFFIX                    "244.0"
#define HWMCA_STORAGE_VFM_INCREMENT_SIZE_SUFFIX           "245.0"
#define HWMCA_INITIAL_VFM_STORAGE_SUFFIX                  "246.0"
#define HWMCA_MAXIMUM_VFM_STORAGE_SUFFIX                  "247.0"
#define HWMCA_CURRENT_VFM_STORAGE_SUFFIX                  "248.0"
#define HWMCA_STORAGE_CENTRAL_ORIGIN_SUFFIX               "249.0"
#define HWMCA_STORAGE_CENTRAL_INITIAL_SUFFIX              "250.0"
#define HWMCA_STORAGE_CENTRAL_CURRENT_SUFFIX              "251.0"
#define HWMCA_STORAGE_CENTRAL_MAXIMUM_SUFFIX              "252.0"
#define HWMCA_STORAGE_CENTRAL_GAP_SUFFIX                  "253.0"
#define HWMCA_STORAGE_INITIAL_ORIGIN_SUFFIX               "254.0"
#define HWMCA_STORAGE_INITIAL_CURRENT_SUFFIX              "255.0"
#define HWMCA_STORAGE_INITIAL_GAP_SUFFIX                  "256.0"
#define HWMCA_STORAGE_RESERVED_ORIGIN_SUFFIX              "257.0"
#define HWMCA_STORAGE_RESERVED_CURRENT_SUFFIX             "258.0"
#define HWMCA_STORAGE_RESERVED_GAP_SUFFIX                 "259.0"
#define HWMCA_STORAGE_EXPANDED_ORIGIN_SUFFIX              "260.0"
#define HWMCA_STORAGE_EXPANDED_INITIAL_SUFFIX             "261.0"
#define HWMCA_STORAGE_EXPANDED_CURRENT_SUFFIX             "262.0"
#define HWMCA_STORAGE_EXPANDED_MAXIMUM_SUFFIX             "263.0"
#define HWMCA_STORAGE_EXPANDED_GAP_SUFFIX                 "264.0"
#define HWMCA_ACT_PROFILE_INITIAL_VFM_STORAGE_SUFFIX      "265.0"
#define HWMCA_ACT_PROFILE_MAXIMUM_VFM_STORAGE_SUFFIX      "266.0"
#define HWMCA_STORAGE_INFO_SUFFIX                         "267.0"
#define HWMCA_ACT_PROFILE_SSC_HOST_NAME                   "269.0"
#define HWMCA_ACT_PROFILE_SSC_MASTER_USERID               "270.0"
#define HWMCA_ACT_PROFILE_SSC_MASTER_PW                   "271.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_COUNT               "272.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_CHPID               "273.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_ADDRESS_TYPE        "274.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_VLAN_ID             "275.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_STATIC_IP_INFO      "276.0"
#define HWMCA_ACT_PROFILE_SSC_NETWORK_PORT_NUMBER         "277.0"
#define HWMCA_ACT_PROFILE_SSC_IPV4_GATEWAY_COUNT          "278.0"
#define HWMCA_ACT_PROFILE_SSC_IPV4_GATEWAY_INFO           "279.0"
#define HWMCA_ACT_PROFILE_SSC_IPV6_GATEWAY_COUNT          "280.0"
#define HWMCA_ACT_PROFILE_SSC_IPV6_GATEWAY_INFO           "281.0"
#define HWMCA_ACT_PROFILE_SSC_DNS_COUNT                   "282.0"
```

```
#define HWMCA_ACT_PROFILE_SSC_DNS_INFO                        "283.0"
#define HWMCA_LPAR_CRYPTO_CONTROL_DOMAIN_INDEX                "284.0"
#define HWMCA_LPAR_CRYPTO_USAGE_DOMAIN_INDEX                  "285.0"
#define HWMCA_LPAR_CRYPTO_CANDIDATE_LIST                      "286.0"
#define HWMCA_LPAR_CRYPTO_ONLINE_LIST                         "287.0"
#define HWMCA_LPAR_CRYPTO_CONTROL_DOMAIN_COUNT                "288.0"
#define HWMCA_LPAR_CRYPTO_USAGE_DOMAIN_COUNT                  "289.0"
#define HWMCA_LPAR_CRYPTO_CANDIDATE_LIST_COUNT                "290.0"
#define HWMCA_LPAR_CRYPTO_ONLINE_LIST_COUNT                   "291.0"
#define HWMCA_ACT_PROFILE_SSC_BOOT_SELECTION                  "292.0"
#define HWMCA_CBP_PROCESSOR_NUM_SUFFIX                        "293.0"
#define HWMCA_PENDING_CBP_PROCESSOR_NUM_SUFFIX                "294.0"
#define HWMCA_CBP_WEIGHT_SUFFIX                               "295.0"
#define HWMCA_CBP_CAPPED_SUFFIX                               "296.0"
#define HWMCA_CBP_MINIMUM_WEIGHT_SUFFIX                       "297.0"
#define HWMCA_CBP_MAXIMUM_WEIGHT_SUFFIX                       "298.0"
#define HWMCA_CBP_CURRENT_WEIGHT_SUFFIX                       "299.0"
#define HWMCA_CBP_CURRENT_CAPPED_SUFFIX =                     "300.0"
#define HWMCA_CBP_ABS_CAPPED_SUFFIX                           "301.0"
#define HWMCA_CBP_ABS_CAP_VALUE_SUFFIX                        "302.0"
#define HWMCA_GROUP_PROFILE_CBP_ABS_CAPPED_SUFFIX             "303.0"
#define HWMCA_GROUP_PROFILE_CBP_ABS_CAP_VALUE_SUFFIX          "304.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_CBP_SUFFIX            "305.0"
#define HWMCA_ACT_PROFILE_NUM_DEDICATED_CBP_RESERVED_SUFFIX   "306.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_CBP_SUFFIX               "307.0"
#define HWMCA_ACT_PROFILE_NUM_SHARED_CBP_RESERVED_SUFFIX      "308.0"
#define HWMCA_GROUP_PROFILE_MEMBERS_SUFFIX                    "309.0"
#define HWMCA_LPAR_GROUP_LIST_SUFFIX                          "310.0"
#define HWMCA_ENCRYPT_ECC_FUNCTIONS_SUFFIX                    "311.0"
#define HWMCA_SUBCAPACITY_BOOST_ACTIVE_SUFFIX                 "312.0"
#define HWMCA_ZIIP_CAPACITY_BOOST_ACTIVE_SUFFIX               "313.0"
#define HWMCA_SECURE_EXECUTION_SUFFIX                         "314.0"
#define HWMCA_VERIFY_SOFTWARE_SIGNATURE_SUFFIX                "315.0"
#define HWMCA_LAST_USED_VERIFY_SOFTWARE_SIGNATURE_SUFFIX      "316.0"
#define HWMCA_RECORD_MAX_REAL_ACT_HOURS_SUFFIX                "317.0"
#define HWMCA_RECORD_REM_REAL_ACT_HOURS_SUFFIX                "318.0"
#define HWMCA_RECORD_MAX_TEST_ACT_HOURS_SUFFIX                "319.0"
#define HWMCA_RECORD_REM_TEST_ACT_HOURS_SUFFIX                "320.0"
```

```
/*****************************************************************************/
/* Defines for the Console Command Object IDs.                               */
/*****************************************************************************/
#define HWMCA_ACTIVATE_COMMAND                          "1.3.6.1.4.1.2.6.42.4.1"
#define HWMCA_DEACTIVATE_COMMAND                        "1.3.6.1.4.1.2.6.42.4.2"
#define HWMCA_SEND_OPSYS_COMMAND                        "1.3.6.1.4.1.2.6.42.4.3"
#define HWMCA_RESETNORMAL_COMMAND                       "1.3.6.1.4.1.2.6.42.4.4"
#define HWMCA_START_COMMAND                             "1.3.6.1.4.1.2.6.42.4.5"
#define HWMCA_STOP_COMMAND                              "1.3.6.1.4.1.2.6.42.4.6"
#define HWMCA_PSWRESTART_COMMAND                        "1.3.6.1.4.1.2.6.42.4.7"
#define HWMCA_INITIALIZE_API                            "1.3.6.1.4.1.2.6.42.4.8"
#define HWMCA_TERMINATE_API                             "1.3.6.1.4.1.2.6.42.4.9"
#define HWMCA_LOAD_COMMAND                             "1.3.6.1.4.1.2.6.42.4.10"
#define HWMCA_HW_MESSAGE_REFRESH_COMMAND               "1.3.6.1.4.1.2.6.42.4.11"
#define HWMCA_RESETCLEAR_COMMAND                       "1.3.6.1.4.1.2.6.42.4.12"
#define HWMCA_HW_MESSAGE_DELETE_COMMAND                "1.3.6.1.4.1.2.6.42.4.13"
#define HWMCA_ACTIVATE_CBU_COMMAND                     "1.3.6.1.4.1.2.6.42.4.14"
#define HWMCA_UNDO_CBU_COMMAND                         "1.3.6.1.4.1.2.6.42.4.15"
#define HWMCA_IMPORT_PROFILE_COMMAND                   "1.3.6.1.4.1.2.6.42.4.16"
#define HWMCA_EXPORT_PROFILE_COMMAND                   "1.3.6.1.4.1.2.6.42.4.17"
#define HWMCA_RESERVE_COMMAND                          "1.3.6.1.4.1.2.6.42.4.18"
#define HWMCA_EXTERNAL_INTERRUPT_COMMAND               "1.3.6.1.4.1.2.6.42.4.19"
#define HWMCA_SCSI_LOAD_COMMAND                        "1.3.6.1.4.1.2.6.42.4.20"
#define HWMCA_SCSI_DUMP_COMMAND                        "1.3.6.1.4.1.2.6.42.4.21"
#define HWMCA_SHUTDOWN_RESTART_COMMAND                 "1.3.6.1.4.1.2.6.42.4.22"
#define HWMCA_ACTIVATE_OOCOD_COMMAND                   "1.3.6.1.4.1.2.6.42.4.23"
#define HWMCA_UNDO_OOCOD_COMMAND                       "1.3.6.1.4.1.2.6.42.4.24"
#define HWMCA_ADD_CAPACITY_COMMAND                     "1.3.6.1.4.1.2.6.42.4.25"
#define HWMCA_REMOVE_CAPACITY_COMMAND                  "1.3.6.1.4.1.2.6.42.4.26"
#define HWMCA_SYSPLEX_TIME_SWAP_CTS_COMMAND            "1.3.6.1.4.1.2.6.42.4.27"
#define HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND      "1.3.6.1.4.1.2.6.42.4.28"
#define HWMCA_SYSPLEX_TIME_CHANGE_STP_ONLY_CTN_COMMAND "1.3.6.1.4.1.2.6.42.4.29"
#define HWMCA_SYSPLEX_TIME_JOIN_STP_ONLY_CTN_COMMAND   "1.3.6.1.4.1.2.6.42.4.30"
#define HWMCA_SYSPLEX_TIME_LEAVE_STP_ONLY_CTN_COMMAND  "1.3.6.1.4.1.2.6.42.4.31"
#define HWMCA_SYSPLEX_TIME_SET_DST_COMMAND             "1.3.6.1.4.1.2.6.42.4.33"
#define HWMCA_LOAD_FROM_CDROM_COMMAND                  "1.3.6.1.4.1.2.6.42.4.99"
#define HWMCA_ACTIVATE_COMMAND_SUFFIX                           "1"
#define HWMCA_DEACTIVATE_COMMAND_SUFFIX                         "2"
#define HWMCA_SEND_OPSYS_COMMAND_SUFFIX                         "3"
#define HWMCA_RESETNORMAL_COMMAND_SUFFIX                        "4"
#define HWMCA_START_COMMAND_SUFFIX                              "5"
#define HWMCA_STOP_COMMAND_SUFFIX                               "6"
```

```
#define HWMCA_PSWRESTART_COMMAND_SUFFIX                          "7"
#define HWMCA_INITIALIZE_API_SUFFIX                              "8"
#define HWMCA_TERMINATE_API_SUFFIX                               "9"
#define HWMCA_LOAD_COMMAND_SUFFIX                               "10"
#define HWMCA_HW_MESSAGE_REFRESH_COMMAND_SUFFIX                 "11"
#define HWMCA_RESETCLEAR_COMMAND_SUFFIX                         "12"
#define HWMCA_HW_MESSAGE_DELETE_COMMAND_SUFFIX                  "13"
#define HWMCA_ACTIVATE_CBU_COMMAND SUFFIX                       "14"
#define HWMCA_UNDO_CBU_COMMAND SUFFIX                           "15"
#define HWMCA_IMPORT_PROFILE_COMMAND_SUFFIX                     "16"
#define HWMCA_EXPORT_PROFILE_COMMAND_SUFFIX                     "17"
#define HWMCA_RESERVE_COMMAND_SUFFIX                            "18"
#define HWMCA_EXTERNAL_INTERRUPT_COMMAND_SUFFIX                 "19"
#define HWMCA_SCSI_LOAD_COMMAND_SUFFIX                          "20"
#define HWMCA_SCSI_DUMP_COMMAND_SUFFIX                          "21"
#define HWMCA_SHUTDOWN_RESTART_COMMAND_SUFFIX                   "22"
#define HWMCA_ACTIVATE_OOCOD_COMMAND_SUFFIX                     "23"
#define HWMCA_UNDO_OOCOD_COMMAND_SUFFIX                         "24"
#define HWMCA_ADD_CAPACITY_COMMAND_SUFFIX                       "25"
#define HWMCA_REMOVE_CAPACITY_COMMAND_SUFFIX                    "26"
#define HWMCA_SYSPLEX_TIME_SWAP_CTS_COMMAND_SUFFIX              "27"
#define HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND_SUFFIX        "28"
#define HWMCA_SYSPLEX_TIME_CHANGE_STP_ONLY_CTN_COMMAND_SUFFIX   "29"
#define HWMCA_SYSPLEX_TIME_JOIN_STP_ONLY_CTN_COMMAND_SUFFIX     "30"
#define HWMCA_SYSPLEX_TIME_LEAVE_STP_ONLY_CTN_COMMAND_SUFFIX    "31"
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Message Event Types.          */
/*****************************************************************************/
#define HWMCA_HARDWARE_MESSAGE         1
#define HWMCA_OPSYS_MESSAGE            2
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Log Event Types               */
/*****************************************************************************/
#define HWMCA_CONSOLE_LOG              0
#define HWMCA_SECURITY_LOG             1
#define HWMCA_AUDIT_LOG                2
```

```
/*****************************************************************************/
/* Defines for the CPC Managed Object Degraded Indicator                     */
/*****************************************************************************/
#define HWMCA_NOT_DEGRADED             0x0000
#define HWMCA_DEGRADED_MEM             0x0001
#define HWMCA_DEGRADED_MBA             0x0002
#define HWMCA_DEGRADED_NODE            0x0004
#define HWMCA_DEGRADED_RING            0x0008
#define HWMCA_DEGRADED_CBU             0x0010
#define HWMCA_DEGRADED_MRU             0x0020
#define HWMCA_DEGRADED_AMBIENT         0x0040
#define HWMCA_DEGRADED_MRU_IML         0x0080
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Status Values.                */
/*****************************************************************************/
#define HWMCA_STATUS_OPERATING           0x00000001
#define HWMCA_STATUS_NOT_OPERATING       0x00000002
#define HWMCA_STATUS_NO_POWER            0x00000004
#define HWMCA_STATUS_NOT_ACTIVATED       0x00000008
#define HWMCA_STATUS_EXCEPTIONS          0x00000010
#define HWMCA_STATUS_STATUS_CHECK        0x00000020
#define HWMCA_STATUS_SERVICE             0x00000040
#define HWMCA_STATUS_LINKNOTACTIVE       0x00000080
#define HWMCA_STATUS_POWERSAVE           0x00000100
#define HWMCA_STATUS_SERIOUSALERT        0x00000200
#define HWMCA_STATUS_ALERT               0x00000400
#define HWMCA_STATUS_ENVALERT            0x00000800
#define HWMCA_STATUS_SERVICE_REQ         0x00001000
#define HWMCA_STATUS_DEGRADED            0x00002000
#define HWMCA_STATUS_STORAGE_EXCEEDED    0x01000000
#define HWMCA_STATUS_LOGOFF_TIMEOUT      0x02000000
#define HWMCA_STATUS_FORCED_SLEEP        0x04000000
#define HWMCA_STATUS_IMAGE_NOT_OPERATING 0x08000000
#define HWMCA_STATUS_IMAGE_NOT_ACTIVATED 0x10000000
#define HWMCA_STATUS_IMAGE_NOT_CAPABLE   0x20000000
#define HWMCA_STATUS_UNKNOWN             0x40000000
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console IML Mode Values.              */
/*****************************************************************************/
#define HWMCA_IML_ESA390_MODE           1
#define HWMCA_IML_S370_MODE             2
#define HWMCA_IML_FM_MODE               6
#define HWMCA_IML_FMAE_MODE             7
#define HWMCA_IML_HM_MODE               8
#define HWMCA_IML_HMEA_MODE             9
#define HWMCA_IML_HMEX_MODE            10
#define HWMCA_IML_LPAR_MODE            11
#define HWMCA_IML_ESA390TPF_MODE       12
#define HWMCA_IML_CF_PROD_MODE         13
#define HWMCA_IML_FMEX_MODE            14
#define HWMCA_IML_HMAS_MODE            15
#define HWMCA_IML_LINUXO_MODE          16
#define HWMCA_IML_ZVM_MODE             18
#define HWMCA_IML_ZAWARE_MODE          20
#define HWMCA_IML_ZACI_MODE            21
```

```
/*****************************************************************************/
/* Defines for the Image Activation Profile Operating Mode Values.           */
/*****************************************************************************/
#define HWMCA_GENERAL_OPERATING_MODE          1
#define HWMCA_ESA390_OPERATING_MODE           1
#define HWMCA_ESA390TPF_OPERATING_MODE        2
#define HWMCA_CF_OPERATING_MODE               3
#define HWMCA_LINUX_OPERATING_MODE            4
#define HWMCA_FMEX_OPERATING_MODE             5
#define HWMCA_HMEX_OPERATING_MODE             6
#define HWMCA_HMAS_OPERATING_MODE             7
#define HWMCA_ZVM_OPERATING_MODE              8
#define HWMCA_ZAWARE_OPERATING_MODE           9
#define HWMCA_ZACI_OPERATING_MODE             9
```

```
/*****************************************************************************/
/*Defines for the Hardware Management Console Image Profile Clock Type Values.*/
/*****************************************************************************/
#define HWMCA_CLOCK_TYPE_STANDARD 0
#define HWMCA_CLOCK_TYPE_LPAR     1
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Image Profile SSC Network     */
/* Address type values.                                                      */
/*****************************************************************************/
#define HWMCA_ADDRESS_TYPE_UNDEFINED 0
#define HWMCA_ADDRESS_TYPE_DHCP      1
#define HWMCA_ADDRESS_TYPE_STATIC    2
#define HWMCA_ADDRESS_TYPE_LINKLOCAL 3
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console IPL Type Values.              */
/*****************************************************************************/
#define HWMCA_IPLTYPE_STANDARD          1
#define HWMCA_IPLTYPE_SCSI              2
#define HWMCA_IPLTYPE_SCSIDUMP          3
#define HWMCA_IPLTYPE_NVME              4
#define HWMCA_IPLTYPE_NVMEDUMP          5
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Load Type Values.             */
/*****************************************************************************/
#define HWMCA_LOADTYPE_NORMAL            0
#define HWMCA_LOADTYPE_CLEAR             1
```

```
/*****************************************************************************/
/* Defines for the Hardware Management Console Boot Selection Type Values.   */
/*****************************************************************************/
#define HWMCA_BC_INSTALLER               1
#define HWMCA_BC_APPLIANCE               2
```

```
/*****************************************************************************/
/* Defines for the Console Object Type Values.                               */
/*****************************************************************************/
```

```
#define HWMCA_CPC_GROUP              1
#define HWMCA_CPC_IMAGE_GROUP        2
#define HWMCA_CPC_USER_GROUP         3
#define HWMCA_CPC_IMAGE_USER_GROUP   4
#define HWMCA_CPC_OBJECT             5
#define HWMCA_CPC_IMAGE_OBJECT       6
#define HWMCA_CF_OBJECT              7
#define HWMCA_ACT_PROFILE_RESET      8
#define HWMCA_ACT_PROFILE_IMAGE      9
#define HWMCA_ACT_PROFILE_LOAD       10
#define HWMCA_ACT_PROFILE_GROUP      11
#define HWMCA_CAPACITY_RECORD        12
#define HWMCA_LPAR_GROUP             18

/*****************************************************************************/
/* Defines for the Hardware Management Console Shutdown/Restart Types.       */
/*****************************************************************************/
#define HWMCA_RESTART_APPLICATION           1
#define HWMCA_RESTART_CONSOLE               2
#define HWMCA_SHUTDOWN_CONSOLE              3
#define HWMCA_RESTART_APPLICATION_ALTERNATE  4
#define HWMCA_RESTART_CONSOLE_ALTERNATE      5
#define HWMCA_SHUTDOWN_CONSOLE_ALTERNATE     6

/*****************************************************************************/
/* Defines for the Hardware Management Console Ended Event Reasons.          */
/*****************************************************************************/
#define HWMCA_ENDED_USER                    1
#define HWMCA_ENDED_AUTOMATION              2
#define HWMCA_ENDED_OTHER                   3

/*****************************************************************************/
/* Defines for the Hardware Management Console Processor Running Time types.  */
/*****************************************************************************/
#define HWMCA_DETERMINED_SYSTEM             0
#define HWMCA_DETERMINED_USER               1

/*****************************************************************************/
/* Defines for the type of capacity record.                                  */
/*****************************************************************************/
#define HWMCA_CAPACITY_RECORD_TYPE_CBU           1
#define HWMCA_CAPACITY_RECORD_TYPE_OOCOD         2
#define HWMCA_CAPACITY_RECORD_TYPE_PLANNED_EVENT 3
#define HWMCA_CAPACITY_RECORD_TYPE_LOANER        4
#define HWMCA_CAPACITY_RECORD_TYPE_CONTAINER     5
```

```
/*****************************************************************************/
/* Defines for the activation status of a capacity record.                   */
/*****************************************************************************/
#define HWMCA_CAPACITY_RECORD_STATUS_NOT_ACTIVATED    1
#define HWMCA_CAPACITY_RECORD_STATUS_REAL             2
#define HWMCA_CAPACITY_RECORD_STATUS_TEST             3
#define HWMCA_CAPACITY_RECORD_STATUS_CAN_BE_ACTIVATED 4

/*****************************************************************************/
/* Defines for the type of change for a HWMCA_EVENT_CAPACITY_CHANGE event.   */
/*****************************************************************************/
#define HWMCA_CAPACITY_FENCED_BOOK                0
#define HWMCA_CAPACITY_DEFECTIVE_PROCESSOR        1
#define HWMCA_CAPACITY_CONCURRENT_BOOK_REPLACE    2
#define HWMCA_CAPACITY_CONCURRENT_BOOK_ADD        3
#define HWMCA_CAPACITY_CHECK_STOP                 4
#define HWMCA_CAPACITY_CHANGES_ALLOWED            5
#define HWMCA_CAPACITY_CHANGES_NOT_ALLOWED        6

/*****************************************************************************/
/* Defines for the type of change for a HWMCA_EVENT_CAPACITY_RECORD_CHANGE   */
/* event.                                                                    */
/*****************************************************************************/
#define HWMCA_CAPACITY_RECORD_ADD         0
#define HWMCA_CAPACITY_RECORD_DELTA       1
#define HWMCA_CAPACITY_RECORD_DELETE      2
#define HWMCA_CAPACITY_RECORD_ACCOUNTING  3
#define HWMCA_CAPACITY_ACTIVATION_LEVEL   4
#define HWMCA_CAPACITY_PRIORITY_PENDING   5
#define HWMCA_CAPACITY_RECORD_OTHER       6
```

```
/*****************************************************************************/
/* Defines for the type of capacity record.                                  */
/*****************************************************************************/
```

```
#define HWMCA_CAPACITY_RECORD_TYPE_CBU           1
#define HWMCA_CAPACITY_RECORD_TYPE_OOCOD         2
#define HWMCA_CAPACITY_RECORD_TYPE_PLANNED_EVENT 3
#define HWMCA_CAPACITY_RECORD_TYPE_LOANER        4

/*****************************************************************************/
/* Defines for the activation status of a capacity record.                   */
/*****************************************************************************/
#define HWMCA_CAPACITY_RECORD_STATUS_NOT_ACTIVATED    1
#define HWMCA_CAPACITY_RECORD_STATUS_REAL             2
#define HWMCA_CAPACITY_RECORD_STATUS_TEST             3
#define HWMCA_CAPACITY_RECORD_STATUS_CAN_BE_ACTIVATED 4
```

```
/*****************************************************************************/
/* Defines for the type of change for a HWMCA_EVENT_CAPACITY_CHANGE event.   */
/*****************************************************************************/
#define HWMCA_CAPACITY_FENCED_BOOK                0
#define HWMCA_CAPACITY_DEFECTIVE_PROCESSOR        1
#define HWMCA_CAPACITY_CONCURRENT_BOOK_REPLACE    2
#define HWMCA_CAPACITY_CONCURRENT_BOOK_ADD        3
#define HWMCA_CAPACITY_CHECK_STOP                 4
#define HWMCA_CAPACITY_CHANGES_ALLOWED            5
#define HWMCA_CAPACITY_CHANGES_NOT_ALLOWED        6

/*****************************************************************************/
/* Defines for the type of change for a HWMCA_EVENT_CAPACITY_RECORD_CHANGE   */
/* event.                                                                    */
/*****************************************************************************/
#define HWMCA_CAPACITY_RECORD_ADD         0
#define HWMCA_CAPACITY_RECORD_DELTA       1
#define HWMCA_CAPACITY_RECORD_DELETE      2
#define HWMCA_CAPACITY_RECORD_ACCOUNTING  3
#define HWMCA_CAPACITY_ACTIVATION_LEVEL   4
#define HWMCA_CAPACITY_PRIORITY_PENDING   5
#define HWMCA_CAPACITY_RECORD_OTHER       6
```

Data exchange APIs SNMP target structure (HWMCA_SNMP_TARGET_T)

```
/*****************************************************************************/
/* Console SNMP Target Structure                                             */
/*****************************************************************************/
struct HWMCA_SNMP_TARGET_S {
   PVOID                pHost;        /* A pointer to a null terminated      */
                                      /* string specifying the host name or  */
                                      /* internet address for the target     */
                                      /* Console.                            */
                                      /*                                     */
   CHAR szCommunity[HWMCA_MAX_COMMUNITY_LEN]; /* Community name to be used    */
                                      /* for requests.                       */
   UINT ulSecurityVersion;            /* Security version used v2c or v3     */
   CHAR szUsername[HWMCA_MAX_USERNAME_LEN]; /* Username to be used for v3 auth */

   CHAR szPassword[HWMCA_MAX_USERNAME_LEN]; /* Password to be used for v3 auth */

   UINT ulReserved;                   /* Reserved field.                      */

};

typedef struct HWMCA_SNMP_TARGET_S HWMCA_SNMP_TARGET_T;
typedef HWMCA_SNMP_TARGET_T  *     HWMCA_SNMP_TARGET_P;
#define HWMCA_SNMP_TARGET_SIZE sizeof(HWMCA_SNMP_TARGET_T)
```

Data exchange APIs initialize structure (HWMCA_INITIALIZE_T)

```
/*****************************************************************************/
/* Console Initialize Structure                                              */
/*****************************************************************************/
struct HWMCA_INITIALIZE_S {
   PVOID                pTarget;      /* Pointer to data specifying the      */
                                      /* target Hardware Management Console  */
                                      /* for the request.                    */
                                      /*                                     */
                                      /* For the SNMP APIs, this is an       */
                                      /* HWMCA_SNMP_TARGET_S structure.      */
                                      /*                                     */
   UINT                 ulEventMask;  /* A mask specifying the event         */
                                      /* notifications that the application  */
                                      /* wants to register for.              */
                                      /*                                     */
                                      /* - HWMCA_EVENT_COMMAND_RESPONSE      */
                                      /* - HWMCA_EVENT_MESSAGE               */
                                      /* - HWMCA_EVENT_STATUS_CHANGE         */
                                      /* - HWMCA_EVENT_NAME_CHANGE           */
                                      /* - HWMCA_EVENT_ACTIVATE_PROF_CHANGE  */
                                      /* - HWMCA_EVENT_CREATED               */
                                      /* - HWMCA_EVENT_DESTROYED             */
                                      /* - HWMCA_EVENT_EXCEPTION_STATE       */
                                      /* - HWMCA_EVENT_ENDED                 */
                                      /* - HWMCA_EVENT_HARDWARE_MESSAGE      */
                                      /* - HWMCA_EVENT_OPSYS_MESSAGE         */
                                      /* - HWMCA_EVENT_NO_REFRESH_MESSAGE    */
                                      /* - HWMCA_EVENT_STARTED               */
                                      /* - HWMCA_EVENT_HARDWARE_MESSAGE_DELETE*/
                                      /* - HWMCA_DIRECT_INITIALIZE           */
                                      /* - HWMCA_FORCE_CLIENT_PATH           */
                                      /* - HWMCA_SNMP_VERSION_2              */
                                      /*                                     */
   ULONG                ulReserved;   /* Must be zero.                       */
   union {
      struct {
         INT             iAgentSocket; /* Socket used to communicate with the */
                                       /* SNMP agent on the target Console.   */
         UINT            ulInetAddr;   /* Internet address for the SNMP agent.*/
         UINT            uiSecVersion;
         CHAR szCommunity[HWMCA_MAX_COMMUNITY_LEN]; /* Community name to be   */
                                       /* used for requests.                  */
         struct {
            unsigned char bAuthEngineId[HWMCA_MAX_ID_LEN]; //
            UINT ulAuthEngineIdLength;
            CHAR szUsername[HWMCA_MAX_USERNAME_LEN];
            CHAR szPassword[HWMCA_MAX_USERNAME_LEN];
            UINT ulAuthEngineBoots;
            UINT ulAuthEngineTime;
            UINT ulMsgId;
            unsigned char bPrivateKey[16];
            UINT uiSalt;
         } v3;
      } snmp;
   } protocol;
};

typedef struct HWMCA_INITIALIZE_S HWMCA_INITIALIZE_T;
typedef HWMCA_INITIALIZE_T  *     HWMCA_INITIALIZE_P;
#define HWMCA_INITIALIZE_SIZE sizeof(HWMCA_INITIALIZE_T)
```

Data exchange APIs datatype structure (HWMCA_DATATYPE_T)

```
/*****************************************************************************/
/* Console Data Type Structure                                               */
/*****************************************************************************/
struct HWMCA_DATATYPE_S {
   UCHAR                ucType;       /* Type of the data:                   */
                                      /*  - HWMCA_TYPE_SEQUENCE              */
                                      /*  - HWMCA_TYPE_INTEGER               */
                                      /*  - HWMCA_TYPE_OCTETSTRING           */
                                      /*  - HWMCA_TYPE_NULL                  */
                                      /*  - HWMCA_TYPE_OBJECTID              */
                                      /*  - HWMCA_TYPE_IPADDRESS             */
   ULONG                ulLength;     /* Length of the data.                 */
   PVOID                pData;        /* Pointer to the data itself.         */
   struct HWMCA_DATATYPE_S *pNext;       /* Pointer to next data type structure */
};

typedef struct HWMCA_DATATYPE_S HWMCA_DATATYPE_T;
typedef HWMCA_DATATYPE_T  *     HWMCA_DATATYPE_P;
#define HWMCA_DATATYPE_SIZE sizeof(HWMCA_DATATYPE_T)
```

Data exchange APIs variable binding structure (HWMCA_VARBIND_T)

```
/*****************************************************************************/
/* Hardware Management Console Variable Binding Structure                    */
/*****************************************************************************/
struct HWMCA_VARBIND_S {
   UCHAR                  ucType;     /* Type of the data:                   */
                                      /*  - HWMCA_TYPE_SEQUENCE              */
                                      /*  - HWMCA_TYPE_INTEGER               */
                                      /*  - HWMCA_TYPE_OCTETSTRING           */
                                      /*  - HWMCA_TYPE_NULL                  */
                                      /*  - HWMCA_TYPE_OBJECTID              */
                                      /*  - HWMCA_TYPE_IPADDRESS             */
   ULONG                  status;     /* Error status value.                 */
   ULONG                  ulLength;   /* Length of the data.                 */
   PVOID                  pData;      /* Pointer to the data itself.         */
   PSZ                    pszOid;     /* Pointer to the object identifier    */
   struct HWMCA_VARBIND_S *pNext;     /* Pointer to next data type structure */
};

typedef struct HWMCA_VARBIND_S HWMCA_VARBIND_T;
typedef HWMCA_VARBIND_T  *     HWMCA_VARBIND_P;
#define HWMCA_VARBIND_SIZE sizeof(HWMCA_VARBIND_T)
```

Data exchange APIs event qualifier structure (HWMCA_EVENT_QUALIFIER_T)

```
/*****************************************************************************/
/* Hardware Management Console Event Qualifier Structure                     */
/*****************************************************************************/
struct HWMCA_EVENT_QUALIFIER_S {
   unsigned int ulEventMask;          /* Event mask for qualifier           */
   unsigned int ulType;               /* Qualifier type                     */

   union {
      char szName[256];               /* Image name for OS msgs events      */
      unsigned int number;            /* Number value for an event          */
      char cReserved[256];            /* Reserved space                     */
   } type;                            /* union of qualifier data            */
   struct HWMCA_EVENT_QUALIFIER_S *pNext;/* Pointer to next qualifier struct  */
};
typedef struct HWMCA_EVENT_QUALIFIER_S HWMCA_EVENT_QUALIFIER_T;
typedef HWMCA_EVENT_QUALIFIER_T  *     HWMCA_EVENT_QUALIFIER_P;
#define HWMCA_EVENT_QUALIFIER_SIZE sizeof(HWMCA_EVENT_QUALIFIER_T)
#define HWMCA_QUALIFIER_TYPE_NAME      0x00000001
#define HWMCA_QUALIFIER_TYPE_NUMBER    0x00000002
#define HWMCA_QUALIFIER_LOG_TYPE       0X00000003
```

Function prototypes

```
/*****************************************************************************/
/* Console Data Exchange Function Prototypes                                 */
/*****************************************************************************/
```

```
extern ULONG EXPENTRY HwmcaInitialize(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */

extern ULONG EXPENTRY HwmcaGet(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       PSZ,                      /* Pointer to null terminated object ID      */
                                 /* string.                                   */
       PVOID,                    /* Pointer to an output buffer for the       */
                                 /* returned data.                            */
       ULONG,                    /* Size of the output buffer.                */
       PULONG,                   /* Pointer to an area where the number of    */
                                 /* total bytes needed for this Get request   */
                                 /* is returned.                              */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */
```

```
extern ULONG EXPENTRY HwmcaGetNext(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       PSZ,                      /* Pointer to null terminated object ID      */
                                 /* string.                                   */
       PVOID,                    /* Pointer to an output buffer for the       */
                                 /* returned data.                            */
       ULONG,                    /* Size of the output buffer.                */
       PULONG,                   /* Pointer to an area where the number of    */
                                 /* total bytes needed for this Get request   */
                                 /* is returned.                              */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */

extern ULONG EXPORTTYPE HwmcaGetBulk(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       HWMCA_DATATYPE_P,         /* Pointer to a linked list of               */
                                 /* HWMCA_DATATYPE_T structures used to       */
                                 /* specify the object IDs to use in the      */
                                 /* GetBulk request.                          */
       UINT,                     /* Count of non-repeaters for the request.   */
       UINT,                     /* Maximum repititions for the request.      */
       PVOID,                    /* Pointer to an output buffer for the       */
                                 /* returned data.                            */
       ULONG,                    /* Size of the output buffer.                */
       PULONG,                   /* Pointer to an area where the number of    */
                                 /* total bytes needed for this Get request   */
                                 /* is returned.                              */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */

extern ULONG EXPORTTYPE HwmcaEnhancedGet(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       HWMCA_VARBIND_P,          /* Pointer to a linked list of               */
                                 /* HWMCA_VARBIND_T structures used to        */
                                 /* specify the object IDs to use in the      */
                                 /* Get request.                              */
       PVOID,                    /* Pointer to an output buffer for the       */
                                 /* returned data.                            */
       ULONG,                    /* Size of the output buffer.                */
       PULONG,                   /* Pointer to an area where the number of    */
                                 /* total bytes needed for this Get request   */
                                 /* is returned.                              */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */
```

```
extern ULONG EXPENTRY HwmcaSet(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
                                 /* structure.                                */
       PSZ,                      /* Pointer to null terminated object ID      */
                                 /* string.                                   */
       HWMCA_DATATYPE_P,         /* Pointer to a linked list of               */
                                 /* HWMCA_DATATYPE_T structures used to       */
                                 /* represent the data.                       */
       ULONG);                   /* Time to wait for the next event           */
                                 /* notification (in milliseconds).           */
extern ULONG EXPORTTYPE HwmcaEnhancedSet(
       HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization   */
```

```
                                /* structure.                                   */
       HWMCA_VARBIND_P,         /* Pointer to a linked list of                  */
                                /* HWMCA_VARBIND_T structures used to           */
                                /* specify the object IDs to use in the        */
                                /* Set request.                                 */
       ULONG);                  /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */
extern ULONG EXPENTRY HwmcaWaitEvent(
       HWMCA_INITIALIZE_P,      /* Pointer to data exchange initialization      */
                                /* structure.                                   */
       PVOID,                   /* Pointer to an output buffer for the          */
                                /* returned data.                               */
       ULONG,                   /* Size of the output buffer.                   */
       PULONG,                  /* Pointer to an area where the number of       */
                                /* total bytes needed for this Get request      */
                                /* is returned.                                 */
       ULONG);                  /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */
extern ULONG EXPENTRY HwmcaTerminate(
       HWMCA_INITIALIZE_P,      /* Pointer to data exchange initialization      */
                                /* structure.                                   */
       ULONG);                  /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */

extern ULONG EXPENTRY HwmcaCommand(
       HWMCA_INITIALIZE_P,      /* Pointer to data exchange initialization      */
                                /* structure.                                   */
       PSZ,                     /* Pointer to null terminated object ID         */
                                /* string that the command target.              */
       PSZ,                     /* Pointer to null terminated object ID         */
                                /* string that command identifier.              */
       HWMCA_DATATYPE_P,        /* Pointer to a linked list of                  */
                                /* HWMCA_DATATYPE_T structures used to          */
                                /* represent the argument data.                 */
       ULONG);                  /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */
extern ULONG EXPORTTYPE HwmcaCorrelatedCommand(
      HWMCA_INITIALIZE_P,       /* Pointer to data exchange initialization      */
                                /* structure.                                   */
      PSZ,                      /* Pointer to null terminated object ID         */
                                /* string that the command target.              */
      PSZ,                      /* Pointer to null terminated object ID         */
                                /* string that command identifier.              */
      HWMCA_DATATYPE_P,         /* Pointer to a linked list of                  */
                                /* HWMCA_DATATYPE_T structures used to          */
                                /* represent the argument data.                 */
      ULONG,                    /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */
      void *,                   /* Pointer to correlator data.                  */
      unsigned int);            /* Size of correlator data.                     */

extern ULONG EXPORTTYPE HwmcaRegister(
       HWMCA_INITIALIZE_P,      /* Pointer to data exchange initialization      */
       UINT,                    /* New event mask to be used                    */

       HWMCA_EVENT_QUALIFIER_P, /* New event qualifiers to be used              */
       ULONG);                  /* Time to wait for the next event              */
                                /* notification (in milliseconds).              */
```

```
extern ULONG EXPENTRY HwmcaBuildId(
       PSZ,                     /* Pointer to a buffer where the built object*/
                                /* identifier string is to be placed.           */
       PSZ,                     /* Pointer to the prefix string to be used      */
                                /* for the object identifier to be built.       */
                                /*  - HWMCA_CONSOLE_ID                          */
                                /*  - HWMCA_CFG_CPC_GROUP_ID                    */
                                /*  - HWMCA_CFG_CPC_ID                          */
                                /*  - HWMCA_CPC_IMAGE_GROUP_ID                  */
                                /*  - HWMCA_CPC_IMAGE_ID                        */
                                /*  - HWMCA_GROUPS_GROUP_ID                     */
                                /*  - HWMCA_COMMAND_PREFIX                      */
                                /*  - HWMCA_ACT_RESET_OBJECT_ID                 */
                                /*  - HWMCA_ACT_IMAGE_OBJECT_ID                 */
                                /*  - HWMCA_ACT_LOAD_OBJECT_ID                  */
       PSZ,                     /* Pointer to the attribute suffix string to */
                                /* be used for the object identifier to be      */
                                /* build.  This can be specified as NULL,       */
                                /* when building an ID for an object itself, */
                                /* as opposed to an attribute object ID.        */
       PSZ,                     /* Pointer to the Group name to be used for  */
                                /* building the object identifier.  This can */
```

```
                                /* be specified as NULL, when building an ID */
                                /* for a predefined group or an object from  */
                                /* a predefined group.                       */
       PSZ);                    /* Pointer to the Object name to be used for */
                                /* building the object identifier.  This can */
                                /* be specified as NULL, when building an ID */
                                /* for a group object.                       */

extern ULONG EXPENTRY HwmcaBuildAttributeId(
       PSZ,                     /* Pointer to a buffer where the built object*/
                                /* identifier string for the attribute is to */
                                /* be placed.                                */
       PSZ,                     /* Pointer to the object identifier for the  */
                                /* object for which the attribute identifier */
                                /* is to be built.                           */
       PSZ);                    /* Pointer to the attribute suffix string to */
                                /* be used for the attribute identifier to be*/
                                /* build.                                    */
```

Data exchange APIs and commands API example

Refer to the following pages for some example code using the Console Data Exchange APIs and Commands API. A copy of this code can be found on Resource Link at http://www.ibm.com/servers/resourcelink. Click **Services**, and then Click **API**.

For more information about the parameters required for this example, simply execute the program with no arguments. This will print out help information to the screen. Some sample invocations for this example program are:

- HWMCATST 1 9.130.1.1 1.3.6.1.4.1.2.6.42.0.23.0

  This will perform a get operation for the *Group Contents* attribute of the Console object.
- HWMCATST 1 9.130.1.1 1.3.6.1.4.1.2.6.42.1.23.0

  Performs a get operation for the *Group Contents* attribute of the Defined CPC Group object.
- HWMCATST 1 9.130.1.1 1.3.6.1.4.1.2.6.42.2.23.0

  Performs a get operation for the *Group Contents* attribute of the CPC Images Group object.
- HWMCATST 1 9.130.1.1 1.3.6.1.4.1.2.6.42.1.0.10.0.3362806951

  Performs a get operation for the *Status* attribute of the Defined CPC object named CPC01.
- HWMCATST 4 9.130.1.1 1.3.6.1.4.1.2.6.42.1.0.3362806951 1.3.6.1.4.1.2.6.42.4.1

  Sends an Activate command request to the Defined CPC object named CPC01.
- HWMCATST 5 9.130.1.1 255 -1

  Waits forever for all types of event notifications.

```
/**********************   Defines        **********************************/
#define INCL_DOS

#define HWMCAAPI_TIMEOUT 30000
#define COMMUNITY "public"

/**********************   Include Files  **********************************/
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <memory.h>
#include "hwmcaapi.h"

/**********************   Function Prototypes  ****************************/
VOID parse_and_print_get(HWMCA_DATATYPE_P);
VOID parse_and_print_get_group_contents(HWMCA_DATATYPE_P);


/**********************   Main  *******************************************/
void main(argc, argv, envp)
   int argc;
```

```
   char *argv[];
   char *envp[];
{
   APIRET              usRc;               /* Local return code from API calls       */
   ULONG               ulLength;           /* Number of bytes passed to an API call  */
   ULONG               ulBytesNeeded;      /* Number of bytes needed for an API call */
   USHORT              usfContinue;        /* Local continue processing flag         */
   HWMCA_DATATYPE_T    tHwmcaDataType;     /* HWMCA DataType structure                 */
   HWMCA_DATATYPE_P    pHwmcaDataType;     /* Ptr to a HWMCA DataType structure        */
   HWMCA_DATATYPE_T    aHwmcaDataType[10]; /* HWMCA DataType structure                 */
   ULONG               aulCmdData[10];     /* array of command integer data          */
   HWMCA_INITIALIZE_T  tHwmcaInitialize;   /* Structure for HwmcaInitialize API call   */
   HWMCA_SNMP_TARGET_T tHwmcaSnmpTarget;   /* Target structure for HwmcaInitialize call*/
   INT                 i, j;               /* loop variables                         */
   CHAR                cEventBuf[HWMCA_MAX_EVENT_BUF_SIZE];
   CHAR                szOID[HWMCA_MAX_ID_LEN];
   PSZ                 pszAttribute, pszGroupName, pszObjectName;

   usfContinue  = TRUE;
   memset(&tHwmcaInitialize,'\0',HWMCA_INITIALIZE_SIZE);
   if (argc >= 4) {  /* Proper number of initial arguments passed */
      switch (atoi(argv[1])) {
         case 1: /* Get request */
            break;
         case 2: /* Get-Next request */
            break;
         case 3: /* Set request */
            if (argc != 6 ) { /* Proper number of arguments passed */
               usfContinue = FALSE;
            } /* endif */
            break;
         case 4: /* Command request */
            if (argc < 5 ) { /* Proper number of arguments passed */
               usfContinue = FALSE;
            } /* endif */
            break;
         case 5: /* WaitEvent request */
            if (argc < 5 ) { /* Proper number of arguments passed */
               usfContinue = FALSE;
            } else {
               tHwmcaInitialize.ulEventMask = (ULONG)atol(argv[3]);
            } /* endif */
            break;
         case 6: /* BuildId request */
            break;
         case 7: /* BuildAttributeId request */
            if (argc < 5 ) { /* Proper number of arguments passed */
               usfContinue = FALSE;
            } /* endif */
            break;
         default:
            usfContinue = FALSE;
            break;
      } /* endswitch */
      if (usfContinue) {
         tHwmcaInitialize.pTarget = &tHwmcaSnmpTarget;
         tHwmcaSnmpTarget.pHost   = argv[2];
         strcpy(tHwmcaSnmpTarget.szCommunity,COMMUNITY);
         usRc = HwmcaInitialize(&tHwmcaInitialize,(ULONG)HWMCAAPI_TIMEOUT);
         if (!usRc) {  /* Initialize with HWMCA API server successful */
            printf("HwmcaInitialize call was successful\n");
            printf("HwmcaInitialize target host            = %s\n",
                   tHwmcaSnmpTarget.pHost);
            printf("HwmcaInitialize target community name  = %s\n",
                   tHwmcaInitialize.protocol.snmp.szCommunity);
            printf("HwmcaInitialize socket                 = %ld\n",
                   tHwmcaInitialize.protocol.snmp.iAgentSocket);
            printf("HwmcaInitialize agent Internet address = %x\n",
                   tHwmcaInitialize.protocol.snmp.ulInetAddr);
            switch ((atoi(argv[1]))) {
            case 1: /* Get request */
               ulLength     = HWMCA_DATATYPE_SIZE;
               pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
               memset(&tHwmcaDataType,'\0',HWMCA_DATATYPE_SIZE);
               usRc = HwmcaGet(&tHwmcaInitialize,argv[3],&tHwmcaDataType,
                               ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
               if (!usRc) { /* Data returned from HwmcaGet */
                  /* Need a larger buffer for the Get request */
                  if (ulBytesNeeded > ulLength) {
                     pHwmcaDataType = (HWMCA_DATATYPE_P)(malloc(ulBytesNeeded));
                     if (pHwmcaDataType) {
                        memset(pHwmcaDataType,'\0',ulBytesNeeded);
```

```
                  ulLength = ulBytesNeeded;
                  usRc = HwmcaGet(&tHwmcaInitialize,argv[3],pHwmcaDataType,
                                  ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
                  if (!usRc) { /* Get request successful */
                     /* Check if it is a Group contents Get */
                     if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
                        parse_and_print_get_group_contents(pHwmcaDataType);
                     } else {
                        parse_and_print_get(pHwmcaDataType);
                     } /* endif */
                     free(pHwmcaDataType);
                  } else {
                     printf("Error in HwmcaGet call return code = %ld\n",usRc);
                  } /* endif */
               } else {
                  printf("Error in allocating %ld bytes for an HwmcaGet call",
                         ulBytesNeeded);
               } /* endif */
            } else {
               /* Check if it is a Group contents Get */
               if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
                  parse_and_print_get_group_contents(&tHwmcaDataType);
               } else {
                  parse_and_print_get(&tHwmcaDataType);
               } /* endif */
            } /* endif */
         } else {
            printf("Error in HwmcaGet call return code = %ld\n",usRc);
         } /* endif */
         break;
      case 2: /* Get-Next request */
         ulLength     = HWMCA_DATATYPE_SIZE;
         pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
         memset(&tHwmcaDataType,'\0',HWMCA_DATATYPE_SIZE);
         usRc = HwmcaGetNext(&tHwmcaInitialize,argv[3],&tHwmcaDataType,
                             ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
         if (!usRc) { /* Data returned from HwmcaGetNext */
            /* Need a larger buffer for the Get request */
            if (ulBytesNeeded > ulLength) {
               pHwmcaDataType = (HWMCA_DATATYPE_P)(malloc(ulBytesNeeded));
               if (pHwmcaDataType) {
                  memset(pHwmcaDataType,'\0',ulBytesNeeded);
                  ulLength = ulBytesNeeded;
                  usRc = HwmcaGetNext(&tHwmcaInitialize,argv[3],
                                      pHwmcaDataType,ulLength,
                                      &ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
                  if (!usRc) { /* Get request successful */
                     /* Check if it is a Group contents Get-Next */
                     if ((pHwmcaDataType->ucType == HWMCA_TYPE_OBJECTID) &&
                         (strstr(pHwmcaDataType->pData,HWMCA_GROUP_CONTENTS_SUFFIX))) {
                        parse_and_print_get_group_contents(pHwmcaDataType);
                     } else {
                        parse_and_print_get(pHwmcaDataType);
                     } /* endif */
                     free(pHwmcaDataType);
                  } else {
                     printf("Error in HwmcaGetNext call return code = %ld\n",usRc);
                  } /* endif */
               } else {
                  printf("Error in allocating %ld bytes for an HwmcaGet call",
                         ulBytesNeeded);
               } /* endif */
            } else {
               /* Check if it is a Group contents Get-Next */
               if ((pHwmcaDataType->ucType == HWMCA_TYPE_OBJECTID) &&
                   (strstr(pHwmcaDataType->pData,HWMCA_GROUP_CONTENTS_SUFFIX))) {
                  parse_and_print_get_group_contents(&tHwmcaDataType);
               } else {
                  parse_and_print_get(&tHwmcaDataType);
               } /* endif */
            } /* endif */
         } else {
            printf("Error in HwmcaGetNext call return code = %ld\n",usRc);
         } /* endif */
         break;
      case 3: /* Set request */
         ulLength     = HWMCA_DATATYPE_SIZE;
         pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
         memset(&tHwmcaDataType,'\0',HWMCA_DATATYPE_SIZE);
         tHwmcaDataType.ucType = (UCHAR)atoi(argv[4]);
         if (tHwmcaDataType.ucType == HWMCA_TYPE_OCTETSTRING) {
            tHwmcaDataType.ulLength = strlen(argv[5])+1;
```

```
                tHwmcaDataType.pData    = argv[5];
             } else {
                tHwmcaDataType.ulLength = sizeof(ULONG);
                ulBytesNeeded         = atol(argv[5]);
                tHwmcaDataType.pData    = &ulBytesNeeded;
             } /* endif */
             usRc = HwmcaSet(&tHwmcaInitialize,argv[3],&tHwmcaDataType,
(ULONG)HWMCAAPI_TIMEOUT);
             if (usRc) {
                printf("Error in HwmcaSet call return code = %ld\n",usRc);
             } /* endif */
             break;
          case 4: /* Command request */
             for (i=5, j=0; (((i+2) <= argc) && (j < 10)); i+=2, j++) {
                memset(&ahWmcaDataType[j],'\0',HWMCA_DATATYPE_SIZE);
                aHwmcaDataType[j].pNext = &(aHwmcaDataType[j+1]);
                aHwmcaDataType[j].ucType = (UCHAR)atoi(argv[i]);
                switch (aHwmcaDataType[j].ucType) {
                   case HWMCA_TYPE_OCTETSTRING:
                      aHwmcaDataType[j].ulLength = strlen(argv[i+1])+1;
                      aHwmcaDataType[j].pData    = argv[i+1];
                      break;
                   case HWMCA_TYPE_NULL:
                      aHwmcaDataType[j].ulLength = 0;
                      aHwmcaDataType[j].pData    = (PVOID)NULL;
                      break;
                   default:
                      aHwmcaDataType[j].ulLength = sizeof(ULONG);
                      aulCmdData[j]            = atol(argv[i+1]);
                      aHwmcaDataType[j].pData    = &aulCmdData[j];
                      break;
                } /* endswitch */
             } /* endfor */
             if (j == 0) {
                pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
             } else {
                aHwmcaDataType[j-1].pNext = (HWMCA_DATATYPE_P)NULL;
                pHwmcaDataType = aHwmcaDataType;
             } /* endif */
             usRc = HwmcaCommand(&tHwmcaInitialize,argv[3],argv[4],
                                 pHwmcaDataType,(ULONG)HWMCAAPI_TIMEOUT);
             if (!usRc) {
                printf("HwmcaCommand request was successful; waiting for the command response
event.\n");
                while (!usRc) {
                   usRc = HwmcaWaitEvent(&tHwmcaInitialize,cEventBuf,sizeof(cEventBuf),
                                         &ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
                   if (!usRc) { /* WaitEvent request successful */
                      if (ulBytesNeeded <= sizeof(cEventBuf)) {
                         parse_and_print_get((HWMCA_DATATYPE_P)cEventBuf);
                      } else {
                         printf("Event buffer not large enough!\n");
                      } /* endif */
                   } else {
                      printf("Error in HwmcaWaitEvent call return code = %ld\n",usRc);
                   } /* endif */
                } /* endwhile */
             } else {
                printf("Error in HwmcaCommand call return code = %ld\n",usRc);
             } /* endif */
             break;
          case 5: /* WaitEvent request */
             usRc = 0;
             while ((!usRc) && (argc >= 5)) {
                usRc = HwmcaWaitEvent(&tHwmcaInitialize,cEventBuf,sizeof(cEventBuf),
                                      &ulBytesNeeded,(ULONG)atol(argv[4]));
                if (!usRc) { /* WaitEvent request successful */
                   if (ulBytesNeeded <= sizeof(cEventBuf)) {
                      parse_and_print_get((HWMCA_DATATYPE_P)cEventBuf);
                   } else {
                      printf("Event buffer not large enough!\n");
                   } /* endif */
                } else {
                   printf("Error in HwmcaWaitEvent call return code = %ld\n",usRc);
                } /* endif */
             } /* endwhile */
             break;
          case 6: /* Build Id request */
             pszAttribute = pszGroupName = pszObjectName = (PSZ)NULL;
             switch (argc) {
                case 7:
                   if (strlen(argv[6])) {
```

```
                  pszObjectName = argv[6];
               } /* endif */
            case 6:
               if (strlen(argv[5])) {
                  pszGroupName = argv[5];
               } /* endif */
            case 5:
               if (strlen(argv[4])) {
                  pszAttribute = argv[4];
               } /* endif */
               break;
            default:
               break;
         } /* endswitch */
         usRc = HwmcaBuildId(szOID,argv[3],pszAttribute,pszGroupName,pszObjectName);
         if (!usRc) {
            printf("HwmcaBuildId build object identifier %s.\n",szOID);
            ulLength     = HWMCA_DATATYPE_SIZE;
            pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
            memset(&tHwmcaDataType,'\0',HWMCA_DATATYPE_SIZE);
            usRc = HwmcaGet(&tHwmcaInitialize,szOID,&tHwmcaDataType,
                        ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
            if (!usRc) { /* Data returned from HwmcaGet */
               /* Need a larger buffer for the Get request */
               if (ulBytesNeeded > ulLength) {
                  pHwmcaDataType = (HWMCA_DATATYPE_P)(malloc(ulBytesNeeded));
                  if (pHwmcaDataType) {
                     memset(pHwmcaDataType,'\0',ulBytesNeeded);
                     ulLength = ulBytesNeeded;
                     usRc = HwmcaGet(&tHwmcaInitialize,szOID,pHwmcaDataType,
                                     ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
                     if (!usRc) { /* Get request successful */
                        /* Check if it is a Group contents Get */
                        if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
                           parse_and_print_get_group_contents(pHwmcaDataType);
                        } else {
                           parse_and_print_get(pHwmcaDataType);
                        } /* endif */
                        free(pHwmcaDataType);
                     } else {
                        printf("Error in HwmcaGet call return code = %ld\n",usRc);
                     } /* endif */
                  } else {
                     printf("Error in allocating %ld bytes for an HwmcaGet call",
                             ulBytesNeeded);
                  } /* endif */
               } else {
                  /* Check if it is a Group contents Get */
                  if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
                     parse_and_print_get_group_contents(&tHwmcaDataType);
                  } else {
                     parse_and_print_get(&tHwmcaDataType);
                  } /* endif */
               } /* endif */
            } else {
               printf("Error in HwmcaGet call return code = %ld\n",usRc);
            } /* endif */
         } else {
            printf("Error in HwmcaBuildId call return code = %ld\n",usRc);
         } /* endif */
         break;
      case 7: /* Build Attribute Id request */
         usRc = HwmcaBuildAttributeId(szOID,argv[3],argv[4]);
         if (!usRc) {
            printf("HwmcaBuildAttributeId build object identifier %s.\n",szOID);
            ulLength     = HWMCA_DATATYPE_SIZE;
            pHwmcaDataType = (HWMCA_DATATYPE_P)NULL;
            memset(&tHwmcaDataType,'\0',HWMCA_DATATYPE_SIZE);
            usRc = HwmcaGet(&tHwmcaInitialize,szOID,&tHwmcaDataType,
                        ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
            if (!usRc) { /* Data returned from HwmcaGet */
               /* Need a larger buffer for the Get request */
               if (ulBytesNeeded > ulLength) {
                  pHwmcaDataType = (HWMCA_DATATYPE_P)(malloc(ulBytesNeeded));
                  if (pHwmcaDataType) {
                     memset(pHwmcaDataType,'\0',ulBytesNeeded);
                     ulLength = ulBytesNeeded;
                     usRc = HwmcaGet(&tHwmcaInitialize,szOID,pHwmcaDataType,
                                     ulLength,&ulBytesNeeded,(ULONG)HWMCAAPI_TIMEOUT);
                     if (!usRc) { /* Get request successful */
                        /* Check if it is a Group contents Get */
                        if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
```

```
                              parse_and_print_get_group_contents(pHwmcaDataType);
                           } else {
                              parse_and_print_get(pHwmcaDataType);
                           } /* endif */
                           free(pHwmcaDataType);
                        } else {
                           printf("Error in HwmcaGet call return code = %ld\n",usRc);
                        } /* endif */
                     } else {
                        printf("Error in allocating %ld bytes for an HwmcaGet call",
                               ulBytesNeeded);
                     } /* endif */
                  } else {
                     /* Check if it is a Group contents Get */
                     if (strstr(argv[3],HWMCA_GROUP_CONTENTS_SUFFIX)) {
                        parse_and_print_get_group_contents(&tHwmcaDataType);
                     } else {
                        parse_and_print_get(&tHwmcaDataType);
                     } /* endif */
                  } /* endif */
               } else {
                  printf("Error in HwmcaGet call return code = %ld\n",usRc);
               } /* endif */
            } else {
               printf("Error in HwmcaBuildId call return code = %ld\n",usRc);
            } /* endif */
            break;
         default:
           break;
         } /* endswitch */
         usRc = HwmcaTerminate(&tHwmcaInitialize,(ULONG)HWMCAAPI_TIMEOUT);
         if (!usRc) {  /* Terminate with HWMCA API server successful */
            printf("HwmcaTerminate socket                = %ld\n",
                   tHwmcaInitialize.protocol.snmp.iAgentSocket);
            printf("HwmcaTerminate agent Internet address = %x\n",
                   tHwmcaInitialize.protocol.snmp.ulInetAddr);
         } else {
            printf("Error in HwmcaTerminate call return code = %ld\n",usRc);
         } /* endif */
      } else {
         printf("Error in HwmcaInitialize call return code = %ld\n",usRc);
      } /* endif */
   } /* endif */
} else {
   usfContinue = FALSE;
} /* endif */
if (!usfContinue) {
   printf("*********************************************************\n");
   printf("*** Program requires the following parameters:       ***\n");
   printf("***                                                  ***\n");
   printf("***    Type of request: (use 1 - 7 as the parameter  ***\n");
   printf("***                      for the type of request)    ***\n");
   printf("***       1 - Get               request              ***\n");
   printf("***       2 - Get-Next          request              ***\n");
   printf("***       3 - Set               request              ***\n");
   printf("***       4 - Command           request              ***\n");
   printf("***       5 - Wait Event        request              ***\n");
   printf("***       6 - Build Id          request              ***\n");
   printf("***       7 - Build Attribute Id request             ***\n");
   printf("***                                                  ***\n");
   printf("***    Internet address of the Console Application   ***\n");
   printf("***        9.130.1.133                               ***\n");
   printf("***                                                  ***\n");
   printf("***    Request specific parameters:                  ***\n");
   printf("***       For a Get or Get-Next request:             ***\n");
   printf("***          Object ID (1.3.6.1.etc)                 ***\n");
   printf("***                                                  ***\n");
   printf("***       For a Set request:                         ***\n");
   printf("***          Object ID (1.3.6.1.etc)                 ***\n");
   printf("***          Set data type (2-integer,4-string,etc)  ***\n");
   printf("***          Set data                                ***\n");
   printf("***                                                  ***\n");
   printf("***       For a Command request:                     ***\n");
   printf("***          Target Object ID (1.3.6.1.etc)          ***\n");
   printf("***          Command Object ID (1.3.6.1.etc)         ***\n");
   printf("***                                                  ***\n");
   printf("***       For a Wait Event request:                  ***\n");
   printf("***          Event mask                              ***\n");
   printf("***          Timeout value in milliseconds           ***\n");
   printf("***             (-1 --> forever)                     ***\n");
   printf("***                                                  ***\n");
   printf("***       For a Build Id request:                    ***\n");
```

```
      printf("***          Object ID Prefix (1.3.6.1.etc)          ***\n");
      printf("***          Attribute suffix (optional)             ***\n");
      printf("***          Group name       (optional)             ***\n");
      printf("***          Object name      (optional)             ***\n");
      printf("***                                                  ***\n");
      printf("***       For a Build Attribute Id request:          ***\n");
      printf("***          Object ID (1.3.6.1.etc)                 ***\n");
      printf("***          Attribute suffix                        ***\n");
      printf("******************************************************\n");
   } /* endif */

}  /* end main */
VOID parse_and_print_get(HWMCA_DATATYPE_P pHwmcaDataType)
{
   HWMCA_DATATYPE_P   pLoopHwmcaDataType;

   pLoopHwmcaDataType = pHwmcaDataType;
   while (pLoopHwmcaDataType) {
      switch (pLoopHwmcaDataType->ucType) {
      case HWMCA_TYPE_SEQUENCE:
         break;
      case HWMCA_TYPE_INTEGER:
         printf("HWMCA_TYPE_INTEGER returned size = %d and pData = %d\n",
                pLoopHwmcaDataType->ulLength,*((PINT)(pLoopHwmcaDataType->pData)));
         break;
      case HWMCA_TYPE_OCTETSTRING:
         printf("HWMCA_TYPE_OCTETSTRING returned size = %d and pData = %s\n",
                pLoopHwmcaDataType->ulLength,pLoopHwmcaDataType->pData);
         break;
      case HWMCA_TYPE_NULL:
         printf("HWMCA_TYPE_NULL returned size = %d\n",pLoopHwmcaDataType->ulLength);
         break;
      case HWMCA_TYPE_OBJECTID:
         printf("HWMCA_TYPE_OBJECTID returned size = %d and pData = %s\n",
                pLoopHwmcaDataType->ulLength,pLoopHwmcaDataType->pData);
         break;
      case HWMCA_TYPE_IPADDRESS:
         printf("HWMCA_TYPE_IPADDRESS returned size = %d and pData = %x\n",
                pLoopHwmcaDataType->ulLength,*((PINT)(pLoopHwmcaDataType->pData)));
         break;
      default:
         printf("UNKNOWN Data type returned = %d\n",pLoopHwmcaDataType->ucType);
        break;
      } /* endswitch */
      pLoopHwmcaDataType = pLoopHwmcaDataType->pNext;
   } /* endwhile */

} /* end of parse_and_print_get */
VOID parse_and_print_get_group_contents(HWMCA_DATATYPE_P pHwmcaDataType)
{
   PSZ              pszGroupContents;
   PUCHAR           pBlank;
   HWMCA_DATATYPE_P   pLoopHwmcaDataType;

   pLoopHwmcaDataType = pHwmcaDataType;
   while (pLoopHwmcaDataType) {
      switch (pLoopHwmcaDataType->ucType) {
      case HWMCA_TYPE_OBJECTID:
         printf("HWMCA_TYPE_OBJECTID returned size = %d and pData = %s\n",
                pLoopHwmcaDataType->ulLength,pLoopHwmcaDataType->pData);
         break;
      case HWMCA_TYPE_OCTETSTRING:
         printf("HWMCA_TYPE_OCTETSTRING returned size = %d and pData = %s\n",
                pLoopHwmcaDataType->ulLength,pLoopHwmcaDataType->pData);
         pszGroupContents = (PSZ)pLoopHwmcaDataType->pData;
         pBlank = pszGroupContents;
         pBlank = strchr(pBlank,' ');
         while (pBlank) {
            *pBlank = '\0';
            printf("Group contents Object ID = %s\n",pszGroupContents);
            pBlank++;
            pszGroupContents = pBlank;
            pBlank = strchr(pBlank,' ');
         } /* endwhile */
         printf("Group contents Object ID = %s\n",pszGroupContents);
         break;
      case HWMCA_TYPE_NULL:
         printf("HWMCA_TYPE_NULL returned size = %d\n",pLoopHwmcaDataType->ulLength);
         break;
      default:
         printf("UNKNOWN Data type returned = %d\n",pLoopHwmcaDataType->ucType);
        break;
```

```
      } /* endswitch */
      pLoopHwmcaDataType = pLoopHwmcaDataType->pNext;
   } /* endwhile */

} /* end of parse_and_print_get_group_contents */
```

Chapter 4. Console application managed objects

This chapter contains definitions of the objects the Console application manages. Each object contains the following:

- **Object Type**
- **Object Name Bindings:** Shows the name of the base object that is used in the Management Commands API.
- **Object Attributes:** Describes each attribute an object contains and the operations supported against that attribute. The operations supported are:

  Get: Retrieve the current attribute value of an object
  Set: The attribute value of an object

  It also shows the attribute name of an object (SNMP MIB name) that is used in the management APIs.

  Important information about object attributes: Unless otherwise specified in Appendix E, "Object Attribute Availability," on page 203, it can be assumed that each object attribute described in this chapter is valid for any level of object. For any object attribute that is not valid for all levels, Table 2 on page 203 defines the level of objects required for the attribute.
- **Object Relationship:** Describes any pertinent relationships the object contains with other objects.
- **Commands that can be performed on that object:** Describes each command that is valid for the object and also shows the name of the command that is used in the Management Commands API when requesting a command to be performed on the object. For the SNMP version, the command name is called the SNMP MIB Name.
- **Emitted object asynchronous notifications:** Describes the significant notifications an object will emit to a registered application.

Console application object identifier conventions

All the objects managed by the Console application follow the same object identifier naming scheme. The naming scheme used by the Console breaks the object identifiers into four distinct portions:

prefix.attribute.group.object

The meanings and options for each of these portions are described in the following pages:

prefix

This portion of the object identifier must be one of the following:

1.3.6.1.4.1.2.6.42.0

An attribute of the Console object or the Console object itself.

1.3.6.1.4.1.2.6.42.1

An attribute of the Defined CPCs group object or the Defined CPCs group object itself.

1.3.6.1.4.1.2.6.42.1.0

An attribute of a Defined CPC object or a Defined CPC object itself.

1.3.6.1.4.1.2.6.42.2

An attribute of the CPC Images group object or the CPC Images group object itself.

1.3.6.1.4.1.2.6.42.2.0

An attribute of a CPC Image object or a CPC Image object itself.

1.3.6.1.4.1.2.6.42.3

An attribute of a user-defined group object or a user-defined group object itself.

1.3.6.1.4.1.2.6.42.3.0

An attribute of an object contained within a user-defined group object or an object contained within a user-defined group object itself.

1.3.6.1.4.1.2.6.42.4

A Console application command object.

1.3.6.1.4.1.2.6.42.5

An attribute of a Reset Activation Profile or a Reset Activation Profile object itself.

1.3.6.1.4.1.2.6.42.6

An attribute of an Image Activation Profile or an Image Activation Profile object itself.

1.3.6.1.4.1.2.6.42.7

An attribute of a Load Activation Profile or a Load Activation Profile object itself.

1.3.6.1.4.1.2.6.42.8

An attribute of a Group Activation Profile or a Group Activation Profile object itself.

1.3.6.1.4.1.2.6.42.9.0

An attribute of a Capacity Record object or a Capacity Record object itself.

1.3.6.1.4.1.2.6.42.13.0

An attribute of an LPAR capacity group object or an LPAR capacity group object itself.

attribute

This portion of the object identifier is used when specifying an object identifier for an attribute of an object. It is optional and when not specified results in an object identifier for the object itself.

group

This portion of the object identifier is used to uniquely specify which user-defined group this object identifier pertains to. It is optional and should only be used for the following object identifiers:

- User-defined groups
- User-defined group attributes
- Objects contained within user-defined groups
- Attributes of objects contained within user-defined groups
- Reset Activation Profile, Image Activation Profile, and Load Activation Profile objects (in this case the group value is used to identify the CPC object that the activation profile pertains to)
- Attributes of Reset Activation Profile, Image Activation Profile, and Load Activation Profile objects (in this case the group value is used to identify the CPC object that the activation profile attribute pertains to).

This value is generated using the name attribute of the group object.

object

This portion of the object identifier is used to uniquely specify which object within a group this object identifier pertains to. It is optional and should only be used for the following object identifiers:

- Objects contained within a group
- Attributes of objects contained within a group
- Reset Activation Profile, Image Activation Profile, and Load Activation Profile objects
- Attributes of Reset Activation Profile, Image Activation Profile, and Load Activation Profile objects.

This value is generated using the name attribute of the object.

Console application object

Console application name bindings

Console object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.x.x

Where ***x.x*** equals the attribute identifier for the object.

Console attributes

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.1.0

SNA address

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING (The OCTET string returned contains the SNA address in the form NetId.Name)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.16.0

Group contents

Get: Null terminated collection of blank separated object identifier strings.

- Data type(s) returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.*
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.23.0

Version

Get: The version number for the console.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.151.0

Internet Protocol (IP) addresses

Get: A null terminated list of blank separated IP addresses being used by the console.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.161.0

Engineering Change (EC)/Microcode Level (MCL)

Get: An XML string that describes the EC and MCL levels for the console.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING

  Note: Refer to Appendix F, "XML descriptions," on page 209 for a detailed description of this XML data.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.0.162.0

Console application commands

Hardware message refresh

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.11 (HWMCA_HW_MESSAGE_REFRESH_COMMAND)

Hardware message delete

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.13 (HWMCA_HW_MESSAGE_DELETE_COMMAND)

Shutdown/Restart

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.22 (HWMCA_SHUTDOWN_RESTART_COMMAND)

Console application notifications

Security log event (HWMCA_EVENT_SECURITY_EVENT)

- An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the security log.
- An HWMCA_TYPE_OCTETSTRING that specifies the text of the security log.

Console application started (HWMCA_EVENT_STARTED)

This event has no additional data.

Console application ended (HWMCA_EVENT_ENDED)

Used to notify the application that the Console application is ending.

The additional data for this event consists of the following object identifier/value pairs:

1. An HWMCA_TYPE_INTEGER that specifies the reason for the event. The possible values are:
   - HWMCA_ENDED_USER - the event was initiated by a user,
   - HWMCA_ENDED_AUTOMATION - the event was initiated by automation, or
   - HWMCA_ENDED_OTHER - the event was initiated by the Console application itself (for example, recovery action, change management, etc.)
2. An HWMCA_TYPE_OCTETSTRING that specifies the name of the Console application component that caused the event.
3. An HWMCA_TYPE_INTEGER that specifies the shutdown type for the event. The possible values are:
   - HWMCA_SHUTDOWN_CONSOLE - the console has been shut down and will take manual intervention to be restarted,
   - HWMCA_RESTART_APPLICATION - the console application has been stopped and will automatically be restarted, or
   - HWMCA_RESTART_CONSOLE - the console has been stopped and will automatically be restarted.

4. An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to.

Message (HWMCA_EVENT_MESSAGE)

- An HWMCA_TYPE_INTEGER that specifies that the message is a Console or Optical Network message (HWMCA_HARDWARE_MESSAGE).
- An HWMCA_TYPE_OCTETSTRING that specifies a description of the new or refreshed Console or Optical Network message.
- An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).
- An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the new or refresh message.
- An HWMCA_TYPE_OCTETSTRING that specifies the name(s) of the CPC Image object(s) associated with the object that generated the new or refresh message.

Message deletion (HWMCA_EVENT_HARDWARE_MESSAGE_DELETE)

- An HWMCA_TYPE_INTEGER that specifies that the message being deleted is a Console or Optical Network message (HWMCA_HARDWARE_MESSAGE).
- An HWMCA_TYPE_OCTETSTRING that specifies the message text of the Console or Optical Network message being deleted.
- An HWMCA_TYPE_INTEGER which is always HWMCA_FALSE for this event.
- An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the message being deleted.
- An HWMCA_TYPE_OCTETSTRING that specifies the name(s) of the CPC Image object(s) associated with the object for which the message is being deleted.

Group

Group name bindings

Defined CPCs group object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1

CPC images group object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2

CPC user group object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3

CPC images user group object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3

Group attributes

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.1.0
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.1.0
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.1.0.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.1.0.*

Status error

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Contains one or more objects that are in a state which is not an acceptable status.

  HWMCA_FALSE
  All objects contained within the group are in an acceptable status state.
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.7.0
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.7.0
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.7.0.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.7.0.*

Busy

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object in a busy state (currently performing a task).

  HWMCA_FALSE
  Object not in a busy state.
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.8.0
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.8.0
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.8.0.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.8.0.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  One of the following values:

  – HWMCA_CPC_GROUP
  – HWMCA_CPC_IMAGE_GROUP
  – HWMCA_CPC_USER_GROUP
  – HWMCA_CPC_IMAGE_USER_GROUP
  – HWMCA_VM_GROUP
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.22.0
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.22.0
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.22.0.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.22.0.*

Contents

Get: Null terminated collection of blank separated object identifier strings.

- Data type(s) returned on Get: HWMCA_TYPE_OCTETSTRING
- Defined CPCs Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.23.0
- CPC Images Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.23.0
- CPC User Group Object SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.23.0.*
- CPC Images User Group SNMP MIB Name: 1.3.6.1.4.1.2.6.42.3.23.0.*

Note: In some cases the size of the data associated with this attribute is larger than what many applications can traditionally handle. In this situation the same information can be determined by issuing a series of GetNext requests to build the collection of object identifier strings.

Group commands

H/W (CPC) group

Commands that can be performed on this group are the same as the commands listed in the Defined CPC object's definition in "Defined CPC commands" on page 96 except for the HWMCA_HW_MESSAGE_REFRESH_COMMAND and HWMCA_HW_MESSAGE_DELETE_COMMAND commands.

CPC image group

Commands that can be performed on this group are the same as the commands listed in CPC image object's definition in "CPC image commands" on page 125. However, the send operating system 1.3.6.1.4.1.2.6.42.4.3 (HWMCA_SEND_OPSYS_COMMAND) listed in this chapter is not valid for sending to a group.

CF image group

Commands that can be performed on this group are the same as the commands listed in CF image object's definition in "Coupling facility commands" on page 138. However, the send operating system 1.3.6.1.4.1.2.6.42.4.3 (HWMCA_SEND_OPSYS_COMMAND) listed in this chapter is not valid for sending to a group.

Group notifications

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the *HwmcaGet* to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Defined CPC

Defined CPC name bindings

CPC object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.x.x.*

Where ***x.x.*** equals the attribute identifier for the object and ***** equals a unique number for that specific instance of the CPC Object.

Defined CPC attributes

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.1.0.*

Status error

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object is in a state which is not an acceptable status.

  HWMCA_FALSE
  Object is in an acceptable status state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.7.0.*

Busy

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object is in a busy state (currently performing a task).

  HWMCA_FALSE
  Object is not in a busy state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.8.0.*

Message indicator

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object has a hardware message.

  HWMCA_FALSE
  Object does not have a hardware message.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.9.0.*

Status

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  One of the following bit values will be set to on:

  – HWMCA_STATUS_OPERATING
  – HWMCA_STATUS_NOT_OPERATING
  – HWMCA_STATUS_NO_POWER
  – HWMCA_STATUS_EXCEPTIONS
  – HWMCA_STATUS_STATUS_CHECK
  – HWMCA_STATUS_SERVICE
  – HWMCA_STATUS_LINKNOTACTIVE
  – HWMCA_STATUS_POWERSAVE
  – HWMCA_STATUS_SERVICE_REQ
  – HWMCA_STATUS_DEGRADED
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.10.0.*

Acceptable status

Get/Set:

- Data type returned on Get: HWMCA_TYPE_INTEGER

- Data type for Set: HWMCA_TYPE_INTEGER

  One or more of the following bit values will be set to on:

  - HWMCA_STATUS_OPERATING
  - HWMCA_STATUS_NOT_OPERATING
  - HWMCA_STATUS_NO_POWER
  - HWMCA_STATUS_EXCEPTIONS
  - HWMCA_STATUS_STATUS_CHECK
  - HWMCA_STATUS_SERVICE
  - HWMCA_STATUS_LINKNOTACTIVE
  - HWMCA_STATUS_POWERSAVE
  - HWMCA_STATUS_SERVICE_REQ
  - HWMCA_STATUS_DEGRADED
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.11.0.*

IML mode

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER
  - HWMCA_IML_ESA390_MODE
  - HWMCA_IML_LPAR_MODE
  - HWMCA_IML_ESA390TPF_MODE
  - HWMCA_IML_LINUX_MODE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.12.0.*

Activation profile name

Get/Set (Reset or Load profile):

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A maximum length of 17 bytes is allowed for the activation profile name, including the null terminator.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.13.0.*

Last used activation profile

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.14.0.*

Internet address

Get:

- Data type returned on Get: HWMCA_TYPE_IPADDRESS
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.15.0.*

SNA address

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING (The OCTET string returned will contain the SNA address in the form NetId.Name.)

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.16.0.*

Computer (machine) model

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.17.0.*

Computer (machine) type

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.18.0.*

Computer (machine) serial

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.19.0.*

CPC serial number

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.20.0.*

CPC identifier

Get: Node descriptor identifier calculated by using location within computer (machine).

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.21.0.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_CPC_OBJECT
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.22.0.*

List of reset activation profiles

Get: This returns a null terminated collection of blank separated object identifiers for each Reset Activation profile.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.24.0.*

List of image activation profiles

Get: This returns a null terminated collection of blank separated object identifiers for each Image Activation profile.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.25.0.*

List of load activation profiles

Get: This returns a null terminated collection of blank separated object identifiers for each Load Activation profile.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.26.0.*

CBU installed

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  CBU is installed.

  HWMCA_FALSE
  CBU is not installed.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.32.0.*

CBU activated

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  CBU is activated.

  HWMCA_FALSE
  CBU is not activated.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.33.0.*

CBU activation date

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.34.0.*

CBU expiration date

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.35.0.*

Number of CBU tests left

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.36.0.*

Real CBU activation available

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Real CBU is available.

  HWMCA_FALSE
  Real CBU is not available.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.37.0.*

Reserve ID

Note: This attribute is available only on a Support Element console.

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING (The OCTET string returned contains the name of the application that currently holds the reserve. A zero length string implies that no application holds the reserve.)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.44.0.*

Service required indicator

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Service Required indicator is on.

  HWMCA_FALSE
  Service Required indicator is not on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.46.0.*

Degraded indicator

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER
  - HWMCA_NOT_DEGRADED
  - HWMCA_DEGRADED_MEM
  - HWMCA_DEGRADED_MBA
  - HWMCA_DEGRADED_NODE
  - HWMCA_DEGRADED_RING
  - HWMCA_DEGRADED_CBU
  - HWMCA_DEGRADED_MRU
  - HWMCA_DEGRADED_AMBIENT
  - HWMCA_DEGRADED_MRU_IML
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.47.0.*

CBU enabled

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  CBU is enabled.

  HWMCA_FALSE
  CBU is not enabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.48.0.*

List of cluster members

Get: This returns a null terminated collection of blank separated SNA addresses for all other Support Elements considered to be within the same cluster.

Note: This attribute is available only when targeting a Support Element.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.50.0.*

Processor running time type

Get/Set: Defines whether the processor running time is dynamically determined by the system or set to a constant value for the Defined CPC object.

Note: Performing this operation on a Defined CPC object which is not currently in an IML state returns a no such name error.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0 (HWMCA_DETERMINED_SYSTEM)
  The processor running is dynamically determined by the system.

  1 (HWMCA_DETERMINED_USER)
  The processor running time is set to a constant value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.78.0.*

Processor running time

Get/Set: Defines the amount of continuous time allowed for logical processors to perform jobs on shared processors for the Defined CPC object.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  A value 1 - 100 for the user-defined processor running time.

  Note: This value can only be set if the processor running time type is set to 1 (HWMCA_DETERMINED_USER).
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.79.0.*

End timeslice if CPC image enters a wait state

Get/Set: Defines whether CPC Images lose their share of running time when they enter a wait state. (Only supported for Defined CPC objects earlier than version 2.14.0).

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that a CPC Image should lose its share of running time when it enters a wait state.

  HWMCA_FALSE
  Indicates that a CPC Image should not lose its share of running time when it enters a wait state.

  Note: This value can only be set if the processor running time type is set to 1 (HWMCA_DETERMINED_USER).
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.80.0.*

On/Off Capacity on Demand (On/Off CoD) installed

Get: Defines whether On/Off Capacity on Demand is installed for the Defined CPC object.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  On/Off CoD is installed.

  HWMCA_FALSE
  On/Off CoD is not installed.

  Note: The attribute On/Off Capacity on Demand (On/Off CoD) Installed and attribute On/Off Capacity on Demand (On/Off CoD) Activated always have the same value, either HWMCA_TRUE or HWMCA_FALSE.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.87.0.*

On/Off Capacity on Demand (On/Off CoD) activated

Get: Defines whether On/Off Capacity on Demand is currently activated for the Defined CPC object.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  On/Off CoD is activated.

  HWMCA_FALSE
  On/Off CoD is not activated

  Note: The attribute On/Off Capacity on Demand (On/Off CoD) Installed and attribute On/Off Capacity on Demand (On/Off CoD) Activated always have the same value, either HWMCA_TRUE or HWMCA_FALSE.

  .
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.88.0.*

On/Off Capacity on Demand (On/Off CoD) enabled

Get: Defines whether On/Off Capacity on Demand is enabled for the Defined CPC object.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  On/Off CoD is enabled.

  HWMCA_FALSE
  On/Off CoD is not enabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.89.0.*

On/Off Capacity on Demand (On/Off CoD) activation date

Get: Defines the time stamp for when On/Off CoD was activated for the Defined CPC object.

- Data type for Get: HWMCA_TYPE_OCTETSTRING

  A time stamp string describing when On/Off CoD was activated or an empty string if On/Off CoD is not activated.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.90.0.*

List of group profiles

Get: This returns a null terminated collection of blank separated object identifiers for each Group profile.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.91.0.*

Temporary capacity records

Get: A blank separated list of SNMP object identifiers for the installed temporary capacity records.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.119.0.*

Permanent software model

Get: The software model based on the permanent processors.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.120.0.*

Permanent plus billable software model

Get: The software model based on the permanent plus billable processors.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.121.0.*

Permanent plus all temporary software model

Get: The software model based on the permanent plus all temporary processors.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.122.0.*

Permanent MSU

Get: The MSU value associated with the software model based on the permanent processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.123.0.*

Permanent plus billable MSU

Get: The MSU value associated with the software model based on the permanent plus billable processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.124.0.*

Permanent plus all temporary MSU

Get: The MSU value associated with the software model based on the permanent plus all temporary processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.125.0.*

General purpose processors

Get: The count of general purpose processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.126.0.*

Service assist processors

Get: The count of service assist processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.127.0.*

Application Assist Processor (AAP) processors

Get: The count of Application Assist Processor (AAP) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.128.0.*

Integrated Facility for Linux (IFL) processors

Get: The count of Integrated Facility for Linux (IFL) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.129.0.*

Internal Coupling Facility (ICF) processors

Get: The count of Internal Coupling Facility (ICF) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.130.0.*

z Integrated Information Processors (zIIP) processors

Get: The count of Integrated Information Processors (zIIP) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.131.0.*

Defective processors

Get: The count of defective processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.132.0.*

Spare processors

Get: The count of spare processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.133.0.*

Pending processors

Get: The count of processors pending activation.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.134.0.*

Temporary capacity change allowed

Get: This value is used to determine if API applications are allowed to make changes to temporary capacity.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  API applications are allowed to perform temporary capacity changes.

  HWMCA_FALSE
  API applications are not allowed to perform temporary capacity changes.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.149.0.*

Version

Get: The version number for the Defined CPC.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.151.0.*

Internet Protocol (IP) addresses

Get: A null terminated list of blank separated IP addresses being used by the defined CPC object.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.161.0

Engineering Change (EC)/Microcode Level (MCL)

Get: An XML string that describes the EC and MCL levels for the defined CPC object. For more information, see Appendix F, “XML descriptions,” on page 209.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.162.0

Automatic switch enabled

Get: This value is used to determine if automatic switching between primary and alternate Support Elements is enabled for the Defined CPC object.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Automatic switching is enabled.

  HWMCA_FALSE
  Automatic switching is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.163.0.*

Server Time Protocol (STP) configuration

Get: An XML string that describes the STP configuration for the defined CPC object. For more information, see Appendix F, "XML descriptions," on page 209.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.165.0

Pending General Purpose Processors

Get: The count of pending general purpose processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.175.0.*

Pending Service Assist Processors

Get: The count of pending service assist processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.176.0.*

Pending Application Assist Processor (AAP) Processors

Get: The count of pending Application Assist Processor (AAP) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.177.0.*

Pending Integrated Facility for Linux (IFL) Processors

Get: The count of pending Integrated Facility for Linux (IFL) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.178.0.*

Pending Internal Coupling Facility (ICF) Processors

Get: The count of pending Internal Coupling Facility (ICF) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.179.0.*

Pending z Integrated Information Processors (zIIP) Processors

Get: The count of pending Integrated Information Processors (zIIP) processors.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.180.0.*

Processor Information

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the HwmcaEnhancedGet interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all of the processor related attributes for the Defined CPC. These attributes are:

  - Permanent software model
  - Permanent plus billable software model
  - Permanent plus all temporary software model
  - Permanent MSU
  - Permanent plus billable MSU
  - Permanent plus all temporary MSU
  - General Purpose processors
  - Service Assist processors
  - Application Assist Processor processors
  - Integrated Facility for Linux processors
  - Internal Coupling Facility processors
  - Integrated Information Processing processors
  - Defective processors
  - Spare processors
  - Pending processors
  - Pending General Purpose processors
  - Pending Service Assist processors
  - Pending Application Assist Processor processors
  - Pending Integrated Facility for Linux processor
  - Pending Internal Coupling Facility processors
  - Pending Integrated Information Processing processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.235.0.*

LPAR Controls

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all of the LPAR Control related attributes for each CPC Image object associated with the Defined CPC. These attributes are:

  - Initial processing weight
  - Initial processing weight capped
  - Minimum processing weight
  - Maximum processing weight
  - Current processing weight
  - Current processing weight capped

  - Workload manager enabled
  - Defined capacity
  - Initial Application Assist Processor processing weight
  - Initial Application Assist Processor processing weight capped
  - Minimum Application Assist Processor processing weight
  - Maximum Application Assist Processor processing weight
  - Current Application Assist Processor processing weight
  - Current Application Assist Processor processing weight capped
  - Initial Internal Coupling Facility processing weight
  - Initial Internal Coupling Facility processing weight capped
  - Minimum Internal Coupling Facility processing weight
  - Maximum Internal Coupling Facility processing weight
  - Current Internal Coupling Facility processing weight
  - Current Internal Coupling Facility processing weight capped
  - Initial Integrated Facility for Linux processing weight
  - Initial Integrated Facility for Linux processing weight capped
  - Minimum Integrated Facility for Linux processing weight
  - Maximum Integrated Facility for Linux processing weight
  - Current Integrated Facility for Linux processing weight
  - Current Integrated Facility for Linux processing weight capped
  - Initial z Integrated Information Processors processing weight
  - Initial z Integrated Information Processors processing weight capped
  - Minimum z Integrated Information Processors processing weight
  - Maximum z Integrated Information Processors processing weight
  - Current z Integrated Information Processors processing weight
  - Current z Integrated Information Processors processing weight capped
  - Absolute capping type
  - Absolute capping value
  - Application Assist Processor absolute capping type
  - Application Assist Processor absolute capping value
  - Integrated Facility for Linux absolute capping type
  - Integrated Facility for Linux absolute capping value
  - z Integrated Information Processor absolute capping type
  - z Integrated Information Processor absolute capping value
  - Internal Coupling Facility absolute capping type
  - Internal Coupling Facility absolute capping value
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.236.0.*

Group Capacity

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings..

The sequence contains all of the Group capacity related attributes for each CPC Image object associated with the Defined CPC. These attributes are:

  - Group profile name
  - Group profile capacity
  - Absolute capping type
  - Absolute capping value
  - Application Assist Processor absolute capping type
  - Application Assist Processor absolute capping value
  - Integrated Facility for Linux absolute capping type
  - Integrated Facility for Linux absolute capping value
  - z Integrated Information Processor absolute capping type
  - z Integrated Information Processor absolute capping value

  The group profile name for each CPC Image object will be contained in the returned data, but all of the other attributes are optionally returned based on whether the CPC Image is currently activated and if absolute capping is supported.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.237.0.*

Total installed storage

Get: The amount of installed storage, in megabytes, for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.238.0.*

Hardware system area storage

Get: The amount of storage, in megabytes, reserved for the base hardware system area (HSA) for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.239.0.*

Customer storage

Get: The amount of storage, in megabytes, for use by the customer for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.240.0.*

Customer central storage

Get: The amount of storage, in megabytes, which is the central storage in use across the active CPC Images for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.241.0.*

Customer expanded storage

Get: The amount of storage, in megabytes, which is the expanded storage in use across the active CPC Images for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.242.0.*

Customer available storage

Get: The amount of storage, in megabytes, which is not in use for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.243.0.*

Total installed Virtual Flash Memory

Get: The total amount of Virtual Flash Memory, in gigabytes, installed for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.244.0.*

Virtual Flash Memory increment size

Get: The total increment size, in gigabytes, for Virtual Flash Memory for a Defined CPC object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.245.0.*

Container Based Processors (CBP)

Get:The count of Container Based Processors (CBP).

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.293.0.*

Pending Container Based Processors (CBP)

Get: The count of pending Container Based Processors (CBP).

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.294.0.*

Storage information

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE. The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the HwmcaEnhancedGet interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all the storage related attributes for the Defined CPC and each currently activated CPC Image object associated with it. These attributes are:

  – Defined CPC
    - Total installed storage
    - Hardware system area storage
    - Customer storage
    - Customer central storage
    - Customer expanded storage
    - Customer available storage
    - Total installed Virtual Flash Memory
    - Virtual Flash Memory increment size
  – CPC Image
    - Central storage origin
    - Initial central storage
    - Current central storage
    - Maximum central storage
    - Central storage gap

- Initial expanded storage origin
- Current initial expanded storage
- Initial expanded storage gap
- Reserved expanded storage origin
- Current reserved expanded storage
- Reserved expanded storage gap
- Expanded storage origin
- Initial expanded storage
- Current expanded storage
- Maximum expanded storage
- Expanded storage gap

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.267.0.*

List of LPAR capacity groups

Get: This returns a null terminated collection of blank separated object identifiers for each LPAR capacity group.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.310.0.*

Defined CPC relationships

Cluster (String)

A CPC is a member of a cluster. There are one or more CPCs per cluster.

Support Element

A CPC has a one-to-one relationship with a Support Element (provider of services).

CF/CPC image

A CPC contains one or more images. This is determined by the activation reset profile.

Defined CPC commands

Activate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.1 (HWMCA_ACTIVATE_COMMAND)

Deactivate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.2 (HWMCA_DEACTIVATE_COMMAND)

Hardware message refresh

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.11 (HWMCA_HW_MESSAGE_REFRESH_COMMAND)

Hardware message delete

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.13 (HWMCA_HW_MESSAGE_DELETE_COMMAND)

Activate CBU

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.14 (HWMCA_ACTIVATE_CBU_COMMAND)

Undo CBU

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.15 (HWMCA_UNDO_CBU_COMMAND)

Import profiles

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.16 (HWMCA_IMPORT_PROFILE_COMMAND)

Export profiles

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.17 (HWMCA_EXPORT_PROFILE_COMMAND)

Reserve

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.18 (HWMCA_RESERVE_COMMAND)

Activate On/Off Capacity on Demand (On/Off CoD)

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.23 (HWMCA_ACTIVATE_OOCOD_COMMAND)

Undo On/Off Capacity on Demand (On/Off CoD)

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.24 (HWMCA_UNDO_OOCOD_COMMAND)

Add temporary capacity

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.25 (HWMCA_ADD_CAPACITY_COMMAND)

Remove temporary capacity

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.26 (HWMCA_REMOVE_CAPACITY_COMMAND)

Swap Current Time Server

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.27(HWMCA_SYSPLEX_TIME_SWAP_CTS_COMMAND)

Set STP configuration

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.28(HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND)

Change STP-only CTN

SNMP MIB Name:
1.3.6.1.4.1.2.6.42.4.29(HWMCA_SYSPLEX_TIME_CHANGE_STP_ONLY_CTN_COMMAND)

Join STP-only CTN

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.30(HWMCA_SYSPLEX_TIME_JOIN_STP_ONLY_CTN_COMMAND)

Leave STP-only CTN

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.31 (HWMCA_SYSPLEX_TIME_LEAVE_STP_ONLY_CTN_COMMAND)

Set STP Daylight Savings Time

SNMP MIB Name:1.3.6.1.4.1.2.6.42.4.33 (HWMCA_SYSPLEX_TIME_SET_DST_COMMAND)

Defined CPC notifications

Message (HWMCA_EVENT_MESSAGE)

An HWMCA_TYPE_INTEGER that specifies whether the message is a hardware or operating system message (HWMCA_HARDWARE_MESSAGE or HWMCA_OPSYS_MESSAGE).

For hardware messages:

- An HWMCA_TYPE_OCTETSTRING that specifies a description of the new or refreshed hardware message.
- An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).
- An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the new or refresh message.

- An HWMCA_TYPE_OCTETSTRING that specifies the name(s) of the CPC Image object(s) associated with the object that generated the new or refresh message.

Message deletion (HWMCA_EVENT_HARDWARE_MESSAGE_DELETE)

- An HWMCA_TYPE_INTEGER that specifies that the message being deleted is a CPC-related hardware message (HWMCA_HARDWARE_MESSAGE).
- An HWMCA_TYPE_OCTETSTRING that specifies the message text of the hardware message being deleted.
- An HWMCA_TYPE_INTEGER which is always HWMCA_FALSE for this event.
- An HWMCA_TYPE_OCTETSTRING that specifies the time stamp of the message being deleted.
- An HWMCA_TYPE_OCTETSTRING that specifies the name(s) of the CPC Image object(s) associated with the object for which the message is being deleted.

Status change (HWMCA_EVENT_STATUS_CHANGE)

- An HWMCA_TYPE_INTEGER that specifies the new status value
- An HWMCA_TYPE_INTEGER that specifies the old status value.

Object's name change (HWMCA_EVENT_NAME_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new object name
- An HWMCA_TYPE_OCTETSTRING that specifies the old object name.

Object's activation profile change (HWMCA_EVENT_ACTIVATE_PROF_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new activation profile name
- An HWMCA_TYPE_OCTETSTRING that specifies the old activation profile name.

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the *HwmcaGet* to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Object entered an exception state (HWMCA_EVENT_EXCEPTION_STATE)

- An HWMCA_TYPE_INTEGER that specifies whether the object is entering into an exception state (HWMCA_TRUE) or leaving an exception state (HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies the status value for the object.

Capacity change (HWMCA_EVENT_CAPACITY_CHANGE)

- An HWMCA_TYPE_INTEGER that specifies the type of capacity change that occurred.
- An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a Defined CPC object).

Capacity record change (HWMCA_EVENT_CAPACITY_RECORD_CHANGE)

- An HWMCA_TYPE_INTEGER that specifies the type of capacity record change that occurred.
- An HWMCA_TYPE_OCTETSTRING for the temporary capacity record that has changed.
- An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a Defined CPC object).

CPC image

CPC image name bindings

Image Object Identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.x.x.*

Where ***x.x.*** equals the attribute identifier for the object and ***** equals a unique number for that specific instance of the CPC Image Object.

CPC image attributes

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.1.0.*

Parent's name

Get (CPC's logical name):

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.2.0.*

Operating system name

Get: Name of Operating System running in image, if known.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.3.0.*

Operating system type

Get: Type of Operating System running in image, if known.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.4.0.*

Operating system level

Get: Level of Operating System running in image, if known.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.5.0.*

Sysplex name

Get: Applicable only for MVS™, if known.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.6.0.*

Status error

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object is in a state which is not an acceptable status.

  HWMCA_FALSE
  Object is in an acceptable status state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.7.0.*

Busy

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object in a busy state (currently performing a task)

  HWMCA_FALSE
  Object not in a busy state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.8.0.*

Message indicator

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object has an operating system message.

  HWMCA_FALSE
  Object does not have an operating system message.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.9.0.*

Status

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  One of the following bit values will be set to on:

  – HWMCA_STATUS_OPERATING
  – HWMCA_STATUS_NOT_OPERATING
  – HWMCA_STATUS_NOT_ACTIVATED
  – HWMCA_STATUS_EXCEPTIONS
  – HWMCA_STATUS_STATUS_CHECK
  – HWMCA_STATUS_POWERSAVE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.10.0.*

Acceptable status

Get/Set:

- Data type returned on Get: HWMCA_TYPE_INTEGER
- Data type for Set: HWMCA_TYPE_INTEGER

  One or more of the following bit values will be set to on:

  – HWMCA_STATUS_OPERATING
  – HWMCA_STATUS_NOT_OPERATING
  – HWMCA_STATUS_NOT_ACTIVATED
  – HWMCA_STATUS_EXCEPTIONS
  – HWMCA_STATUS_STATUS_CHECK
  – HWMCA_STATUS_POWERSAVE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.11.0.*

IML/partition activation mode

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  One of the following bit values will be set to on:

  – HWMCA_IML_ESA390_MODE
  – HWMCA_IML_S370_MODE
  – HWMCA_IML_ESA390TPF_MODE
  – HWMCA_IML_CF_PROD_MODE
  – HWMCA_IML_LINUX_MODE
  – HWMCA_IML_ZVM_MODE
  – HWMCA_IML_ZAWARE_MODE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.12.0.*

Activation profile name

Get/Set (Image or Load profile):

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A maximum length of 17 bytes is allowed for the activation profile name, including the null terminator.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.13.0.*

Last used activation profile

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.14.0.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_CPC_IMAGE_OBJECT
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.22.0.*

Initial processing weight

Get/Set: The relative amount of shared general purpose processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the relative amount of shared general purpose processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared general purpose processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated general purpose processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.30.0.*

Initial processing weight capped

Get/Set: Whether or not the initial processing weight for general purpose processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial general purpose processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of general purpose processor resources, regardless of the availability of excess general purpose processor resources.

  HWMCA_FALSE (0)
  Indicates that the initial general purpose processor processing weight for the CPC Image is not capped. It represents the share of general purpose processor resources guaranteed to a logical partition when all general purpose processor resources are in use. Otherwise, when excess general purpose processor resources are available, the logical partition can use them if necessary.

  Note: The initial general purpose processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.

  This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.31.0.*

Minimum processing weight

Get/Set: The minimum relative amount of shared general purpose processor resources allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the minimum relative amount of shared general purpose processor resources allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and less than or equal to the Initial Processing Weight used to define the minimum relative amount of shared general purpose processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated general purpose processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.38.0.*

Maximum processing weight

Get/Set: The maximum relative amount of shared general purpose processor resources allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.

1 - 999
Represents the maximum relative amount of shared general purpose processor resources allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and greater than or equal to the Initial Processing Weight used to define the maximum relative amount of shared general purpose processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated general purpose processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.39.0.*

Current processing weight

Get: The relative amount of shared general purpose processor resources currently allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the relative amount of shared general purpose processor resources currently allocated to the CPC Image object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.41.0.*

Current processing weight capped

Get: Whether or not the current general purpose processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the current general purpose processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the current general purpose processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The current general purpose processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated general purpose processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.42.0.*

Workload Manager enabled

Get/Set: Whether or not WorkLoad Manager is allowed to change processing weight related attributes.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that WorkLoad Manager is allowed to change processing weight related attributes for this CPC Image object.

  HWMCA_FALSE
  Indicates that WorkLoad Manager is not allowed to change processing weight related attributes for this CPC Image object.

This attribute and the various capping attributes are mutually exclusive and cannot be enabled at the same time. Therefore in order to enable this attribute it may be necessary to first disable any capping attribute that is currently enabled. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and all capping attributes being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.40.0.*

Defined capacity

Get/Set: The defined capacity expressed in terms of Millions of Service Units (MSU)s per hour. MSUs is a measure of processor resource consumption. The amount of MSUs a logical partition consumes is dependent on the model, the number of logical processors available to the partition, and the amount of time the logical partition is dispatched. The defined capacity value specifies how much capacity the logical partition is to be managed to by WorkLoad Manager for the purpose of software pricing.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  No defined capacity is specified for this logical partition.

  1 - nnnn
  Represents the amount of defined capacity specified for this logical partition.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.43.0.*

Cluster name

Get: LPAR cluster name.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.50.0*

Partition identifier

Get: The partition identifier for the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.51.0*

Initial Application Assist Processor processing weight

Get/Set: The relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1 - 999
  Represents the relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.60.0.*

Initial Application Assist Processor processing weight capped

Get/Set: Whether or not the initial processing weight for Application Assist Processor (AAP) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial Application Assist Processor (AAP) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Application Assist Processor (AAP) processor resources, regardless of the availability of excess Application Assist Processor (AAP) processor resources.

  HWMCA_FALSE (0)
  Indicates that the initial Application Assist Processor (AAP) processor processing weight for the CPC Image is not capped. It represents the share of Application Assist Processor (AAP) processor resources guaranteed to a logical partition when all Application Assist Processor (AAP) processor resources are in use. Otherwise, when excess Application Assist Processor (AAP) processor resources are available, the logical partition can use them if necessary.

  Note: The initial Application Assist Processor (AAP) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it may be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.61.0.*

Minimum Application Assist Processor processing weight

Get/Set: The minimum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1-999
  Represents the minimum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.62.0.*

Maximum Application Assist Processor processing weight

Get/Set: The maximum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

0
CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

1-999
Represents the maximum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.63.0.*

Current Application Assist Processor processing weight

Get: The current relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1-999
  Represents the current relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.64.0.*

Current Application Assist Processor processing weight capped

Get: Whether or not the current Application Assist Processor (AAP) processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the current Application Assist Processor (AAP) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the current Application Assist Processor (AAP) processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The current Application Assist Processor (AAP) processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.65.0.*

Initial Integrated Facility for Linux processing weight

Get/Set: The relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

1-999
Represents the relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.66.0.*

Initial Integrated Facility for Linux processing weight capped

Get/Set: Whether or not the initial processing weight for Integrated Facility for Linux (IFL) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial Integrated Facility for Linux (IFL) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Integrated Facility for Linux (IFL) processor resources, regardless of the availability of excess Integrated Facility for Linux (IFL) processor resources.

  HWMCA_FALSE (0)
  Indicates that the initial Integrated Facility for Linux (IFL) processor processing weight for the CPC Image is not capped. It represents the share of Integrated Facility for Linux (IFL) processor resources guaranteed to a logical partition when all Integrated Facility for Linux (IFL) processor resources are in use. Otherwise, when excess Integrated Facility for Linux (IFL) processor resources are available, the logical partition can use them if necessary.

  Note: The initial Integrated Facility for Linux (IFL) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.67.0.*

Minimum Integrated Facility for Linux processing weight

Get/Set: The minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  1-999
  Represents the minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.68.0.*

Maximum Integrated Facility for Linux processing weight

Get/Set: The maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object. The maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  1-999
  Represents the maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.69.0.*

Current Integrated Facility for Linux processing weight

Get: The current relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  1-999
  Represents the current relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.70.0.*

Current Integrated Facility for Linux Processing weight capped

Get: Whether or not the current Integrated Facility for Linux (IFL) processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the current Integrated Facility for Linux (IFL) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the current Integrated Facility for Linux (IFL) processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The current Integrated Facility for Linux (IFL) processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.71.0.*

Initial z Integrated Information Processors processing weight

Get/Set: The relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  1-999
  Represents the relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.81.0.*

Initial z Integrated Information Processors processing weight capped

Get/Set: Whether or not the initial processing weight for z Integrated Information Processors (zIIP) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial z Integrated Information Processors (zIIP) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of z Integrated Information Processors (zIIP) processor resources, regardless of the availability of excess z Integrated Information Processors (zIIP) processor resources.

  HWMCA_FALSE (0)
  Indicates that the initial z Integrated Information Processors (zIIP) processor processing weight for the CPC Image is not capped. It represents the share of z Integrated Information Processors (zIIP) processor resources guaranteed to a logical partition when all z Integrated Information Processors (zIIP) processor resources are in use. Otherwise, when excess z Integrated Information Processors (zIIP) processor resources are available, the logical partition can use them if necessary.

  Note: The initial z Integrated Information Processors (zIIP) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it may be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.82.0.*

Minimum z Integrated Information Processors processing weight

Get/Set: The minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

0
CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

1-999
Represents the minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.83.0.*

Maximum z Integrated Information Processors Processing Weight

Get/Set: The maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  1-999
  Represents the maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.84.0.*

Current z Integrated Information Processors processing weight

Get: The current relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  1-999
  Represents the current relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.85.0.*

Current z Integrated Information Processors processing weight capped

Get: Whether or not the current z Integrated Information Processors (zIIP) processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

HWMCA_TRUE
Indicates that the current z Integrated Information Processors (zIIP) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

HWMCA_FALSE
Indicates that the current z Integrated Information Processors (zIIP) processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

Note: The current z Integrated Information Processors (zIIP) processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.86.0.*

Group profile name

Get/Set: Defines the name of the group capacity profile that is being used for the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.93.0.*.*

Program Status Word (PSW) information

Get: An XML string that describes the current PSW information for each logical processor associated with the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING

  Note: Refer to Appendix F, “XML descriptions,” on page 209 for a detailed description of this XML data.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.150.0

IPL Token

Get: The Token used in the last IPL.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.164.0

Group Profile capacity

Get/Set: The current capacity value of the Group Profile the CPC Image object is associated with.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.192.0

Last Used Load Address

Get: The load addressed used in the last IPL.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.201.0

Last Used Load Parameter

Get: The load parameter used in the last IPL.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.202.0

Absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.217.0.*

Absolute capping value

Get/Set: The value used for absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  1-nnnnn
  String form of an integer representing the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.218.0.*

Application Assist Processor absolute capping type

Get/Set: The type of absolute capping to perform for Application Assist Processor (AAP) processors.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Application Assist Processor (AAP) processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.219.0.*

Application Assist Processor absolute capping value

Get/Set: The value used for Application Assist Processor (AAP) absolute capping.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  1-nnnnn
  String form of an integer representing the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.220.0.*

Integrated Facility for Linux absolute capping type

Get/Set: The type of absolute capping to perform for Integrated Facility for Linux (IFL) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Integrated Facility for Linux (IFL) processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.221.0.*

Integrated Facility for Linux absolute capping value

Get/Set: The value used for Integrated Facility for Linux (IFL) absolute capping (if enabled).

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
None

1-nnnnn
String form of an integer representing the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.222.0.*

z Integrated Information Processor absolute capping type

Get/Set: The type of absolute capping to perform for z Integrated Information Processor (zIIP) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of z Integrated Information Processor (zIIP) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.225.0.*

z Integrated Information Processor absolute capping value

Get/Set: The value used for z Integrated Information Processor (zIIP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

1-nnnnn
String form of an integer representing the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.226.0.*

Initial Container Based Processor processing weight

Get/Set: The relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Get: HWMCA_TYPE_INTEGER

0
CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

1-999
Represents the relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

1-999
A value 1 - 999 used to define the relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object.

Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Container Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.295.0.*

Initial Container Based Processor processing weight capped

Get/Set: Enables/disables the use of the basic CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE (1)
Indicates that the initial Container Based Processor (CBP) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Container Based Processor (CBP) resources, regardless of the availability of excess Container Based Processor (CBP) resources.

HWMCA_FALSE (0)
Indicates that the initial Container Based Processor (CBP) processing weight for the CPC Image is not capped. It represents the share of Container Based Processor (CBP) resources guaranteed to a logical partition when all Container Based Processor (CBP) resources are in use. Otherwise, when excess Cloud Based Processor (CBP) resources are available, the logical partition can use them if necessary.

Note: The initial Container Based Processor (CBP) processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP). This attribute and the Workload manager enabled attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable the Workload manager enabled attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the Workload manager enabled attribute being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.296.0.*

Minimum Container Based Processor processing weight

Get/Set: The minimum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

  1-999
  Represents the minimum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  1-999
  A value 1 - 999 used to define the minimum relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Cloud Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.297.0.*

Maximum Container Based Processor processing weight

Get/Set: The maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object. The maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

  1-999
  Represents the maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

1-999
A value 1 - 999 used to define the maximum relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object.

Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Container Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.298.0.*

Current Container Based Processor processing weight

Get: The current relative amount of shared Container Based (CBP) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

0
CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

1-999
Represents the current relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.299.0.*

Current Container Based Processor processing weight capped

Get: Whether or not the current Container Based Processor (CBP) processing weight is a limit or a target.

- Data type returned on Get: HWMCA_TYPE_INTEGER

HWMCA_TRUE
Indicates that the current Container Based Processor (CBP) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

HWMCA_FALSE
Indicates that the current Container Based Processor (CBP) processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

Note: The current Container Based Processor (CBP) processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.300.0.*

Container Based Processor absolute capping type

Get/Set: The type of absolute capping to perform for Container Based Processors (CBP).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

0
None.

1
Absolute capping in number of CBP processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.301.0.*

Container Based Processor absolute capping value

Get/Set: The value used for Container Based Processor (CBP) absolute capping.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

1-nnnnn
String form of an integer representing the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.302.0.*

Basic CPU counter authorization control

Get/Set: Enables/disables the use of the basic CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
The authorization control is enabled.

HWMCA_FALSE
The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.168.0.*.*

Problem state CPU counter authorization control

Get/Set: Enables/disables the use of the problem state CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
The authorization control is enabled.

HWMCA_FALSE
The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.169.0.*.*

Crypto activity CPU counter authorization control

Get/Set: Enables/disables the use of the crypto activity CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
The authorization control is enabled.

HWMCA_FALSE
The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.170.0.*.*

Extended CPU counter authorization control

Get/Set: Enables/disables the use of the extended CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
The authorization control is enabled.

HWMCA_FALSE
The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.171.0.*.*

Basic CPU sampling authorization control

Get/Set: Enables/disables the use of the basic CPU sampling facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
The authorization control is enabled.

HWMCA_FALSE
The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.173.0.*.*

Diagnosis CPU sampling authorization control

Get/Set: Enables/disables the use of the diagnosis CPU sampling facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.174.0.*.*

Permit AES key import functions

Get/Set: Enables/disables the importing of AES keys for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The importing of AES keys is enabled.

  HWMCA_FALSE
  The importing of AES keys is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.183.0.*.*

Permit DEA key import functions

Get/Set: Enables/disables the importing of DEA keys for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The importing of DEA keys is enabled.

  HWMCA_FALSE
  The importing of DEA keys is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.184.0.*.*

Global performance data control

Get/Set: Enables/disables the global performance data control setting for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The global performance data control is enabled.

  HWMCA_FALSE
  The global performance data control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.211.0.*.*

Input/Output configuration control

Get/Set: Enables/disables the I/O configuration control setting for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The I/O configuration control is enabled.

  HWMCA_FALSE
  The I/O configuration control is disabled.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.212.0.*.*

Cross partition authority control

Get: The cross partition authority control setting for the associated CPC Image.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The cross partition authority control is enabled.

  HWMCA_FALSE
  The cross partition authority control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.213.0.*.*

Logical partition isolation control

Get/Set: Enables/disables the logical partition isolation control setting for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The logical partition isolation control is enabled.

  HWMCA_FALSE
  The logical partition isolation control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.214.0.*.*

Group absolute capping type

Get/Set: The type of group absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.227.0.*.*

Group absolute capping value

Get/Set: The value used for group absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1-nnnnn
  Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.228.0.*.*

Integrated Facility for Linux group absolute capping type

Get/Set: The type of group absolute capping to perform for Integrated Facility for Linux (IFL) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Integrated Facility for Linux (IFL) processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.231.0.*.*

Integrated Facility for Linux group absolute capping value

Get/Set: The value used for Integrated Facility for Linux (IFL) absolute capping (if enabled).

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1-nnnnn
  Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.232.0.*.*

z Integrated Information Processor group absolute capping type

Get/Set: The type of group absolute capping to perform for z Integrated Information Processor (zIIP) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of z Integrated Information Processor (zIIP) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.233.0.*.*

z Integrated Information Processor group absolute capping value

Get/Set: The value used for z Integrated Information Processor (zIIP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  Absolute capping not enabled.

  1-nnnnn
  Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.234.0.*.*

LPAR Controls

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all of the LPAR Control related attributes for the associated CPC Image. These attributes are:

  – Initial processing weight
  – Initial processing weight capped
  – Minimum processing weight
  – Maximum processing weight
  – Current processing weight
  – Current processing weight capped
  – Workload manager enabled
  – Defined capacity
  – Initial Application Assist Processor processing weight

  - Initial Application Assist Processor processing weight capped
  - Minimum Application Assist Processor processing weight
  - Maximum Application Assist Processor processing weight
  - Current Application Assist Processor processing weight
  - Current Application Assist Processor processing weight capped
  - Initial Internal Coupling Facility processing weight
  - Initial Internal Coupling Facility processing weight capped
  - Minimum Internal Coupling Facility processing weight
  - Maximum Internal Coupling Facility processing weight
  - Current Internal Coupling Facility processing weight
  - Current Internal Coupling Facility processing weight capped
  - Initial Integrated Facility for Linux processing weight
  - Initial Integrated Facility for Linux processing weight capped
  - Minimum Integrated Facility for Linux processing weight
  - Maximum Integrated Facility for Linux processing weight
  - Current Integrated Facility for Linux processing weight
  - Current Integrated Facility for Linux processing weight capped
  - Initial z Integrated Information Processors processing weight
  - Initial z Integrated Information Processors processing weight capped
  - Minimum z Integrated Information Processors processing weight
  - Maximum z Integrated Information Processors processing weight
  - Current z Integrated Information Processors processing weight
  - Current z Integrated Information Processors processing weight capped
  - Absolute capping type Absolute capping value
  - Application Assist Processor absolute capping type
  - Application Assist Processor absolute capping value
  - Integrated Facility for Linux absolute capping type
  - Integrated Facility for Linux absolute capping value
  - z Integrated Information Processor absolute capping type
  - z Integrated Information Processor absolute capping value
  - Internal Coupling Facility absolute capping type
  - Internal Coupling Facility absolute capping value
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.236.0.*

Group Capacity

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings..

  The sequence contains all of the Group capacity related attributes for the associated CPC Image. These attributes are:

  - Group profile name
  - Group profile capacity

  - Absolute capping type
  - Absolute capping value
  - Application Assist Processor absolute capping type
  - Application Assist Processor absolute capping value
  - Integrated Facility for Linux absolute capping type
  - Integrated Facility for Linux absolute capping value
  - z Integrated Information Processor absolute capping type
  - z Integrated Information Processor absolute capping value

  The group profile name for each CPC Image object will be contained in the returned data, but all of the other attributes are optionally returned based on whether the CPC Image is currently activated and if absolute capping is supported.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.237.0.*

Initial Virtual Flash Memory

Get: The initial amount of Virtual Flash Memory in Gigabytes (GB) allocated for a CPC image object

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.246.0.*

Maximum Virtual Flash Memory

Get: The maximum amount of Virtual Flash Memory in Gigabytes (GB) that can be allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.247.0.*

Current Virtual Flash Memory

Get: The current amount of Virtual Flash Memory in Gigabytes (GB) allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.248.0.*

Central storage origin

Get: The origin, in megabytes, of central storage in memory for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.249.0.*

Initial central storage

Get: The initial amount, in megabytes, of central storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.250.0.*

Current central storage

Get: The current amount, in megabytes, of central storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.251.0.*

Maximum central storage

Get: The maximum amount, in megabytes, of central storage that can be allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.252.0.*

Central storage gap

Get: The gap, in megabytes, between the central storage for a CPC Image object and the start of the central storage for the next CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.253.0.*

Initial expanded storage origin

Get: The origin, in megabytes, of initial expanded storage in memory for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.254.0.*

Current initial expanded storage

Get: The current amount, in megabytes, of initial expanded storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.255.0.*

Initial expanded storage gap

Get: The gap, in megabytes, between the initial expanded storage for a CPC Image object and the start of the expanded storage for the next CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.256.0.*

Reserved expanded storage origin

Get:The origin, in megabytes, of reserved expanded storage in memory for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.257.0.*

Current reserved expanded storage

Get: The current amount, in megabytes, of reserved expanded storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.258.0.*

Reserved expanded storage gap

Get: The gap, in megabytes, between the reserved expanded storage for a CPC Image object and the start of the expanded storage for the next CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.259.0.*

Expanded storage origin

Get: The origin, in megabytes, of expanded storage in memory for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.260.0.*

Initial expanded storage

Get: The initial amount, in megabytes, of expanded storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.261.0.*

Current expanded storage

Get: The current amount, in megabytes, of expanded storage that is allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.262.0.*

Maximum expanded storage

Get: The maximum amount, in megabytes, of expanded storage that can be allocated for a CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.263.0.*

Expanded storage gap

Get: The gap, in megabytes, between the expanded storage for a CPC Image object and the start of the expanded storage for the next CPC image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.264.0.*

Storage information

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE. The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the HwmcaEnhancedGet interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all the storage related attributes for a CPC Image object. These attributes are:

  - Central storage origin
  - Initial central storage
  - Current central storage
  - Maximum central storage
  - Central storage gap
  - Initial expanded storage origin
  - Current initial expanded storage
  - Initial expanded storage gap
  - Reserved expanded storage origin
  - Current reserved expanded storage
  - Reserved expanded storage gap
  - Expanded storage origin
  - Initial expanded storage
  - Current expanded storage
  - Maximum expanded storage
  - Expanded storage gap
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.267.0.*

Container Based Processor group absolute capping type

Get/Set: The type of group absolute capping to perform for Container Based Processors (CBP).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None.

  1
  Absolute capping in number of CBP processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.303.0.*

Container Based Processor group absolute capping value

Get/Set: The value used for Container Based Processor (CBP) absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None.

  1-nnnnn
  Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.304.0.*

Encrypt ECC functions

Get/Set: LPAR security "ECC key" bit (control enablement of digital signatures).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  ECC key is off.

  1
  ECC key is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.311.0.*

Sub-capacity boost

Get: Sub-capacity boost value for the CPC Image.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Sub-capacity boost is off.

  1
  Sub-capacity boost is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.312.0.*

zIIP-capacity boost

Get: The zIIP capacity boost value for the CPC Image.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  zIIP-capacity boost is off.

  1
  zIIP-capacity boost is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.313.0.*

Secure execution

Get: The secure execution flag for the CPC Image.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Secure execution is off.

  1
  Secure execution is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.314.0.*

Last used verify software signature

Get: The last used verify software signature flag for the CPC Image.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Last used verify software signature is off.

  1
  Last used verify software signature is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.316.0.*

CPC image relationships

CPC (H/W image)

A CPC image is a member of a CPC (H/W image); there can be from ***1*** to ***n*** CPC Images. ***N*** is determined by the Licensed Internal Code.

Software image

A CPC image has one software image running in it.

Note: Some operating systems support running guest operating systems within themselves.

CPC image commands

Activate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.1 (HWMCA_ACTIVATE_COMMAND)

Reset normal

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.4 (HWMCA_RESETNORMAL_COMMAND)

Deactivate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.2 (HWMCA_DEACTIVATE_COMMAND)

Start

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.5 (HWMCA_START_COMMAND)

Stop

SNMP MIB Name - 1.3.6.1.4.1.2.6.42.4.6 (HWMCA_STOP_COMMAND)

PSW restart

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.7 (HWMCA_PSWRESTART_COMMAND)

Send operating system command

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.3 (HWMCA_SEND_OPSYS_COMMAND)

Load

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.10 (HWMCA_LOAD_COMMAND)

Reset clear

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.12 (HWMCA_RESETCLEAR_COMMAND)

External interrupt

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.19 (HWMCA_EXTERNAL_INTERRUPT_COMMAND)

SCSI load

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.20 (HWMCA_SCSI_LOAD_COMMAND)

SCSI dump

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.21 (HWMCA_SCSI_DUMP_COMMAND)

CPC image notifications

Message (operating system - HWMCA_EVENT_MESSAGE)

An HWMCA_TYPE_INTEGER that specifies whether the message is a hardware or operating system message (HWMCA_HARDWARE_MESSAGE or HWMCA_OPSYS_MESSAGE).

For operating system messages:

- An HWMCA_TYPE_OCTETSTRING that specifies the new or refreshed operating system message text.
- An HWMCA_TYPE_OCTETSTRING that specifies the message identifier of the new operating system message.
- An HWMCA_TYPE_OCTETSTRING that specifies the date of the new operating system message.
- An HWMCA_TYPE_OCTETSTRING that specifies the time of the new operating system message.
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message should cause the alarm to be sounded (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a priority message or not (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a held message or not (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_OCTETSTRING that specifies the prompt text that should be associated with the new operating system message or an HWMCA_TYPE_NULL indicating that there is no prompt text for this new operating system message.
- An HWMCA_TYPE_OCTETSTRING that specifies the name of the operating system that generated this new operating system message or an HWMCA_TYPE_NULL indicating that there is no operating system name associated with this new operating system message.
- An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).

Status change (HWMCA_EVENT_STATUS_CHANGE)

- An HWMCA_TYPE_INTEGER that specifies the new status value
- An HWMCA_TYPE_INTEGER that specifies the old status value.

Object name change (HWMCA_EVENT_NAME_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new object name

- An HWMCA_TYPE_OCTETSTRING that specifies the old object name.

Object activation profile change (HWMCA_EVENT_ACTIVATE_PROF_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new activation profile name
- An HWMCA_TYPE_OCTETSTRING that specifies the old activation profile name.

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the *HwmcaGet* to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Object entered an exception state (HWMCA_EVENT_EXCEPTION_STATE)

- An HWMCA_TYPE_INTEGER that specifies whether the object is entering into an exception state (HWMCA_TRUE) or leaving an exception state (HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies the status value for the object.

Disabled wait (HWMCA_EVENT_DISABLED_WAIT)

- An HWMCA_TYPE_OCTETSTRING for the name of the Defined CPC that is associated with the CPC Image that entered a disabled wait state.
- An HWMCA_TYPE_OCTETSTRING for the disabled wait PSW value.
- An HWMCA_TYPE_INTEGER for the partition identifier of the CPC Image that entered a disabled wait state.
- An HWMCA_TYPE_INTEGER for number of the processor that entered a disabled wait state.
- An HWMCA_TYPE_OCTETSTRING for the serial number of the Defined CPC that is associated with the CPC Image that entered a disabled wait state.
- An HWMCA_TYPE_OCTETSTRING that specifies the name of the object that the event pertains to (in this case a CPC Image object).

Coupling facility

Coupling facility name bindings

Coupling facility object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.x.x.*

Where ***x.x.*** equals the attribute identifier for the object and an ***** equals a unique number for that specific instance of the Coupling Facility Object.

Coupling facility attributes

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.1.0.*

Parent name

Get (CPC's logical name):

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.2.0.*

Status error

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object is in a state which is not an acceptable status.

  HWMCA_FALSE
  Object is in an acceptable status state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.7.0.*

Busy

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object in a busy state (currently performing a task).

  HWMCA_FALSE
  Object not in a busy state.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.8.0.*

Message indicator

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Object has an operating system message.

  HWMCA_FALSE
  Object does not have an operating system message.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.9.0.*

Status

Get:

- Data type returned on Get: HWMCA_TYPE_INTEGER

  One of the following bit values will be set to on:

  – HWMCA_STATUS_OPERATING
  – HWMCA_STATUS_NOT_OPERATING
  – HWMCA_STATUS_NOT_ACTIVATED
  – HWMCA_STATUS_EXCEPTIONS
  – HWMCA_STATUS_STATUS_CHECK
  – HWMCA_STATUS_POWERSAVE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.10.0.*

Acceptable status

Get/Set:

- Data type returned on Get: HWMCA_TYPE_INTEGER
- Data type for Set: HWMCA_TYPE_INTEGER

- One or more of the following bit values will be set to on:
  - HWMCA_STATUS_OPERATING
  - HWMCA_STATUS_NOT_OPERATING
  - HWMCA_STATUS_NOT_ACTIVATED
  - HWMCA_STATUS_EXCEPTIONS
  - HWMCA_STATUS_STATUS_CHECK
  - HWMCA_STATUS_POWERSAVE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.11.0.*

Activation profile name

Get (always the Image profile):

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.13.0.*

Last used activation profile

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING or HWMCA_TYPE_NULL
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.14.0.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_CF_OBJECT
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.22.0.*

Initial processing weight

Get/Set: The relative amount of shared general purpose processor resources initially allocated to the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purposee processor.

  1 - 999
  Represents the relative amount of shared general purpose processor resources initially allocated to the Coupling Facility object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

  Note: The setting of this attribute is only valid for Coupling Facility objects that represent a logical partition with at least one not dedicated general purpose processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.30.0.*

Initial processing weight capped

Get/Set: Whether or not the initial processing weight for general purpose processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE (1)
Indicates that the initial general purpose processor processing weight for the Coupling Facility object is capped. It represents the logical partition's maximum share of general purpose processor resources, regardless of the availability of excess general purpose processor resources.

HWMCA_FALSE (0)
Indicates that the initial general purpose processor processing weight for the Coupling Facility is not capped. It represents the share of general purpose processor resources guaranteed to a logical partition when all general purpose processor resources are in use. Otherwise, when excess general purpose processor resources are available, the logical partition can use them if necessary.

Note: The initial general purpose processor processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purpose processor.

This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.31.0.*

Minimum processing weight

Get/Set: The minimum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the minimum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and less than or equal to the Initial Processing Weight used to define the minimum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

  Note: The setting of this attribute is only valid for Coupling Facility objects that represent a logical partition with at least one not dedicated general purpose processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.38.0.*

Maximum processing weight

Get/Set: The maximum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the maximum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and greater than or equal to the Initial Processing Weight used to define the maximum relative amount of shared general purpose processor resources allocated to the Coupling Facility object.

A value of zero can also be specified to indicate that there is no maximum value for the processing weight.

Note: The setting of this attribute is only valid for Coupling Facility objects that represent a logical partition with at least one not dedicated general purpose processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.39.0.*

Current processing weight

Get: The relative amount of shared general purpose processor resources currently allocated to the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purpose processor.

  1 - 999
  Represents the relative amount of shared general purpose processor resources currently allocated to the Coupling Facility object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.41.0.*

Current processing weight capped

Get: Whether or not the current general purpose processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the current general purpose processing weight for the Coupling Facility object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the general purpose current processing weight for the Coupling Facility is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The current general purpose processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the Coupling Facility does not represent a logical partition or the Coupling Facility does not represent a logical partition with at least one not dedicated general purpose processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.42.0.*

WorkLoad manager enabled

Get/Set: Whether or not WorkLoad Manager is allowed to change processing weight related attributes.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that WorkLoad Manager is allowed to change processing weight related attributes for this CPC Image object.

  HWMCA_FALSE
  Indicates that WorkLoad Manager is not allowed to change processing weight related attributes for this CPC Image object.

  This attribute and the various capping attributes are mutually exclusive and cannot be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable any capping attribute that is currently enabled. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and all capping attributes being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.40.0.*

Defined capacity

Get/Set: The defined capacity expressed in terms of Millions of Service Units (MSU)s per hour. MSUs is a measure of processor resource consumption. The amount of MSUs a logical partition consumes is dependent on the model, the number of logical processors available to the partition, and the amount of time the logical partition is dispatched. The defined capacity value specifies how much capacity the logical partition is to be managed to by WorkLoad Manager for the purpose of software pricing.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  No defined capacity is specified for this logical partition.

  1 - nnnn
  Represents the amount of defined capacity specified for this logical partition.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.43.0.*

Partition identifier

Get: The partition identifier for the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.51.0*

Initial Internal Coupling Facility processing weight

Get/Set: The relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.72.0.*

Initial Internal Coupling Facility processing weight capped

Get/Set: Whether or not the initial processing weight for Internal Coupling Facility (ICF) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial Internal Coupling Facility (ICF) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Internal Coupling Facility (ICF) processor resources, regardless of the availability of excess Internal Coupling Facility (ICF) processor resources.

  HWMCA_FALSE (0)
  Indicates that the initial Internal Coupling Facility (ICF) processor processing weight for the CPC Image is not capped. It represents the share of Internal Coupling Facility (ICF) processor resources guaranteed to a logical partition when all Internal Coupling Facility (ICF) processor resources are in

use. Otherwise, when excess Internal Coupling Facility (ICF) processor resources are available, the logical partition can use them if necessary.

Note: The initial Internal Coupling Facility (ICF) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

This attribute and the **Workload manager enabled** attribute are mutually exclusive and cannot both be enabled at the same time. Therefore in order to enable this attribute it might be necessary to first disable the **Workload manager enabled** attribute. It is also possible to use a value of -1 when setting this attribute, which will result in this attribute being enabled and the **Workload manager enabled** attribute being disabled automatically in a single operation.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.73.0.*

Minimum Internal Coupling Facility processing weight

Get/Set: The minimum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the minimum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.74.0.*

Maximum Internal Coupling Facility processing weight

Get/Set: The maximum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the maximum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.75.0.*

Current Internal Coupling Facility processing weight

Get: The current relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the current relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.76.0.*

Current Internal Coupling Facility processing weight capped

Get: Whether or not the current Internal Coupling Facility (ICF) processing weight is a limit or a target.

- Data type for Get: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the current Internal Coupling Facility (ICF) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the current Internal Coupling Facility (ICF) processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The current Internal Coupling Facility (ICF) processing weight capped attribute cannot be set and the value returned for a get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.77.0.*

Program Status Word (PSW) information

Get: An XML string that describes the current PSW information for each logical processor associated with the Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING

  Note: Refer to Appendix F, “XML descriptions,” on page 209 for a detailed description of this XML data.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.150.0

Internal Coupling Facility absolute capping type

Get/Set: The type of absolute capping to perform for Internal Coupling Facility (ICF) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Internal Coupling Facility (ICF) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.223.0.*

Internal Coupling Facility absolute capping value

Get/Set: The value used for Internal Coupling Facility (ICF) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

1-nnnn
Represents the number of Internal Coupling Facility (ICF) processors when capping in number of processors is enabled.

Note: Though this is an integer value, it must be specified within an HWMCA_TYPE_OCTETSTRING data type. This was done in case future absolute capping types require fractional units.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.224.0.*

Internal Coupling Facility group absolute capping type

Get/Set: The type of group absolute capping to perform for Internal Coupling Facility (ICF) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of Internal Coupling Facility (ICF) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.229.0.*

Internal Coupling Facility group absolute capping value

Get/Set: The value used for Internal Coupling Facility (ICF) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
Absolute capping not enabled.

1-nnnnn
Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.230.0.*

Initial Virtual Flash Memory

Get: The initial amount of Virtual Flash Memory in Gigabytes (GB) allocated for a Coupling Facility object

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.246.0.*

Maximum Virtual Flash Memory

Get: The maximum amount of Virtual Flash Memory in Gigabytes (GB) that can be allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.247.0.*

Current Virtual Flash Memory

Get: The current amount of Virtual Flash Memory in Gigabytes (GB) allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.248.0.*

Central storage origin

Get: The origin, in megabytes, of central storage in memory for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.249.0.*

Initial central storage

Get: The initial amount, in megabytes, of central storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.250.0.*

Current central storage

Get: The current amount, in megabytes, of central storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.251.0.*

Maximum central storage

Get: The maximum amount, in megabytes, of central storage that can be allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.252.0.*

Central storage gap

Get: The gap, in megabytes, between the central storage for a Coupling Facility object and the start of the central storage for the next Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.253.0.*

Initial expanded storage origin

Get: The origin, in megabytes, of initial expanded storage in memory for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.254.0.*

Current initial expanded storage

Get: The current amount, in megabytes, of initial expanded storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.255.0.*

Initial expanded storage gap

Get: The gap, in megabytes, between the initial expanded storage for a Coupling Facility object and the start of the expanded storage for the next Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.256.0.*

Reserved expanded storage origin

Get:The origin, in megabytes, of reserved expanded storage in memory for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.257.0.*

Current reserved expanded storage

Get: The current amount, in megabytes, of reserved expanded storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.258.0.*

Reserved expanded storage gap

Get: The gap, in megabytes, between the reserved expanded storage for a Coupling Facility object and the start of the expanded storage for the next Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.259.0.*

Expanded storage origin

Get: The origin, in megabytes, of expanded storage in memory for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.260.0.*

Initial expanded storage

Get: The initial amount, in megabytes, of expanded storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.261.0.*

Current expanded storage

Get: The current amount, in megabytes, of expanded storage that is allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.262.0.*

Maximum expanded storage

Get: The maximum amount, in megabytes, of expanded storage that can be allocated for a Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.263.0.*

Expanded storage gap

Get: The gap, in megabytes, between the expanded storage for a Coupling Facility object and the start of the expanded storage for the next Coupling Facility object.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.2.0.264.0.*

Storage information

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE. The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the HwmcaEnhancedGet interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings.

  The sequence contains all the storage related attributes for a Coupling Facility object. These attributes are:

  - Central storage origin
  - Initial central storage
  - Current central storage
  - Maximum central storage
  - Central storage gap
  - Initial expanded storage origin
  - Current initial expanded storage
  - Initial expanded storage gap
  - Reserved expanded storage origin
  - Current reserved expanded storage
  - Reserved expanded storage gap
  - Expanded storage origin
  - Initial expanded storage
  - Current expanded storage
  - Maximum expanded storage
  - Expanded storage gap
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.1.0.267.0.*

Coupling facility relationships

CPC (H/W image)

A coupling facility image is a member of a CPC (H/W image) there can be from ***1*** to ***n*** coupling facility images. ***N*** is determined by the Licensed Internal Code.

Coupling Facility Control Code (CFCC)

A coupling facility image is running the Coupling Facility Control Code to perform the Coupling Facility functions.

Coupling facility commands

Activate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.1 (HWMCA_ACTIVATE_COMMAND)

Deactivate

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.2 (HWMCA_DEACTIVATE_COMMAND)

Send operating system command

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.4.3 (HWMCA_SEND_OPSYS_COMMAND)

Coupling facility notifications

Message (operating system - HWMCA_EVENT_MESSAGE)

An HWMCA_TYPE_INTEGER that specifies whether the message is a hardware or operating system message (HWMCA_HARDWARE_MESSAGE or HWMCA_OPSYS_MESSAGE).

For operating system messages:

- An HWMCA_TYPE_OCTETSTRING that specifies the new or refreshed operating system message text.
- An HWMCA_TYPE_OCTETSTRING that specifies the message identifier of the new operating system message.

- An HWMCA_TYPE_OCTETSTRING that specifies the date of the new operating system message.
- An HWMCA_TYPE_OCTETSTRING that specifies the time of the new operating system message.
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message should cause the alarm to be sounded (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a priority message or not (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies whether the new operating system message is a held message or not (HWMCA_TRUE or HWMCA_FALSE).
- An HWMCA_TYPE_OCTETSTRING that specifies the prompt text that should be associated with the new operating system message or an HWMCA_TYPE_NULL indicating that there is no prompt text for this new operating system message.
- An HWMCA_TYPE_OCTETSTRING that specifies the name of the operating system that generated this new operating system message or an HWMCA_TYPE_NULL indicating that there is no operating system name associated with this new operating system message.
- An HWMCA_TYPE_INTEGER that specifies whether the message is a new (HWMCA_FALSE) or refresh message (HWMCA_TRUE).

Status change (HWMCA_EVENT_STATUS_CHANGE)

- An HWMCA_TYPE_INTEGER that specifies the new status value
- An HWMCA_TYPE_INTEGER that specifies the old status value.

Object name change (HWMCA_EVENT_NAME_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new object name
- An HWMCA_TYPE_OCTETSTRING that specifies the old object name.

Object activation profile change (HWMCA_EVENT_ACTIVATE_PROF_CHANGE)

- An HWMCA_TYPE_OCTETSTRING that specifies the new activation profile name
- An HWMCA_TYPE_OCTETSTRING that specifies the old activation profile name.

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the *HwmcaGet* to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Object entered an exception state (HWMCA_EVENT_EXCEPTION_STATE)

- An HWMCA_TYPE_INTEGER that specifies whether the object is entering into an exception state (HWMCA_TRUE) or leaving an exception state (HWMCA_FALSE).
- An HWMCA_TYPE_INTEGER that specifies the status value for the object.

Reset activation profile object

Reset activation profile name bindings

Reset activation profile object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.x.x.y.z

Where ***x.x.*** equals the attribute identifier for the object, ***y*** equals a unique number for the specific instance of the CPC Object, and ***z*** equals a unique number for the specific instance of the Reset Activation Profile.

Reset activation profile attributes

Contents

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings. The sequence contains all of the attributes for a Reset activation profile object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.*y.z* (where *y* equals a unique number for the specific instance of the CPC Object, and *z* equals a unique number for the specific instance of the Reset activation profile)

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.1.0.*.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_ACT_PROFILE_RESET
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.22.0.*.*

IOCDS

Get/Set:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A value of an empty string is used to indicate that the Reset Activation Profile will use the currently active IOCDS.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.27.0.*.*

Processor running time type

Get/Set: Defines whether the processor running time is dynamically determined by the system or set to a constant value for the Defined CPC object.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0 (HWMCA_DETERMINED_SYSTEM)
  The processor running is dynamically determined by the system.

  1 (HWMCA_DETERMINED_USER)
  The processor running time is set to a constant value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.78.0.*.*

Processor running time

Get/Set: Defines the amount of continuous time allowed for logical processors to perform jobs on shared processors for the Defined CPC object.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  A value between 1 and 100 for the user defined processor running time.

Note: This value can only be set if the processor running time type is set to 1 (HWMCA_DETERMINED_USER). Additionally, this value will always be returned as zero if the processor running time type is set to 0 (HWMCA_DETERMINED_SYSTEM).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.79.0.*.*

End timeslice if CPC image enters a wait state

Get/Set: Defines whether CPC Images lose their share of running time when they enter a wait state. (Only supported for Defined CPC objects earlier than version 2.14.0).

- Data type for Get/Set: HWMCA_TYPE_INTEGER

HWMCA_TRUE
Indicates that a CPC Image should lose its share of running time when it enters a wait state.

HWMCA_FALSE
Indicates that a CPC Image should not lose its share of running time when it enters a wait state.

Note: This value can only be set if the processor running time type is set to 1 (HWMCA_DETERMINED_USER). Additionally, this value will always be returned as zero if the processor running time type is set to 0 (HWMCA_DETERMINED_SYSTEM).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.80.0.*.*

Description

Get/Set: The description of the profile with a maximum length of 51 (including the null terminator).

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.5.0.203.0.*.*

Reset activation profile notifications

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the HwmcaGet to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Image activation profile object

Image activation profile name bindings

Image activation profile object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.x.x.y.z

Where ***x.x.*** equals the attribute identifier for the object, ***y*** equals a unique number for the specific instance of the CPC Object, and ***z*** equals a unique number for the specific instance of the Image Activation Profile.

Image activation profile attributes

Contents

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

he data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings. The sequence contains all of the attributes for a Image activation profile object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.*y*.*z* (where *y* equals a unique number for the specific instance of the CPC Object, and *z* equals a unique number for the specific instance of the Image activation profile)

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.1.0.*.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_ACT_PROFILE_IMAGE
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.22.0.*.*

IPL address

Get/Set:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A value of an empty string is used to indicate that the Image Activation Profile will use next IPL address set by HCD.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.28.0.*.*

IPL parameter

Get/Set:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A value of an empty string is used to indicate that the Image Activation Profile will use next IPL parameter set by HCD.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.29.0.*.*

Initial processing weight

Get/Set: The relative amount of shared processor resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition with at least one not dedicated central processor.

  1 - 999
  Represents the relative amount of shared processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated central processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.30.0.*.*

Initial processing weight capped

Get/Set: Whether or not the initial processing weight is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the initial processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of resources, regardless of the availability of excess processor resources.

  HWMCA_FALSE
  Indicates that the initial processing weight for the CPC Image is not capped. It represents the share of resources guaranteed to a logical partition when all processor resources are in use. Otherwise, when excess processor resources are available, the logical partition can use them if necessary.

  Note: The initial processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition with at least one not dedicated central processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.31.0.*.*

Minimum processing weight

Get/Set: The minimum relative amount of shared processor resources allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition with at least one not dedicated central processor.

  1 - 999
  Represents the minimum relative amount of shared processor resources allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and less than or equal to the Initial Processing Weight used to define the minimum relative amount of shared processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated central processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.38.0.*.*

Maximum processing weight

Get/Set: The maximum relative amount of shared processor resources allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition with at least one not dedicated central processor.

  1 - 999
  Represents the maximum relative amount of shared processor resources allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 and greater than or equal to the Initial Processing Weight used to define the maximum relative amount of shared processor resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated central processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.39.0.*.*

WorkLoad manager enabled

Get/Set: Whether or not WorkLoad Manager is allowed to change processing weight related attributes.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that WorkLoad Manager is allowed to change processing weight related attributes for this CPC Image object.

  HWMCA_FALSE
  Indicates that WorkLoad Manager is not allowed to change processing weight related attributes for this CPC Image object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.40.0.*.*

Defined capacity

Get/Set: The defined capacity expressed in terms of Millions of Service Units (MSU)s per hour. MSUs is a measure of processor resource consumption. The amount of MSUs a logical partition consumes is dependent on the model, the number of logical processors available to the partition, and the amount of time the logical partition is dispatched. The defined capacity value specifies how much capacity the logical partition is to be managed to by WorkLoad Manager for the purpose of software pricing.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  No defined capacity is specified for this logical partition.

  1 - nnnn
  Represents the amount of defined capacity specified for this logical partition.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.43.0.*.*

IPL type

Get/Set: The IPL type value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_IPLTYPE_STANDARD
  Indicates that the image activation profile is used to perform a standard load.

  HWMCA_IPLTYPE_SCSI
  Indicates that the image activation profile is used to perform a SCSI load.

  HWMCA_IPLTYPE_SCSIDUMP
  Indicates that the image activation profile is used to perform a SCSI dump.

  HWMCA_IPLTYPE_NVME
  Indicates that the image activation profile is used to perform an NVMe load.

  HWMCA_IPLTYPE_NVMEDUMP
  Indicates that the image activation profile is used to perform an NVMe dump.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.52.0.*.*

Worldwide port name

Get/Set: The worldwide port name value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.53.0.*.*

Boot program selector

Get/Set: The boot program selector value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.54.0.*.*

Logical unit number

Get/Set: The logical unit number value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.55.0.*.*

Boot record logical block address

Get/Set: The boot record logical block address value for the activation profile.

- Data type for get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.56.0.*.*

Operating system specific load parameters

Get/Set: The operating system specific load parameters for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.57.0.*.*

Initial Application Assist Processor processing weight

Get/Set: The relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1-999
  Represents the relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.60.0.*.*

Initial Application Assist Processor processing weight capped

Get/Set: Whether or not the initial processing weight for Application Assist Processor (AAP) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the initial Application Assist Processor (AAP) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Application Assist Processor (AAP) processor resources, regardless of the availability of excess Application Assist Processor (AAP) processor resources.

  HWMCA_FALSE
  Indicates that the initial Application Assist Processor (AAP) processor processing weight for the CPC Image is not capped. It represents the share of Application Assist Processor (AAP) processor resources guaranteed to a logical partition when all Application Assist Processor (AAP) processor resources are in use. Otherwise, when excess Application Assist Processor (AAP) processor resources are available, the logical partition can use them if necessary.

  Note: The initial Application Assist Processor (AAP) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image

does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.61.0.*.*

Minimum Application Assist Processor processing weight

Get/Set: The minimum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1-999
  Represents the minimum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.62.0.*.*

Maximum Application Assist Processor processing weight

Get/Set: The maximum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

  1-999
  Represents the maximum relative amount of shared Application Assist Processor (AAP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Application Assist Processor (AAP) processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no maximum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.63.0.*.*

Initial Integrated Facility for Linux processing weight

Get/Set: The relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  1-999
  Represents the relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.66.0.*.*

Initial Integrated Facility for Linux Processing weight capped

Get/Set: Whether or not the initial processing weight for Integrated Facility for Linux (IFL) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the initial Integrated Facility for Linux (IFL) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Integrated Facility for Linux (IFL) processor resources, regardless of the availability of excess Integrated Facility for Linux (IFL) processor resources.

  HWMCA_FALSE
  Indicates that the initial Integrated Facility for Linux (IFL) processor processing weight for the CPC Image is not capped. It represents the share of Integrated Facility for Linux (IFL) processor resources guaranteed to a logical partition when all Integrated Facility for Linux (IFL) processor resources are in use. Otherwise, when excess Integrated Facility for Linux (IFL) processor resources are available, the logical partition can use them if necessary.

  Note: The initial Integrated Facility for Linux (IFL) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.67.0.*.*

Minimum Integrated Facility for Linux processing weight

Get/Set: The minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

  1-999
  Represents the minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.68.0.*.*

Maximum Integrated Facility for Linux processing weight

Get/Set: The maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

0
CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.

1-999
Represents the maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Integrated Facility for Linux (IFL) processor resources allocated to the CPC Image object. The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Application Assist Processor (AAP) processor. A value of zero can also be specified to indicate that there is no maximum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Integrated Facility for Linux (IFL) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.69.0.*.*

Initial Internal Coupling Facility processing weight

Get/Set: The relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.72.0.*.*

Initial Internal Coupling Facility processing weight capped

Get/Set: Whether or not the initial processing weight for Internal Coupling Facility (ICF) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the initial Internal Coupling Facility (ICF) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Internal Coupling Facility (ICF) processor resources, regardless of the availability of excess Internal Coupling Facility (ICF) processor resources.

  HWMCA_FALSE
  Indicates that the initial Internal Coupling Facility (ICF) processor processing weight for the CPC Image is not capped. It represents the share of Internal Coupling Facility (ICF) processor resources guaranteed to a logical partition when all Internal Coupling Facility (ICF) processor resources are in use. Otherwise, when excess Internal Coupling Facility (ICF) processor resources are available, the logical partition can use them if necessary.

  Note: The initial Internal Coupling Facility (ICF) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not

represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.73.0.*.*

Minimum Internal Coupling Facility processing weight

Get/Set: The minimum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the minimum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.74.0.*.*

Maximum Internal Coupling Facility processing weight

Get/Set: The maximum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

  1-999
  Represents the maximum relative amount of shared Internal Coupling Facility (ICF) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared Internal Coupling Facility (ICF) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Internal Coupling Facility (ICF) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.75.0.*.*

Initial z Integrated Information Processors processing weight

Get/Set: The relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  1-999
  Represents the relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

A value 1 - 999 used to define the relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.81.0.*.*

Initial z Integrated Information Processors processing weight capped

Get/Set: Whether or not the initial processing weight for z Integrated Information Processors (zIIP) processors is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Indicates that the initial z Integrated Information Processors (zIIP) processor processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of z Integrated Information Processors (zIIP) processor resources, regardless of the availability of excess z Integrated Information Processors (zIIP) processor resources.

  HWMCA_FALSE
  Indicates that the initial z Integrated Information Processors (zIIP) processor processing weight for the CPC Image is not capped. It represents the share of z Integrated Information Processors (zIIP) processor resources guaranteed to a logical partition when all z Integrated Information Processors (zIIP) processor resources are in use. Otherwise, when excess z Integrated Information Processors (zIIP) processor resources are available, the logical partition can use them if necessary.

  Note: The initial z Integrated Information Processors (zIIP) processor processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.82.0.*.*

Minimum z Integrated Information Processors processing weight

Get/Set: The minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

  1-999
  Represents the minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the minimum relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.83.0.*.*

Maximum z Integrated Information Processors processing weight

Get/Set: The maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.

1-999
Represents the maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  A value 1 - 999 used to define the maximum relative amount of shared z Integrated Information Processors (zIIP) processor resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no maximum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated z Integrated Information Processors (zIIP) processor.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.84.0.*.*

Group profile name

Get/Set: Defines the name of the group capacity profile that is to be used for the CPC Image object activated with this profile.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.93.0.*.*

Load at activation

Get/Set: Defines if the CPC Image object activated with this profile should be loaded (IPLed) at the end of the activation.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The CPC Image object will be loaded (IPLed) at the end of the activation.

  HWMCA_FALSE
  The CPC Image object will not be loaded (IPLed) at the end of the activation.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.94.0.*.*

Central storage

Get/Set: Defines the initial amount of central storage (in megabytes) to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.95.0.*.*

Reserved central storage

Get/Set: Defines the reserved amount of central storage (in megabytes) to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.96.0.*.*

Expanded storage

Get/Set: Defines the initial amount of expanded storage (in megabytes) to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.97.0.*.*

Note: Expanded storage is only supported on CPCs version 2.13.1 and earlier.

Reserved expanded storage

Get/Set: Defines the reserved amount of expanded storage (in megabytes) to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.98.0.*.*

Note: Expanded storage is only supported on CPCs version 2.13.1 and earlier.

Number of dedicated general purpose processors

Get/Set: Defines the number of dedicated general purpose processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.99.0.*.*

Number of reserved dedicated general purpose processors

Get/Set: Defines the number of reserved dedicated general purpose processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.100.0.*.*

Number of dedicated Application Assist Processor (AAP) processors

Get/Set: Defines the number of dedicated Application Assist Processor (AAP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.101.0.*.*

Number of reserved dedicated Application Assist Processor (AAP) Processors

Get/Set: Defines the number of reserved dedicated Application Assist Processor (AAP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.102.0.*.*

Number of dedicated Integrated Facility for Linux (IFL) processors

Get/Set: Defines the number of dedicated Integrated Facility for Linux (IFL) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.103.0.*.*

Number of reserved dedicated Integrated Facility for Linux (IFL) processors

Get/Set: Defines the number of reserved dedicated Integrated Facility for Linux (IFL) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.104.0.*.*

Number of dedicated Internal Coupling Facility (ICF) processors

Get/Set: Defines the number of dedicated Internal Coupling Facility (ICF) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.105.0.*.*

Number of reserved dedicated Internal Coupling Facility (ICF) processors

Get/Set: Defines the number of reserved dedicated Internal Coupling Facility (ICF) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.106.0.*.*

Number of dedicated z Integrated Information Processors (zIIP) processors

Get/Set: Defines the number of dedicated z Integrated Information Processors (zIIP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.107.0.*.*

Number of reserved dedicated z Integrated Information Processors (zIIP) processors

Get/Set: Defines the number of reserved dedicated z Integrated Information Processors (zIIP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.108.0.*.*

Number of shared general purpose processors

Get/Set: Defines the number of shared general purpose processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.109.0.*.*

Number of reserved shared general purpose processors

Get/Set: Defines the number of reserved shared general purpose processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.110.0.*.*

Number of shared Application Assist Processor (AAP) processors

Get/Set: Defines the number of shared Application Assist Processor (AAP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.111.0.*.*

Number of reserved shared Application Assist Processor (AAP) processors

Get/Set: Defines the number of reserved shared Application Assist Processor (AAP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.112.0.*.*

Number of shared Integrated Facility for Linux (IFL) processors

Get/Set: Defines the number of shared Integrated Facility for Linux (IFL) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.113.0.*.*

Number of reserved shared Integrated Facility for Linux (IFL) processors

Get/Set: Defines the number of reserved shared Integrated Facility for Linux (IFL) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.114.0.*.*

Number of shared Internal Coupling Facility (ICF) processors

Get/Set: Defines the number of shared Internal Coupling Facility (ICF) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.115.0.*.*

Number of reserved shared Internal Coupling Facility (ICF) processors

Get/Set: Defines the number of reserved shared Internal Coupling Facility (ICF) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.116.0.*.*

Number of shared z Integrated Information Processors (zIIP) processors

Get/Set: Defines the number of shared z Integrated Information Processors (zIIP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.117.0.*.*

Number of reserved shared z Integrated Information Processors (zIIP) processors

Get/Set: Defines the number of reserved shared z Integrated Information Processors (zIIP) processors to be used for the CPC Image object activated with this profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.118.0.*.*

Basic CPU counter authorization control

Get/Set: Enables/disables the use of the basic CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.168.0.*.*

Problem state CPU counter authorization control

Get/Set: Enables/disables the use of the problem state CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.169.0.*.*

Crypto activity CPU counter authorization control

Get/Set: Enables/disables the use of the crypto activity CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.170.0.*.*

Extended CPU counter authorization control

Get/Set: Enables/disables the use of the extended CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.171.0.*.*

Coprocessor group CPU counter authorization control

Get/Set: Enables/disables the use of the coprocessor group CPU counter facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.172.0.*.*

Basic CPU sampling authorization control

Get/Set: Enables/disables the use of the basic CPU sampling facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.173.0.*.*

Diagnosis CPU sampling authorization control

Get/Set: Enables/disables the use of the diagnosis CPU sampling facility for the CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The authorization control is enabled.

  HWMCA_FALSE
  The authorization control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.174.0.*.*

Permit AES key import functions

Get/Set: Enables/disables the importing of AES keys for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The importing of AES keys is enabled.

  HWMCA_FALSE
  The importing of AES keys is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.183.0.*.*

Permit DEA key import functions

Get/Set: Enables/disables the importing of DEA keys for the associated CPC Image.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The importing of DEA keys is enabled.

  HWMCA_FALSE
  The importing of DEA keys is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.184.0.*.*

Description

Get/Set: The description of the profile with a maximum length of 51 (including the null terminator).

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.203.0.*.*

Partition Identifier

Get/Set: The partition identifier for the activation profile.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- Data type for Set: HWMCA_TYPE_INTEGER between 0 and 63, inclusive.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.51.0.*.*

Operating mode

Get/Set: The operating mode value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
  - HWMCA_GENERAL_OPERATING_MODE (1)
  - HWMCA_ESA390_OPERATING_MODE (1)
  - HWMCA_ESA390TPF_OPERATING_MODE (2)*

    * This operating mode is only valid for Images running on systems prior to console version 2.14.0.
  - HWMCA_CF_OPERATING_MODE (3)
  - HWMCA_LINUX_OPERATING_MODE (4)
  - HWMCA_FMEX_OPERATING_MODE (5)
  - HWMCA_HMEX_OPERATING_MODE (6)
  - HWMCA_HMAS_OPERATING_MODE (7)
  - HWMCA_ZVM_OPERATING_MODE (8)
  - HWMCA_ZAWARE_OPERATING_MODE (9)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.204.0.*.*

Clock type

Get/Set: The clock type assignment for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
  - HWMCA_CLOCK_TYPE_STANDARD (0)
  - HWMCA_CLOCK_TYPE_LPAR (1)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.205.0.*.*

Time offset days

Get/Set: The time offset days for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER (0 - 999)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.206.0.*.*

Time offset hours

Get/Set: The time offset hours for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER (0 - 23)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.207.0.*.*

Time offset minutes

Get/Set: The time offset minutes for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER (0, 15, 30, or 45)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.208.0.*.*

Time offset increase or decrease

Get/Set: The time offset increase/decrease setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The local time zone is east of GMT.

  HWMCA_FALSE
  The local time zone is west of GMT.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.209.0.*.*

LICCC validation

Get/Set: Enables/disables whether or not the activation profile must conform to the current LICCC configuration.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The activation profile must conform to the current LICCC configuration.

  HWMCA_FALSE
  The activation profile is not required to conform to the current LICCC configuration.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.210.0.*.*

Global performance data control

Get/Set: Enables/disables the global performance data control setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The global performance data control is enabled.

  HWMCA_FALSE
  The global performance data control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.211.0.*.*

Input/Output configuration control

Get/Set: Enables/disables the I/O configuration control setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The I/O configuration control is enabled.

  HWMCA_FALSE
  The I/O configuration control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.212.0.*.*

Cross partition authority control

Get: The cross partition authority control setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The cross partition authority control is enabled.

  HWMCA_FALSE
  The cross partition authority control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.213.0.*.*

Logical partition isolation control

Get/Set: Enables/disables the logical partition isolation control setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The logical partition isolation control is enabled.

  HWMCA_FALSE
  The logical partition isolation control is disabled.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.214.0.*.*

Absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.217.0.*

Absolute capping value

Get/Set: The value used for absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  nnn.nn
  String form of decimal number representing the absolute capping value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.218.0.*

Application Assist Processor absolute capping type

Get/Set: The type of absolute capping to perform for Application Assist Processor (AAP) processors.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Application Assist Processor (AAP) processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.219.0.*

Application Assist Processor absolute capping value

Get/Set: The value used for Application Assist Processor (AAP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  nnn.nn
  String form of decimal number representing the absolute capping value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.220.0.*

Integrated Facility for Linux absolute capping type

Get/Set: The type of absolute capping to perform for Integrated Facility for Linux (IFL) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Integrated Facility for Linux (IFL) processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.221.0.*

Integrated Facility for Linux absolute capping value

Get/Set: The value used for Integrated Facility for Linux (IFL) absolute capping (if enabled).

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  nnn.nn
  String form of decimal number representing the absolute capping value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.222.0.*

Internal Coupling Facility absolute capping type

Get/Set: The type of absolute capping to perform for Internal Coupling Facility (ICF) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Internal Coupling Facility (ICF) processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.223.0.*

Internal Coupling Facility absolute capping value

Get/Set: The value used for Internal Coupling Facility (ICF) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  Absolute capping not enabled.

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.224.0.*

z Integrated Information Processor absolute capping type

Get/Set: The type of absolute capping to perform for z Integrated Information Processor (zIIP) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of z Integrated Information Processor (zIIP) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.225.0.*

z Integrated Information Processor absolute capping value

Get/Set: The value used for z Integrated Information Processor (zIIP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.226.0.*

Initial Virtual Flash Memory

Get/Set: The amount of Virtual Flash Memory in, Gigabytes (GB), to be allocated for a CPC image object at activation.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.265.0.*

Maximum Virtual Flash Memory

Get/Set: The maximum amount of Virtual Flash Memory, in Gigabytes (GB), that can be allocated for a CPC image object.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.266.0.*

Secure Service Container Host Name

Get/Set: The host name, with a maximum length of 64, to be used for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.269.0.*

Secure Service Container Master User ID

Get/Set: The master user ID, with a maximum length of 32, to be used for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.270.0.*

Secure Service Container Master User Password

Get/Set: The master user password, with a maximum length of 512, to be used for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.271.0.*

Secure Service Container Network Count

Get: The count of network adapters defined for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0 – 100)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.272.0.*

Secure Service Container Network Channel Path Identifier

Get/Set: The channel path identifier (CHPID) used for a network adapter for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.273.0-99.*

Note: There are really an array of these values, one for each network adapter. Each array element is uniquely identified a value from 0 to 99 in the SNMP object identifier for the attribute.

Secure Service Container Network Address Type

Get/Set: The address type for a network adapter for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.274.0-99.*
  - HWMCA_ADDRESS_TYPE_DHCP (1)
  - HWMCA_ADDRESS_TYPE_STATIC (2)

    Not valid for set operations, instead the static IP attribute should be set to change the network address type and static IP information at the same time.
  - HWMCA_ADDRESS_TYPE_LINKLOCAL (3)

Note: There are really an array of these values, one for each network adapter. Each array element is uniquely identified a value from 0 to 99 (or in reality the value of the Secure Service Container network count attribute minus 1) in the SNMP object identifier for the attribute.

Secure Service Container Network VLAN Identifier

Get/Set: The VLAN identifier for a network adapter for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER (-1 – 4094)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.275.0-99.*

  A value of -1 means that no VLAN identifier has been set for the network adapter.

Note: There are really an array of these values, one for each network adapter. Each array element is uniquely identified a value from 0 to 99 (or in reality the value of the Secure Service Container network count attribute minus 1) in the SNMP object identifier for the attribute.

Secure Service Container Network Static IP Information

Get/Set: The static IP address for a network adapter for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.276.0-99.*

The network mask value must be appended to the IP address on the Set operation in the format *ip-address/network-mask*. Valid ranges are as follows:

- Valid range for network mask value of an IPv4 address is 0-32.
- Valid range for network mask value of an IPv6 address is 0-128.

Note: There are really an array of these values, one for each network adapter. Each array element is uniquely identified a value from 0 to 99 in the SNMP object identifier for the attribute.

Secure Service Container Network Port Number

Get/Set: The port number for a network adapter for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER (-1 – 1)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.277.0-99.*

Note: There are really an array of these values, one for each network adapter. Each array element is uniquely identified a value from 0 to 99 (or in reality the value of the Secure Service Container network count attribute minus 1) in the SNMP object identifier for the attribute.

Secure Service Container IPv4 Gateway Count

Get: The count of IPv4 addresses to be used as a gateway for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-1)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.278.0.*

Secure Service Container IPv4 Gateway

Get/Set: The IPv4 address to be used as a gateway for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.279.0.*

Secure Service Container IPv6 Gateway Count

Get: The count of IPv6 addresses to be used as a gateway for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-1)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.280.0.*

Secure Service Container IPv6 Gateway

Get/Set: The IPv6 address to be used as a gateway for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.281.0.*

Secure Service Container DNS Count

Get: The count of domain name server (DNS) addresses to be used for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-1)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.282.0.*

Secure Service Container DNS Information

Get/Set: The domain name server (DNS) address to be used for the CPC image object when activated as a Secure Service Container.

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.283.0-1.*

Note: There are really an array of these values, one for DNS address. Each array element is uniquely identified a value from 0 to 1 in the SNMP object identifier for the attribute.

Cryptographic Control Domain Count

Get: The count of cryptographic domains for a CPC Image object when activated.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-84)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.288.0.*

Cryptographic Control Domains

Get/Set: The cryptographic domains for the CPC Image object when activated.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.284.0-84.*

Note: There are really an array of these values, one for each crypto. Each array element is uniquely identified a value from 0 to 84 (or in reality the value of the Cryptographic Control Domain count attribute minus 1) in the SNMP object identifier for the attribute.

Cryptographic Usage Domain Count

Get: The count of cryptographic usage domains for a CPC image object when activated.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-84)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.289.0.*

Cryptographic Usage Domains

Get/Set: The cryptographic usage domains for the CPC image object when activated.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.285.0-84.*

Note: There are really an array of these values, one for each crypto. Each array element is uniquely identified a value from 0 to 84 (or in reality the value of the Cryptographic Usage Domain count attribute minus 1) in the SNMP object identifier for the attribute.

Cryptographic Candidate List Count

Get: The count of cryptos to be assigned to the CPC image object when activated.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-15)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.290.0.*

Cryptographic Candidate List

Get/Set: The cryptos to be assigned to the CPC image object when activated.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.286.0-15.*

Note: There are really an array of these values, one for each crypto. Each array element is uniquely identified a value from 0 to 15 (or in reality the value of the Cryptographic Candidate List count attribute minus 1) in the SNMP object identifier for the attribute.

Cryptographic Online List Count

Get: The count of cryptos to be brought online for the CPC image object when activated.

- Data type returned on Get: HWMCA_TYPE_INTEGER (0-15)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.291.0.*

Cryptographic Online List

Get/Set: The cryptos to be brought online for the CPC image object when activated.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.287.0-15.*

Note: There are really an array of these values, one for each crypto. Each array element is uniquely identified a value from 0 to 15 (or in reality the value of the Cryptographic Online List count attribute minus 1) in the SNMP object identifier for the attribute.

Secure Service Container Boot Selection

Get/Set: Indicates whether to run the Secure Service Container appliance installer or the Secure Service Container appliance itself.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
  - HWMCA_BC_INSTALLER (1)
  - HWMCA_BC_APPLIANCE (2)
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.292.0.*

Initial Container Based Processor processing weight

Get/Set: The relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Get: HWMCA_TYPE_INTEGER

  0
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

  1-999
  Represents the relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.
- Data type for Set: HWMCA_TYPE_INTEGER

  1-999
  A value 1 - 999 used to define the relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Container Based Processor (CBP).
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.295.0.*

Initial Container Based Processor processing weight capped

Get/Set: Whether or not the initial processing weight for Container Based Processors (CBP) is a limit or a target.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE (1)
  Indicates that the initial Container Based Processor (CBP) processing weight for the CPC Image object is capped. It represents the logical partition's maximum share of Container Based Processor (CBP) resources, regardless of the availability of excess Container Based Processor (CBP) resources.

HWMCA_FALSE (0)
Indicates that the initial Container Based Processor (CBP) processing weight for the CPC Image is not capped. It represents the share of Container Based Processor (CBP) resources guaranteed to a logical partition when all Container Based Processor (CBP) resources are in use. Otherwise, when excess Cloud Based Processor (CBP) resources are available, the logical partition can use them if necessary.

Note: The initial Container Based Processor (CBP) processing weight capped attribute cannot be set and the value returned for a Get request is always HWMCA_FALSE when the CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.296.0.*

Minimum Container Based Processor processing weight

Get/Set: The minimum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  1
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

  1-999
  Represents the minimum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

- Data type for Set: HWMCA_TYPE_INTEGER

  1-999
  A value 1 - 999 used to define the minimum relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object. A value of zero can also be specified to indicate that there is no minimum value for the processing weight.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Cloud Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.297.0.*

Maximum Container Based Processor processing weight

Get/Set: The maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object. The maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  1
  CPC Image does not represent a logical partition or the CPC Image does not represent a logical partition with at least one not dedicated Container Based Processor (CBP).

  1-999
  Represents the maximum relative amount of shared Container Based Processor (CBP) resources initially allocated to the CPC Image object.

- Data type for Set: HWMCA_TYPE_INTEGER

  1-999
  A value 1 - 999 used to define the maximum relative amount of shared Container Based Processor (CBP) resources allocated to the CPC Image object.

  Note: The setting of this attribute is only valid for CPC Image objects that represent a logical partition with at least one not dedicated Container Based Processor (CBP).

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.298.0.*

Container Based Processor absolute capping type

Get/Set: The type of absolute capping to perform for Container Based Processors (CBP).

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None.

  1
  Absolute capping in number of CBP processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.301.0.*

Container Based Processor absolute capping value

Get/Set: The value used for Container Based Processor (CBP) absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  nnn.nn
  String form of decimal number representing the absolute capping value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.302.0.*

Container Based Processor Group Profile absolute capping type

Get/Set: The type of Group Profile absolute capping to perform for Container Based Processor (CBP) processors.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of Container Based Processor (CBP) processors.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.303.0.*

Container Based Processor group absolute capping value

Get/Set: The value used for Container Based Processor (CBP) group absolute capping (if enabled)

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1-nnnn
  Represents the processor capping value in hundredths. For example, a capping value of 100 equates to 1.00.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.304.0.*

Number of dedicated Container Based Processors

Get/Set: Defines the number of dedicated Container Based Processors (CBP) to be used for the CPC Image object activated with this profile.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.305.0.*.*

Number of reserved dedicated Container Based Processors

Get/Set: Defines the number of reserved dedicated Container Based Processors (CBP) to be used for the CPC Image object activated with this profile.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.306.0.*.*

Number of shared Container Based Processors

Get/Set: Defines the number of shared Container Based Processors (CBP) to be used for the CPC Image object activated with this profile.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.307.0.*.*

Number of reserved shared Container Based Processors

Get/Set: Defines the number of reserved shared Container Based (CBP) processors to be used for the CPC Image object activated with this profile.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.308.0.*.*

Verify software signature

Get/Set: The verify software signature flag for the Image profiles.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  Verify software signature is off.

  1
  Verify software signature is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.6.0.315.0.*.*

Image activation profile notifications

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the HwmcaGet to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Load activation profile object

Load activation profile name bindings

Load activation profile object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.x.x.y.z

Where ***x.x.*** equals the attribute identifier for the object, ***y*** equals a unique number for the specific instance of the CPC Object, and ***z*** equals a unique number for the specific instance of the Load Activation Profile.

Load activation profile attributes

Contents

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from

the sequence and returned as individual variable bindings. The sequence contains all of the attributes for a Load profile object.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.*y.z* (where *y* equals a unique number for the specific instance of the CPC Object, and *z* equals a unique number for the specific instance of the Load profile)

Name

Get:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.1.0.*.*

Object type

Get: This returns the type of object the object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER

  HWMCA_ACT_PROFILE_LOAD
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.22.0.*.*

IPL address

Get/Set:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A value of an empty string is used to indicate that the Load Activation Profile will use next IPL address set by HCD.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.28.0.*.*

IPL parameter

Get/Set:

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

  Note: A value of an empty string is used to indicate that the Load Activation Profile will use next IPL parameter set by HCD.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.29.0.*.*

IPL type

Get/Set: The IPL type value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_IPLTYPE_STANDARD
  Indicates that the load activation profile is used to perform a standard load.

  HWMCA_IPLTYPE_SCSI
  Indicates that the load activation profile is used to perform a SCSI load.

  HWMCA_IPLTYPE_SCSIDUMP
  Indicates that the load activation profile is used to perform a SCSI dump.

  HWMCA_IPLTYPE_NVME
  Indicates that the load activation profile is used to perform an NVMe load.

  HWMCA_IPLTYPE_NVMEDUMP
  Indicates that the load activation profile is used to perform an NVMe dump.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.52.0.*.*

Worldwide port name

Get/Set: The worldwide port name value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.53.0.*.*

Boot program selector

Get/Set: The boot program selector value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.54.0.*.*

Logical unit number

Get/Set: The logical unit number value for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.55.0.*.*

Boot record logical block address

Get/Set: The boot record logical block address value for the activation profile.

- Data type for get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.56.0.*.*

Operating system specific load parameters

Get/Set: The operating system specific load parameters for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.57.0.*.*

Store Status

Get/Set: The store status setting for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  The store status is performed before the load starts.

  HWMCA_FALSE
  The store status is not performed before the load starts.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.166.0.*.*.*.*

Load Type

Get/Set: The load type for the activation profile.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

  HWMCA_TRUE
  Clears main storage during the load.

  HWMCA_FALSE
  Performs the load without clearing main storage.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.167.0.*.*.*.*

Description

Get/Set: The description of the profile with a maximum length of 51 (including the null terminator).

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.203.0.*.*

Verify software signature

Get/Set: The verify software signature flag for the Load profiles.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  Verify software signature is off.

  1
  Verify software signature is on.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.7.0.315.0.*.*

Load activation profile notifications

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the HwmcaGet to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

Group profile object

Group profile name bindings

Group profile object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.x.x.y.z
Where x.x. equals the attribute identifier for the object, y equals a unique number for the specific instance of the CPC Object, and z equals a unique number for the specific instance of the Group Profile.

Group profile attributes

Contents

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings. The sequence contains all of the attributes for a Group profile object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.*y.z* (where *y* equals a unique number for the specific instance of the CPC Object, and *z* equals a unique number for the specific instance of the Group profile)

Name

Get: This returns the name of object the group profile object identifier represents.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.1.0.*.*.*.*

Object type

Get: This returns the type of object the group profile object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER HWMCA_ACT_PROFILE_GROUP
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.22.0.*.*.*.*

Capacity

Get/Set: This returns the capacity value of object the group profile object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.92.0.*.*.*.*

Description

Get/Set: The description of the profile with a maximum length of 51 (including the null terminator).

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- Data type for Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.203.0.*.*

Group absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.227.0.*

Group absolute capping value

Get/Set: The value used for absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

  nnn.nn
  String form of decimal number representing the absolute capping value.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.228.0.*

Internal Coupling Facility group absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

  0
  None

  1
  Absolute capping in number of processors
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.229.0.*

Internal Coupling Facility group absolute capping value

Get/Set: The value used for Internal Coupling Facility (ICF) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

  0
  None

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.230.0.*

Integrated Facility for Linux group absolute capping type

Get/Set: The type of absolute capping to perform for Integrated Facility for Linux (IFL) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of Integrated Facility for Linux (IFL) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.231.0.*

Integrated Facility for Linux group absolute capping value

Get/Set: The value used for Integrated Facility for Linux (IFL) absolute capping (if enabled).

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
None

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.232.0.*

z Integrated Information Processor group absolute capping type

Get/Set: The type of absolute capping to perform for z Integrated Information Processor (zIIP) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of z Integrated Information Processor (zIIP) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.233.0.*

z Integrated Information Processor group absolute capping value

Get/Set: The value used for z Integrated Information Processor (zIIP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.234.0.*

Members

Get/Set: A blank separator list of Image activation profile object names that are members of this group.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.8.0.309.0.*

Group profile notifications

Object created (HWMCA_EVENT_CREATED)

This event has no additional data. The object identifier can be used with the HwmcaGet to get any data required for this newly created object.

Object destruction (HWMCA_EVENT_DESTROYED)

This event has no additional data.

LPAR capacity group object

LPAR capacity group name bindings

LPAR capacity group object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.x.x.y.z
Where x.x. equals the attribute identifier for the object, y equals a unique number for the specific instance of the CPC Object, and z equals a unique number for the specific instance of the LPAR capacity group.

LPAR capacity group attributes

Contents

Get:

- Data type returned on Get: HWMCA_TYPE_OPAQUE_SEQUENCE

  The data for this attribute is a sequence of object identifier/data value pairs. When this object identifier is used with the *HwmcaEnhancedGet* interface the individual pairs in the sequence are extracted from the sequence and returned as individual variable bindings. The sequence contains all of the attributes for a LPAR capacity group object.
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.*y.z* (where *y* equals a unique number for the specific instance of the CPC Object, and *z* equals a unique number for the specific instance of the LPAR capacity group)

Name

Get: This returns the name of object the LPAR capacity group object identifier represents.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.1.0.*.*.*.*

Object type

Get: This returns the type of object the LPAR capacity group object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER HWMCA_LPAR_GROUP
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.22.0.*.*.*.*

Capacity

Get/Set: This returns the capacity value of object the LPAR capacity group object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.192.0.*.*.*.*

Group absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of processors

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.227.0.*

Group absolute capping value

Get/Set: The value used for absolute capping (if enabled).

- Data type returned on Get/Set: HWMCA_TYPE_OCTETSTRING

0
None

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.228.0.*

Internal Coupling Facility group absolute capping type

Get/Set: The type of absolute capping to perform.

- Data type returned on Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of processors

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.229.0.*

Internal Coupling Facility group absolute capping value

Get/Set: The value used for Internal Coupling Facility (ICF) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
None

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.230.0.*

Integrated Facility for Linux group absolute capping type

Get/Set: The type of absolute capping to perform for Integrated Facility for Linux (IFL) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of Integrated Facility for Linux (IFL) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.231.0.*

Integrated Facility for Linux group absolute capping value

Get/Set: The value used for Integrated Facility for Linux (IFL) absolute capping (if enabled).

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
None

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.232.0.*

z Integrated Information Processor group absolute capping type

Get/Set: The type of absolute capping to perform for z Integrated Information Processor (zIIP) processors.

- Data type for Get/Set: HWMCA_TYPE_INTEGER

0
None

1
Absolute capping in number of z Integrated Information Processor (zIIP) processors.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.233.0.*

z Integrated Information Processor group absolute capping value

Get/Set: The value used for z Integrated Information Processor (zIIP) absolute capping.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING

0
Absolute capping not enabled.

nnn.nn
String form of decimal number representing the absolute capping value.

- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.234.0.*

Members

Get/Set: A blank separator list of Image activation profile object names that are members of this group.

- Data type for Get/Set: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.13.0.309.0.*

Capacity record object

Capacity record name bindings

Capacity record object identifier

SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.x.x.y.z
Where x.x. equals the attribute identifier for the object, y equals a unique number for the specific instance of the Defined CPC Object, and z equals a unique number for the specific instance of the Capacity Record. Additionally, the capacity record itself can be queried using an object identifier of the form 1.3.6.1.4.1.2.6.42.9.0.y.z. When the capacity record itself is queried, it returns a data type of HWMCA_TYPE_OCTETSTRING with the data being an XML string describing all aspects of the record. Refer to Appendix F, “XML descriptions,” on page 209 for details on the format of the XML that is returned.

Capacity record attributes

Object type

Get: This returns the type of object the capacity record object identifier represents.

- Data type returned on Get: HWMCA_TYPE_INTEGER HWMCA_CAPACITY_RECORD
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.22.0.*.*

Record identifier

Get: This returns the identifier for the capacity record.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.135.0.*.*

Record type

Get: This returns a value that indicates the type of capacity record.

- Data type returned on Get: HWMCA_TYPE_INTEGER HWMCA_CAPACITY_RECORD_TYPE_CBU HWMCA_CAPACITY_RECORD_TYPE_OOCOD HWMCA_CAPACITY_RECORD_TYPE_PLANNED_EVENT HWMCA_CAPACITY_RECORD_TYPE_LOANER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.136.0.*.*

Activation status

Get: This returns an indication if any of the resources defined for the record are currently activated.

- Data type returned on Get: HWMCA_TYPE_INTEGER HWMCA_CAPACITY_RECORD_STATUS_NOT_ACTIVATED HWMCA_CAPACITY_RECORD_STATUS_REAL HWMCA_CAPACITY_RECORD_STATUS_TEST HWMCA_CAPACITY_RECORD_STATUS_CAN_BE_ACTIVATED
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.137.0.*.*

Activation date

Get: Defines the time stamp for when additional capacity for the record was activated.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.138.0.*.*

Record expiration date

Get: Defines the time stamp for when the capacity record will expire.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.139.0.*.*

Activation expiration date

Get: Defines the time stamp for when the additional capacity activated for the record will expire and no longer be active.

- Data type returned on Get: HWMCA_TYPE_OCTETSTRING
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.140.0.*.*

Maximum real days

Get: Defines the maximum days that real additional capacity can be activated for the record. A value of -1 indicates that the number of days is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.141.0.*.*

Maximum test days

Get: Defines the maximum days that test additional capacity can be activated for the record. A value of -1 indicates that the number of days is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.142.0.*.*

Remaining real days

Get: Defines the remaining number of days that additional real capacity can be active for the record. A value of -1 indicates that the number of days is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.143.0.*.*

Remaining test days

Get: Defines the remaining number of days that additional test capacity can be active for the record. A value of -1 indicates that the number of days is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.144.0.*.*

Remaining number of real activations

Get: Defines the number of times that real additional capacity can be activated for the record. A value of -1 indicates that activation count is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.147.0.*.*

Remaining number of test activations

Get: Defines the number of times that test additional capacity can be activated for the record. A value of -1 indicates that activation count is unlimited.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.148.0.*.*

Record real additional capacity maximum hours

Get: Record real additional capacity maximum hours.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.317.0.*.*

Record real additional capacity remaining hours

Get: Record real additional capacity remaining hours.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.318.0.*.*

Record max test additional capacity hours

Get: Record max test additional capacity hours.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.319.0.*.*

Record remaining test activation hours

Get: Record remaining test activation hours.

- Data type returned on Get: HWMCA_TYPE_INTEGER
- SNMP MIB Name: 1.3.6.1.4.1.2.6.42.9.0.320.0.*.*

Chapter 5. Configuring for the data exchange APIs

Before the Console APIs (Data Exchange APIs and Commands API) can be used, some configuration tasks must be performed on the Hardware Management Console or Support Element Console. These configuration tasks fall into two categories:

- SNMP configuration
- Console API configuration.

Refer to the information on the following pages for detailed steps necessary to perform these two types of configuration.

Note: Once these steps have been successfully completed, the Console APIs (Data Exchange APIs and Commands API) can be used while the Hardware Management Console or Support Element Console is up and running. The APIs will not be functional when the console is not running, even if these configuration steps have been completed.

Configuring the console for API

The Console API configuration steps can be performed by using the *Customize API Settings* task.

To configure the Console for API support:

1. Log on to the Console in *Access Administrator* mode.
2. Select **Tasks Index** in the left navigation pane.
3. Select the **Customize API Settings** task from the tasks list.
4. Select **Enable**.
5. Specify any **SNMP agent parameters** desired. **Note:** No special SNMP agent parameters are required for API to work correctly.
6. Add one or more entries in the **Community Names** box by selecting **Add** push button to add a new community name or select the **Change** push button to change an existing community name. It is recommended that one entry be added for each TCP/IP host (machine) that will be making Management API requests.

   Name
   This field should be filled in with any character string. Each community name in the list must be unique. Note the community name(s) that are to be used by applications using the Console APIs, since it will need to be specified on the *HwmcaInitialize* calls issued by those applications.

   Address
   If you are following the recommendation of adding an entry for the Console and each requesting application, then this field should be filled in with the TCP/IP address of the machine that will be using the community name.

   Note: This can be specified as an IPV6 address.

   Network Mask/Prefix
   If you are following the recommendation of adding an entry for the Console itself and each requesting application, then this field should be filled in with a character string of **255.255.255.255**. When using IPV6 addresses, the prefix for the address should be used, instead of a masked value.

   Access Type
   Use this field to specify the type of access that is allowed for the community name. The access type for the community name(s) that are to be used by applications using the Console APIs *MUST* be **read/write**.

7. Specify any additional locations where enterprise-specific SNMP trap messages created by the Console should be sent in the **Event notification information** box. Entries can be added, changed, and deleted throughout with the use of the **Add**, **Change**, and **Delete** push buttons respectively. Adding entries to the **Event notification information** box will cause the Console to send the specified event notifications to TCP/IP port 162 at the locations specified. 8. Select the **OK** or **Apply** push buttons to save the changes.
8. If any of the above data was added or changed, then you need to stop and restart the Console before the changes will be in effect. The Console can be stopped by using the **Shutdown** or **Restart** task found in the **Console Actions** view.

Configuring BCPii Security Controls (for consoles 2.14.0 or later)

For consoles version 2.14.0 or later, it is possible to configure additional permissions for API requests sent via Base Control Program internal interface (BCPii). The following additional capabilities can now be configured.

- Enable the partition to send commands

  Allows for control over which CPC Images (i.e. operating systems) are allowed to send BCPii requests. Prior to Consoles version 2.14.0 the *Cross Partition Authority* flag for the CPC Image was used to control this capability. This setting has been separated from the *Cross Partition Authority* flag and is used solely for BCPii requests. The value of this setting can be changed using the *Change LPAR Security* task to affect the current value for an active CPC image or via the image activation profile to affect the value to be used the next time the CPC Image is activated.
- Enable the partition to receive commands from other partitions

  Allows for control over whether requests targeting a specific object are allowed to be received and handled. These controls are available for Defined CPC and CPC Image objects. The *System Details* task can be used to configure these controls for a Defined CPC and the *Change LPAR Security* task can be used to configure these controls for an active CPC Image. Additionally, the image activation profile now allows for these controls to be configured to define the permissions that should be used when a CPC Image is activated.

  In all cases, it is possible to disallow requests completely for a target object. Conversely, it is also possible to allow requests from any source to be allowed for a target object. It is also possible to only allow requests to from a specific set of sources.

The hardware Management Console and Support Element Console operation guides provide additional information about the tasks used to configure these security controls.

Appendix A. Building an application

All of the files necessary to build and run an API application are available on Resource Link at http://www.ibm.com/servers/resourcelink. Click on **Services**, and then Click **API**.

Appendix B. HWMCA_EVENT_COMMAND_RESPONSE return codes

Following is a list of HWMCA_EVENT_COMMAND_RESPONSE return codes and their descriptions. The return code values are shown as hexadecimal values with the decimal equivalent in parentheses.

0806000A (134610954) **Resource unknown.**

Explanation

The profile name specified in an operations command is not recognized by the receiving node.

Programmer response

Correct the configuration identifier and re-send the request.

08090000 (134807552) **Mode inconsistency: The requested function cannot be performed in the present state of the receiver.**

Explanation

This command is prohibited because the target is in an incompatible mode. For example, an ITIMER request is not accepted when the system is power-on reset in LPAR mode.

Programmer response

This function cannot be performed in the present state of the receiver. Retry the request after the target mode status has changed.

08090001 (134807553) **Mode inconsistency: The requested function cannot be performed in the present state of the receiver.**

Explanation

Acceptance of the command is prohibited because the target is in an incompatible mode. For example, an ITIMER request is not accepted when the system is power-on reset in LPAR mode.

Programmer response

None. This function cannot be performed in the present state of the receiver.

080A000A (134873098) **Permission rejected: The receiver had denied an implicit or explicit request of the sender.**

Explanation

A STATLEV request was rejected because it was not compatible with the status reporting values set in the receiver.

Programmer response

Correct the STATLEV value and re-send the request.

080C0005 (135004165) **Procedure not supported: A procedure specified is not supported in the receiver.**

Explanation

The command is not supported.

Programmer response

re-send the request using a supported command, if possible.

080C0007 (135004167) **Procedure not supported: A procedure specified is not supported in the receiver.**

Explanation

A request for a function is supported by the receiver, but the resource identified in the request does not support that function.

Programmer response

None. This function cannot be canceled.

08120000 (135397376) **Insufficient resource: The receiver cannot act on the request because of a temporary lack of resource.**

Explanation

System resources are temporarily busy.

Programmer response

re-send command if required.

08120011 (135397393) **Insufficient resource: The receiver cannot act on the request because of a temporary lack of resource.**

Explanation

Insufficient storage is available to the target component to satisfy the request.

Programmer response

re-send command.

08150001 (135593985) **Function active: A request to activate an element or procedure was received, but the element or procedure was already active.**

Explanation

Unable to perform the command because the target CPC Subset or CPC Image is operational and the force operand has not indicated the override selection.

Programmer response

Put the system in the appropriate state and re-send the command.

081A0000 (135921664) **Request sequence error.**

Explanation

Unable to perform the command because the target partition is in the deactivated state.

Programmer response

Activate the logical partition, then re-send the original request.

081A0009 (135921673) **Request sequence error.**

Explanation

Unable to perform command because power is not on.

Programmer response

Send a POWERON or ACTIVATE command, then re-send the original request.

081A000A (135921674) **Request sequence error.**

Explanation

Unable to perform command because power-on reset is not complete.

Programmer response

Send a POWERON or ACTIVATE command, then re-send the original request.

081A000B (135921675) **Request sequence error.**

Explanation

Unable to perform command because the targeted CPU is not in the stopped state.

Programmer response

Send a STOP command, then re-send the original request.

081A000E (135921678) **Request sequence error.**

Explanation

Unable to perform command because the interval timer is present only when the CPC Image is operating in S/370 mode.

Programmer response

None. The requested command cannot be performed when the system is power-on reset in either ESA/390 mode or LPAR mode.

081A0010 (135921680) **Request sequence error.**

Explanation

The request is rejected or failed because the target resource is already in the state or condition that the request would have provided.

Programmer response

None. The requested command has already been performed.

081C0005 (136052741) **Request not executable: The requested function cannot be executed because of a permanent error condition in the receiver.**

Explanation

A power-on request failed.

Programmer response

Verify that power is available and re-send the command.

081C0006 (136052742) **Request not executable: The requested function cannot be executed because of a permanent error condition in the receiver.**

Explanation

A POR(YES) or POR(IML) failed. This may be accompanied by a hardware alert.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

081C0007 (136052743)	**Request not executable: The requested function cannot be executed because of a permanent error condition in the receiver.**

Explanation

An operating system load request (for example, LOAD) failed.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

081C000A (136052746)	**Request not executable: A POWEROFF request cannot be performed because of a permanent error condition in the receiver.**

Explanation

A power off request failed due to an unexpected power status.

Programmer response

Reset any abnormal power conditions at the receiver, such as tripped CBs, and retry the power off command. Call for service if the problem persists.

081C00BA (136052922)	**Request not executable: The requested function cannot be executed because of a permanent error condition in the receiver.**

Explanation

The receiver has an error resulting from a licensed internal code problem that prevents execution of the request.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

082D0001 (137166849)	**Busy.**

Explanation

Resources needed to process the request are being used.

Programmer response

Wait for the resources to be released, then re-send the request.

08380000 (137887744)	**Request not executable because of resource or component state incompatibility: The request is not executable because it is not compatible with the state of a resource or component in the receiver.**

Explanation

Unable to perform the command because the system is in an invalid state.

Programmer response

Put the system in a state that is compatible with the requested command and re-send the request.

0838001B (137887771)	**Request not executable because of resource or component state incompatibility: The request is not executable because it is not compatible with the state of a resource or component in the receiver.**

Explanation

Request will not be honored because it was submitted to a node at a time when a local operator or other application reserved control of the node.

Programmer response

Request the local operator to release control (log off), or retry later.

08380037 (137887799)	**MVS is not receiving. The request is not executable because the MVS operating system is not able to respond because it is in an inactive or quiesced state.**

Explanation

Request will not be honored because it requires that the resource operating system is in an active state.

Programmer response

Reissue the command after the operating system has been reactivated.

084F0000 (139395072)	**Resource not available: A requested resource is not available to service the given request.**

Explanation

A resource error exits which may indicate a configuration problem or insufficient resource to execute the command.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

085B0000 (140181504)	**Unknown resource name: The identified resource required to complete the requested command is not known.**

Explanation

The profile name specified in the AUTOACT operand of the RESET profile is not recognized by the receiving node.

Programmer response

Correct the profile name and re-send the request.

085C0000 (140247040)	**System exception. The node experiences an exception condition within a resident system or subsystem that inhibits further processing by the component.**

Explanation

An internal error has occurred with the processing of this request. This may be accompanied by a hardware alert.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

085C0001 (140247041)	**System exception: The node experiences an exception condition within a resident system or subsystem that inhibits further processing by the component.**

Explanation

The exception is identifiable as a system-related problem. This may be accompanied by a hardware alert.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

085C0002 (140247042)	**System exception: The node experiences an exception condition within a resident system or subsystem that inhibits further processing by the component.**

Explanation

The exception is identified as a permanent system-related problem. This may be accompanied by a hardware alert.

Programmer response

If the code is returned for an ACTIVATE request, to complete activation, send another ACTIVATE request to complete the initial program load.

For all other requests, retry the operation. Contact the IBM support system if the problem persists.

08B20002 (145883138)	**Data transmission failure: The data transmission between an application in the support element and an application in the processor was incomplete, causing abnormal termination of the function.**

Explanation

A time-out has occurred while waiting for transmission of data between two applications.

Programmer response

Retry operation. Contact the IBM support system if the problem persists.

100B0001 (269156353)	**Required structure absent.**

Explanation

An operand required by the command was not found in the command string.

Programmer response

Enter the required operand and re-send the request.

100B0003 (269156355) **Multiple occurrences of a nonrepeatable structure.**

Explanation

A value that cannot be repeated was detected in the command string.

Programmer response

Change the duplicate value(s) to unique value(s) and re-send the request.

100B0006 (269156358) **Length outside specified range.**

Explanation

The length of the operand indicated in SDATA is outside the allowable range.

Programmer response

Correct the operand data value and re-send the request.

100B000B (269156363) **Precluded combination of structures and data values present.**

Explanation

One command operand or data value is in conflict with one or more other operands or data values.

Programmer response

Remove the precluded operand(s) or correct the command and re-send the request. Also check the activation profile(s) used for activation, as the error may be the result of incorrect profile data.

100B000C (269156364) **Unknown or unsupported data value.**

Explanation

The data value in the operand indicated by SDATA is either unknown or unsupported.

Programmer response

Correct the operand data value and re-send the request.

100B000D (269156365) **Incompatible data values.**

Explanation

The data value in the operand indicated by SDATA is not compatible with this or other values.

Programmer response

Correct the conflicting operand data value and re-send the request.

100B0012 (269156370) **Recognized but unsupported structure.**

Explanation

The operand indicated by SDATA is recognized but not supported by the target support element.

Programmer response

Remove the unsupported operand and re-send the request.

80180002 (2149056514) **Resource unknown.**

Explanation

The secondary OCR specified in the OCFNAME operand is not recognized.

Programmer response

Ensure that the system is power-on reset in LPAR mode and the secondary name in the OCFNAME operand matches a logical partition name in the active IOCDS.

Appendix C. API return codes

Data exchange API call return codes

Following is a list of return codes and their descriptions, which can be returned from the various Data Exchange API calls. (The decimal values are shown in parentheses).

(0) HWMCA_DE_NO_ERROR

Explanation

A Data Exchange API call has completed successfully.

Programmer response

None.

(1) HWMCA_DE_NO_SUCH_OBJECT

Explanation

A Data Exchange API call specified an object identifier that does not exist.

Programmer response

Check the specified object identifier to ensure that it is valid and that the API support is enabled and functioning correctly on the target console.

(2) HWMCA_DE_INVALID_DATA_TYPE

Explanation

A HwmcaSet Data Exchange API call specified an invalid data type.

Programmer response

Check the specified data type value to ensure that is one of the supported values and that is appropriate for the target object identifier.

(3) HWMCA_DE_INVALID_DATA_LENGTH

Explanation

Either a HwmcaSet Data Exchange API call specified a data length value that is not appropriate for the corresponding data type or is not appropriate for the target object identifier, or the result of a HwmcaGet or HwmcaGetNext is too large to be transported by the underlying transport protocol.

Programmer response

Either check to ensure that a length of zero is used for a data type of HWMCA_TYPE_NULL and that a length of 1, 2, or 4 is used for a data type of HWMCA_TYPE_INTEGER, or use an alternative approach for retrieving the desired data.

(4) HWMCA_DE_INVALID_DATA_PTR

Explanation

A HwmcaSet Data Exchange API call specified a data pointer that is not appropriate for the corresponding data type.

Programmer response

Check to ensure that a null pointer is used for a data type of HWMCA_TYPE_NULL and that a non-null pointer is used for all other data types.

(5) HWMCA_DE_INVALID_DATA_VALUE

Explanation

A HwmcaSet Data Exchange API call specified a data value that is not appropriate for the target object identifier.

Programmer response

Check to make sure that the data value is one of the allowed value for the target object identifier.

(6) HWMCA_DE_INVALID_INIT_PTR

Explanation

A Data Exchange API call specified null as the pointer to the HWMCA_INITIALIZE_T structure.

Programmer response

Make sure that this value is specified as a pointer to a valid HWMCA_INITIALIZE_T structure.

(7) HWMCA_DE_INVALID_ID_PTR

Explanation

A Data Exchange API call specified a null pointer as the object identifier parameter.

Programmer response

Make sure that this value is specified as a pointer to a valid object identifier string.

(8) HWMCA_DE_INVALID_BUF_PTR

Explanation

A Data Exchange API call specified null as the pointer to the output buffer.

Programmer response

Make sure that this value is specified as a pointer to an address of the output buffer.

(9) HWMCA_DE_INVALID_BUF_SIZE

Explanation

A Data Exchange API call specified zero as the length of the output buffer.

Programmer response

Make sure that this parameter is a non-zero value.

(10) HWMCA_DE_INVALID_DATATYPE_ PTR

Explanation

A HwmcaSet Data Exchange API call specified null as the pointer to the HWMCA_DATATYPE_T structure used to describe the data to be used for the set operation.

Programmer response

Make sure that this value is specified as a pointer to an address of a valid HWMCA_DATATYPE_T structure.

(11) HWMCA_DE_INVALID_TARGET

Explanation

A HwmcaInitialize Data Exchange API call specified an invalid host name or internet address for the target console.

Programmer response

. Make sure that the value pointed to by the pHost field of the HWMCA_SNMP_TARGET_T structure is internet address or hostname.

(12) HWMCA_DE_INVALID_EVENT_ MASK

Explanation

A HwmcaInitialize Data Exchange API call specified a value in the ulEventMask field of the HWMCA_INITIALIZE_T structure that is not valid.

Programmer response

Make sure that this field only contains some combination of the valid event mask values.

(13) HWMCA_DE_INVALID_ PARAMETER

Explanation

A Data Exchange API call specified an invalid parameter. Depending on the API call being made, one of the following problems occurred:

HwmcaInitialize

- The HWMCA_INITIALIZE_T structure used on a previous HwmcaInitialize call specifies a host name or internet address specified that is different that what was initially specified.
- The ulReserved field of HWMCA_INITIALIZE_T structure contains a non-null value.

HwmcaBuildAttributeId
The pointer to the attribute suffix string was specified as a null pointer.

HwmcaGet or HwmcaGetNext or HwmcaWaitEvent
The pointer to the value to be filled in with the number of bytes needed for the output buffer was specified as a null pointer.

(14) HWMCA_DE_READ_ONLY_OBJECT

Explanation

A HwmcaSet Data Exchange API call specified a target object identifier that is read only.

Programmer response

Make sure to use a target object identifier that allows for write access.

(15) HWMCA_DE_SNMP_INIT_ERROR

Explanation

A HwmcaInitialize Data Exchange API call encountered an error trying to create/allocate the internal resources necessary to complete the operation. For example, memory could not be

allocated successfully or TCP/IP sockets could not be created.

Programmer response

Make sure that the necessary resources are available to be used on the requesting machine.

(16) HWMCA_DE_INVALID_OBJECT_ID

Explanation

A Data Exchange API call was made with an invalid object identifier.

Programmer response

Check the object identifier specified on the call to make sure that it is specified correctly and is a valid object identifier.

(17) HWMCA_DE_REQUEST_ALLOC_ERROR

Explanation

A Data Exchange API call encountered an error trying to allocate some temporary storage for internal use.

Programmer response

Make sure that enough memory is available on the requesting machine.

(18) HWMCA_DE_REQUEST_SEND_ERROR

Explanation

A Data Exchange API encountered an error trying to send a request to the target console.

Programmer response

This is typically due to a network error of some sort.

(19) HWMCA_DE_TIMEOUT

Explanation

A Data Exchange API timed out while waiting for a response. For the HwmcaWaitEvent API call, this simply means that no events were received within the specified time period, so the calling application should proceed accordingly. For other Data Exchange API calls, the response was not received within the specified time period.

Programmer response

Make sure that the timeout value is large enough to allow for the request to be completed and the response to be returned.

(20) HWMCA_DE_REQUEST_RECV_ERROR

Explanation

A Data Exchange API encountered an error trying to receive a response from the target console. This is typically due to a network error of some sort.

Programmer response

Investigate the possibility of a network error.

(21) HWMCA_DE_SNMP_ERROR

Explanation

A Data Exchange API call received a response that contained an unrecognized error status value.

Programmer response

Make sure that no errors were reported on the target console that would have resulted in incomplete or invalid data to be sent as a response.

(22) HWMCA_DE_INVALID_TIMEOUT

Explanation

A Data Exchange API call was made with the timeout value specified as zero.

Programmer response

Make sure that an appropriate non-zero timeout value is specified on the API call.

(28) HWMCA_DE_INVALID_HOST

Explanation

A HwmcaInitialize Data Exchange API call was made with a null pointer value specified as the pHost field of the HWMCA_SNMP_TARGET_T structure.

Programmer response

Make sure this field points to a valid hostname or internet address.

(29) HWMCA_DE_INVALID_COMMUNITY

Explanation

A HwmcaInitialize Data Exchange API call was made with a zero length string value specified as the szCommunity field of the HWMCA_SNMP_TARGET_T structure.

Programmer response

Make sure that this field contains a community name string with a length greater than zero.

(30) HWMCA_DE_INVALID_QUALIFIER

Explanation

A HwmcaInitialize Data Exchange API call was made that specified event qualification data, but the ulType field of the HWMCA_EVENT_QUALIFIER_T structure contained an invalid value.

Programmer response

Make sure that this field contains a valid event qualifier type value.

(98) HWMCA_DE_REQUIRES_QUALIFIER

Explanation

A HwmcaInitialize Data Exchange API call was made that specified an event mask indicating that requires additional event qualification information to be provided.

Programmer response

Make sure to provide the necessary event qualification information.

(99) HWMCA_TRANSPORT_ERROR

Explanation

A Data Exchange API call was made but an error was encountered in the transport layer that was specified to deliver the request data and return the response data.

Programmer response

Refer to information about the specific transport layer being used for more details regarding this error.

Command API call return codes

Following is a list of return codes and their descriptions, which can be returned from the various Command API call. (The decimal values are shown in parentheses).

(0) HWMCA_CMD_NO_ERROR

Explanation

A HwmcaCommand API call has completed successfully.

Programmer response

None.

(1) HWMCA_CMD_NO_SUCH_OBJECT

Explanation

A HwmcaCommand API call specified an object identifier that does not exist.

Programmer response

Check the specified object identifier to ensure that it is valid and that the API support is enabled and functioning correctly on the target console.

(2) HWMCA_CMD_INVALID_DATA_TYPE

Explanation

A HwmcaCommand API call specified an invalid data type.

Programmer response

Check the specified data type value to ensure that is one of the supported values and that is appropriate for the target object identifier.

(3) HWMCA_CMD_INVALID_DATA_LENGTH

Explanation

A HwmcaCommand API call specified a data length value that is not appropriate for the corresponding data type or is not appropriate for the target object identifier.

Programmer response

Check to ensure that a length of zero is used for a data type of HWMCA_TYPE_NULL and that a length of 1, 2, or 4 is used for a data type of HWMCA_TYPE_INTEGER.

(4) HWMCA_CMD_INVALID_DATA_PTR

Explanation

A HwmcaCommand API call specified a data pointer that is not appropriate for the corresponding data type.

Programmer response

Check to ensure that a null pointer is used for a data type of HWMCA_TYPE_NULL and that a non-null pointer is used for all other data types.

(5) HWMCA_CMD_INVALID_DATA_VALUE

Explanation

A HwmcaCommand API call specified a data value that is not appropriate for the target object identifier.

Programmer response

Check to make sure that the data value is one of the allowed value for the target object identifier.

(6) HWMCA_CMD_INVALID_INIT_PTR

Explanation

A HwmcaCommand API call specified null as the pointer to the HWMCA_INITIALIZE_T structure.

Programmer response

Make sure that this value is specified as a pointer to a valid HWMCA_INITIALIZE_T structure.

(7) HWMCA_CMD_INVALID_ID_PTR

Explanation

A HmcaCommand API call specified a null pointer as the object identifier parameter.

Programmer response

Make sure that this value is specified as a pointer to a valid object identifier string.

(10) HWMCA_CMD_INVALID_DATATYPE_PTR

Explanation

A HwmcaCommand API call specified null as the pointer to the HWMCA_DATATYPE_T structure used to describe the data to be used for the command parameter information.

Programmer response

Make sure that this value is specified as a pointer to an address of a valid HWMCA_DATATYPE_T structure.

(11) HWMCA_CMD_INVALID_TARGET

Explanation

A HwmcaCommand API call specified an invalid target.

Programmer response

Check the specified object identifier to ensure that it is valid and that there is not already an active command for the specified object.

(13) HWMCA_CMD_INVALID_PARAMETER

Explanation

A HwmcaCommand API call specified an invalid parameter.

Programmer response

Make sure all the required parameters are correctly specified.

(17) HWMCA_CMD_REQUEST_ALLOC_ERROR

Explanation

A HwmcaCommand PI call encountered an error trying to allocate some temporary storage for internal use.

Programmer response

Make sure that enough memory is available on the requesting machine.

(18) HWMCA_CMD_REQUEST_SEND_ERROR

Explanation

A HwmcaCommand API encountered an error trying to send a request to the target console. This is typically due to a network error of some sort.

Programmer response

Investigate the possibility of a network error.

(19) HWMCA_CMD_TIMEOUT

Explanation

A HwmcaCommand API timed out while waiting for a response. The response was not received within the specified time period.

Programmer response

Make sure that the timeout value is large enough to allow for the request to be completed and the response to be returned.

(20) HWMCA_CMD_REQUEST_RECV_ERROR

Explanation

A HwmcaCommand API encountered an error trying to receive a response from the target console. This is typically due to a network error of some sort.

Programmer response

Investigate the possibility of a network error.

(21) HWMCA_CMD_SNMP_ERROR

Explanation

A HwmcaCommand API call received a response that contained an unrecognized error status value.

Programmer response

Make sure that no errors were reported on the target console that would have resulted in incomplete or invalid data to be sent as a response.

(22) HWMCA_CMD_INVALID_TIMEOUT

Explanation

A HwmcaCommand API call was made with the timeout value specified as zero.

Programmer response

Make sure that an appropriate non-zero timeout value is specified on the API call.

(23) HWMCA_CMD_INVALID_CMD

Explanation

A HwmcaCommand API call was made with an invalid command object identifier.

Programmer response

Make sure that the command object identifier corresponds to a valid command for the target object.

(24) HWMCA_CMD_OBJECT_BUSY

Explanation

A HwmcaCommand API call was made specifying a target object identifier for an object that is currently busy performing another command.

Programmer response

. Retry the call again after the target object is no longer busy.

(25) HWMCA_CMD_INVALID_OBJECT

Explanation

A HwmcaCommand API call was made with a target object identifier that is not valid for the specified command.

Programmer response

Make sure that the command object identifier corresponds to an appropriate command for the target object.

(26) HWMCA_CMD_COMMAND_FAILED

Explanation

A HwmcaCommand API call failed due to an internal error on the target console.

Programmer response

Check the target console for details regarding the error.

(27) HWMCA_CMD_INITTERM_OK

Explanation

This is a value only used internally and should never be received by the calling application.

HWMCA_EVENT_COMMAND_RESPONSE return codes

The following values are returned as HWMCA_EVENT_COMMAND_RESPONSE return codes for HWMCA_ACTIVATE_CBU_COMMAND, HWMCA_UNDO_CBU_COMMAND, HWMCA_ADD_CAPACITY_COMMAND, and HWMCA_REMOVE_CAPACITY_COMMAND command requests.

(26) HWMCA_CMD_COMMAND_FAILED

Explanation

A command call failed due to an internal error on the target console and a more specific failure code was not provided by the command.

Programmer response

Check the target console for details regarding the error.

(28) HWMCA_CMD_CBU_DISRUPTIVE_OK

Explanation

The command request was successful but requires a system IML for the changes to take effect.

Programmer response

Perform a system IML.

(29) HWMCA_CMD_CBU_PARTIAL_ HW

Explanation

The command request was successful for the available hardware, but complete success for the command request could not be achieved (probably due to defective hardware).

(30) HWMCA_CMD_CBU_NO_SPARES

Explanation

The command request was unsuccessful because the required hardware was not available.

(31) HWMCA_CMD_CBU_TEMPORARY

Explanation

The command request was successful, but there was a problem updating the system SEEPROM so the new capacity will be lost at the next system IML.

(32) HWMCA_CMD_CBU_NOT_ENABLED

Explanation

The command request failed because the CBU feature is not enabled for the target console.

(33) HWMCA_CMD_CBU_NOT_AUTHORIZED

Explanation

The command request failed because the target console is not authorized for the requested command.

(34) HWMCA_CMD_CBU_FAILED

Explanation

The command request failed due to an internal error.

Programmer response

Check the target console for details about the failure.

(35) HWMCA_CMD_CBU_ALREADY_ACTIVE

Explanation

The command request failed because there is already a previous CBU request in effect.

(36) HWMCA_CMD_CBU_INPROGRESS

Explanation

The command request cannot be performed at this time because another operation is being performed at this time.

Programmer response

The command request can be retried at a later time when the currently executing operation is complete.

(37) HWMCA_CMD_CBU_CPSAP_SPLIT_CHG

Explanation

The command request cannot be performed at this time because the current CP/SAP allocation for the machine differs from what it was originally.

Programmer response

The command request can be retried at a later time after the CP/SAP split matches the original values for the machine.

(38) HWMCA_CMD_INVALID_ MACHINE_STATE

Explanation

The command request cannot be performed at this time because the target object is currently not in an appropriate state (i.e. it is not powered on).

Programmer response

The command request can be retried at a later time when the target object is in an appropriate state.

(39) HWMCA_CMD_NO_RECORDID

Explanation

The command request was unsuccessful because the specified a capacity record with the specified identifier does not exist.

Programmer response

The command request can be retried at a later time with an identifier for an existing capacity record.

(40) HWMCA_CMD_NO_SW_ MODEL

Explanation

The command request was unsuccessful because the specified software model is invalid for the target object.

Programmer response

The command request can be retried at a later time with a software model that is valid for the target object.

(41) HWMCA_CMD_NOT_ENOUGH_ RESOURCES

Explanation

The command request was unsuccessful because the request specifies more resources than are currently available on the target object.

Programmer response

The command request can be retried at a later time with a number of resources that are available on the target object.

(42) HWMCA_CMD_NOT_ENOUGH_ ACTIVE_RESOURCES

Explanation

The command request was unsuccessful because the request specifies more resources than are currently active on the target object.

Programmer response

The command request can be retried at a later time with a number of resources less than or equal to those that are currently active on the target object.

(43) HWMCA_CMD_ACT_LESS_ RESOURCES

Explanation

The command request for additional resources was unsuccessful because the request specifies a net decrease in the resources for the target object.

Programmer response

The command request can be retried at a later time with an increase of resources for the target object.

(44) HWMCA_CMD_DEACT_MORE_ RESOURCES

Explanation

The command request for a removal of resources was unsuccessful because the request specifies a net increase in the resources for the target object.

Programmer response

The command request can be retried at a later time with a decrease of resources for the target object.

(45) HWMCA_CMD_ACT_TYPE_ MISMATCH

Explanation

The command request was unsuccessful because the type value specified (real or test) was not valid for the type of capacity record being used.

Programmer response

The command request can be retried at a later time with a valid type value for the capacity record.

(46) HWMCA_CMD_API_NOT_ ALLOWED

Explanation

The command request was unsuccessful because the target Defined CPC was configured to not allow capacity changes via the Console Application Programming Interfaces.

Programmer response

In order to be successful, capacity changes must be allowed for the Console Application Programming Interfaces via the Customized API Settings task.

(47) HWMCA_CMD_CDU_IN_PROGRESS

Explanation

The command request cannot be performed at this time because a concurrent driver upgrade operation is being performed at this time.

Programmer response

The command request can be retried at a later time when the currently executing operation is complete.

(48) HWMCA_CMD_MIRRORING_RUNNING

Explanation

The command request cannot be performed at this time because a support element mirror operation is being performed at this time.

Programmer response

The command request can be retried at a later time when the currently executing operation is complete.

(49) HWMCA_CMD_COMMUNICATIONS_NOT_ACTIVE

Explanation

The command request cannot be performed at this time because communication with the Defined CPC object is not active.

Programmer response

The command request can be retried at a later time when there is active communications with the Defined CPC object.

(50) HWMCA_CMD_RECORD_EXPIRED

Explanation

The command request was unsuccessful because the capacity record being used for the operation has expired.

Programmer response

The command request can be retried at a later time with after the capacity record expiration date has been extended or a different (not expired) capacity record can be used.

(51) HWMCA_CMD_PARTIAL_CAPACITY

Explanation

The command request was successful, but not all the resources requested could be made available. There are two cases that can result in this return code:

1. The request was issued with priority and not enough free resources existed to completely satisfy the request. The additional resources to completely satisfy the request are pending and will be associated with the request as soon as they are no longer being used for other purposes.
2. At the time of the request there were sufficient resources to completely satisfy the request, but during while these resources were being brought online for the request something happened to cause the number of available resources to no longer be sufficient (i.e. a defective processor was detected).

(52) HWMCA_CMD_INVALID_REQUEST

Explanation

The command request was unsuccessful because it is not valid to be performed at this time due to one of the following conditions.

- An EDM (Enhanced Driver Maintenance) operation is currently in progress.
- The target of the request is configured to not allow capacity change API requests.
- The targeted system is not in the correct state to perform the request.

Programmer response

The command request can be retried at a later time after the system is in the correct state to allow the operation.

(53) HWMCA_CMD_ALREADY_ACTIVE

Explanation

The command request was unsuccessful because there is a different capacity record that is already active.

Programmer response

The command request can be retried at a later time after the currently active capacity record is no longer active or a different (the currently active) capacity record can be used.

(54) HWMCA_CMD_RESERVE_HELD

Explanation

The command request was unsuccessful because a task running on the targeted system has reserved control.

Programmer response

Wait until the task is complete and retry the command request.

(55) HWMCA_CMD_GENERAL_XML_PARSING_ERROR

Explanation

The command request was unsuccessful because the XML specified on the command is not well formed and could not be parsed properly.

Programmer response

Retry the command request with a well formed XML document.

(56) HWMCA_CMD_STP_NOT_ENABLED

Explanation

The command request was unsuccessful because STP is not enabled on the target system.

Programmer response

Verify that the request is targeted toward a system that has the STP feature enabled.

(57) HWMCA_CMD_STP_MUST_TARGET_CTS

Explanation

The command request failed because the targeted system is not the Current Time Server for the STP-only Coordinated Timing Network (CTN) or the system specified to become the Current Time Server does not match the targeted system.

Programmer response

Verify that the request is targeted toward the system that will be the Current Time Server after the command request and resubmit the request.

(58) HWMCA_CMD_STP_INVALID_CONFIG_SPECIFIED

Explanation

The command request was unsuccessful because the specified STP configuration is not valid.

Programmer response

Verify that the current configuration will support the command request and that the configuration specified in the command request is valid. Retry the command request with an appropriate STP configuration.

(59) HWMCA_CMD_STP_WRONG_CTN

Explanation

The command request was unsuccessful because the CTN ID of the current STP configuration is not valid for the request. This is most likely the result of the configuration changing between the time that the command request was created and the time the request was processed.

Programmer response

Verify that the current configuration will support the command request and that the configuration specified in the command request is valid. Then retry the command request with an appropriate STP configuration.

(60) HWMCA_CMD_STP_NOT_VALID_FOR_CTS

Explanation

The command request cannot be processed on the Current Time Server. Certain actions are not allowed on the Current Time Server because the result would be disruptive to the entire STP-only CTN.

Programmer response

Verify that the request is targeted toward the appropriate system.

(61) HWMCA_CMD_STP_IN_ETR_MIGRATION

Explanation

The command request was unsuccessful because the CPC is a member of an STP-only CTN that is migrating back to a Mixed CTN, which uses a Sysplex Timer. STP-related commands are not allowed until this procedure is complete.

Programmer response

Determine the appropriate action after the ETR migration is complete.

(62) HWMCA_CMD_STP_NODE_NOT_FOUND_IN_SYSTEM_LIST

Explanation

The command request was unsuccessful because the specified NodeName could not be converted into a NodeID.

Programmer response

The system referenced in the NodeName tag needs to be a Defined CPC object on the HMC console. Add the object to the HMC and retry the command request.

(63) HWMCA_CMD_STP_CTNID_TAG_ERROR

Explanation

The command request was unsuccessful because the CTN ID portion of the set STP configuration XML was not correct.

Programmer response

Retry the command request after verifying that the CTN ID information specified is in the proper format.

(64) HWMCA_CMD_STP_NODE_TAG_ERROR

Explanation

The command request was unsuccessful because the Preferred Time Server, Backup Time Server, or Arbiter portion of the set STP configuration XML was not correct.

Programmer response

Retry the command request after verifying that the node information specified is in the proper format.

(65) HWMCA_CMD_STP_CONFIG_TAG_NOT_FOUND

Explanation

The command request was unsuccessful because the STPConfiguration tag was not found in the set STP configuration XML.

Programmer response

Retry the command request specifying STPConfiguration as the outermost tag of the XML document.

(66) HWMCA_CMD_STP_ACTIVE_CTS_TAG_ERROR

Explanation

The command request was unsuccessful because the CurrentTimeServer portion of the set STP configuration XML was not correct.

Programmer response

Retry the command request verifying that the CurrentTimeServer information specified is in the proper format.

(67) HWMCA_CMD_STP_INITIALIZE_INCOMPLETE

Explanation

The command request was unsuccessful because the specified STP configuration cannot be set until initialization of the STP-only CTN is complete. The time zone and leap second values need to be set.

Programmer response

Manually set the time zone and/or leap second values via the Initialize Time button on the Network Configuration tab in the System (Sysplex) Time task and retry the command request.

(68) HWMCA_CMD_STP_INVALID_STP_ID

Explanation

The command request was unsuccessful because the STP ID specified on the command was not correct.

Programmer response

Retry the command request specifying an STP ID with 1-8 valid characters.

(69) HWMCA_CMD_STP_LINKS_ERROR

Explanation

The command request was unsuccessful because the communication links between the Preferred Time Server, Backup Time Server, and/or Arbiter systems in the STP-only CTN are not active.

Programmer response

Manually check the links between systems in the STP-only CTN with the roles of Preferred Time Server, Backup Time Server, and/or Arbiter or retry the command request with force, if applicable.

(70) HWMCA_CMD_STP_REQUIRES_FORCE_TO_CONFIGURE

Explanation

The command request was unsuccessful because of the current state of the targeted system. Verification of connections between systems with key roles in the STP-only CTN is done to ensure that the configuration will function properly. Once the connections are verified, the force parameter is required to ensure that the customer is not creating an island STP-only CTN.

Programmer response

Retry the set configuration command request with force, if applicable.

Data exchange and command API (REXX version) return codes

The following return codes are specific to the REXX version of the Data Exchange and Command APIs.

(1000) HWMCA_RX_INVALID_STEM_VAR

Explanation

The name of a REXX stem variable that was specified on one of the REXX API calls did not end with a ".".

Programmer response

Make sure the stem variable name ends with a ".".

Appendix D. APIs for Java (com.ibm.hwmca.api)

The purpose of the **com.ibm.hwmca.api** package is to allow Java™ applications, local or remote, the ability to exchange data related to the objects that the Console application manages. Specifically, this support allows other applications to request the Console application to:

- Query (Get/Get-Next) the attributes of objects
- Change (Set) certain attributes of objects
- Receive notification of significant events occurring to objects
- Generate enterprise-specific Simple Network Management Protocol traps for significant events occurring to objects.

The **com.ibm.hwmca.api** package uses the Simple Network Management Protocol (SNMP) as the transport mechanism. The attributes of objects can be queried/changed through the underlying SNMP Set, Get, Get-Next requests, while event notification is accomplished through the user of the enterprise-specific SNMP Trap message. The underlying SNMP protocol is encapsulated in several APIs in order to reduce the complexities for the application programmer. The **com.ibm.hwmca.api** package is part of the **HWMCAAPI.JAR** jar file. The most up to date copy of this file is available on Resource Link at http://www.ibm.com/servers/resourcelink. Click **Services**, and then click **API**.

For documentation describing this Java application, see *Application Programming Interfaces for Java*.

Appendix E. Object Attribute Availability

Except for the attributes found in Table 2 on page 203, it can be assumed that each object attribute described in Chapter 4, “Console application managed objects,” on page 75 is valid for any level of object. The following table defines the required level of object for each attribute.

Table 2. Level of objects required on attributes

Attribute	Availability
Console Attributes	
Version	Available on consoles version 2.9.2 or later
Internet Protocol (IP) Addresses	Available on consoles version 2.10.0 or later
Engineering Change (EC)/Microcode Level (MCL)	Available on consoles version 2.10.0 or later
Defined CPC attributes	
On/Off Capacity on Demand (On/Off CoD) installed	Available for Defined CPCs version 2.9.0 or later.
On/Off Capacity on Demand (On/Off CoD) activated	Available for Defined CPCs version 2.9.0 or later.
On/Off Capacity on Demand (On/Off CoD) enabled	Available for Defined CPCs version 2.9.0 or later.
On/Off Capacity on Demand (On/Off CoD) activation date	Available for Defined CPCs version 2.9.0 or later.
List of group profiles	Available for Defined CPCs version 2.9.2 or later.
Temporary capacity records	Available for Defined CPCs version 2.10.0 or later.
Permanent software model	Available for Defined CPCs version 2.10.0 or later.
Permanent plus billable software model	Available for Defined CPCs version 2.10.0 or later.
Permanent plus all temporary software model	Available for Defined CPCs version 2.10.0 or later.
Permanent MSU	Available for Defined CPCs version 2.10.0 or later.
Permanent plus billable MSU	Available for Defined CPCs version 2.10.0 or later.
Permanent plus all temporary MSU	Available for Defined CPCs version 2.10.0 or later.
General purpose processors	Available for Defined CPCs version 2.10.0 or later.
Service assist processors	Available for Defined CPCs version 2.10.0 or later.
Application Assist Processor (zAAP) processors	Available for Defined CPCs version 2.10.0 or later.
Integrated Facility for Linux (IFL) processors	Available for Defined CPCs version 2.10.0 or later.
Internal Coupling Facility (ICF) processors	Available for Defined CPCs version 2.10.0 or later.
z Integrated Information Processors (zIIP) processors	Available for Defined CPCs version 2.10.0 or later.
Container Based Processors (CBP) processors	Available for Defined CPCs version 2.14.0 with the latest available microcode.
Defective processors	Available for Defined CPCs version 2.10.0 or later.
Spare processors	Available for Defined CPCs version 2.10.0 or later.
Pending processors	Available for Defined CPCs version 2.10.0 or later.

Table 2. Level of objects required on attributes (continued)	
Attribute	**Availability**
Temporary capacity change allowed	Available for Defined CPCs version 2.10.0 or later.
Version	Available for Defined CPCs version 2.9.2 or later or version 1.8.2 with the latest level of microcode applied.
Server Time Protocol (STP) configuration	Available for Defined CPCs, with STP enabled, 2.9.2 or later or 2.9.0 with the latest available microcode.
Pending General Purpose Processors	Available for Defined CPCs version 2.10.1 or later.
Pending Service Assist Processors	Available for Defined CPCs version 2.10.1 or later.
Pending Application Assist Processor (zAAP) Processors	Available for Defined CPCs version 2.10.1 or later.
Pending Integrated Facility for Linux (IFL) Processors	Available for Defined CPCs version 2.10.1 or later.
Pending Internal Coupling Facility (ICF) Processors	Available for Defined CPCs version 2.10.1 or later.
Pending z Integrated Information Processors (zIIP) Processors	Available for Defined CPCs version 2.10.1 or later.
Pending Container Based Processors (CBP) processors	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
CPC Image attributes	
Initial z Integrated Information Processors processing weight	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Initial z Integrated Information Processors processing weight capped	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Minimum z Integrated Information Processors processing weight	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Maximum z Integrated Information Processors processing weight	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Current z Integrated Information Processors processing weight	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Current z Integrated Information Processors processing weight capped	Available for CPC Images running on Defined CPCs version 2.9.0 or later.
Program Status Word (PSW) information	Available for CPC Images running on Defined CPCs version 2.10.0 or later.
IPL Token	Available for CPC Images running on Defined CPCs version 2.10.1 or later.
Group Profile capacity	Available for CPC Images running on Defined CPCs version 2.9.2 or later.
Initial Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Initial Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.

Table 2. Level of objects required on attributes (continued)	
Attribute	**Availability**
Minimum Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Minimum Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Maximum Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Maximum Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Current Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Current Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Container Based Processor absolute capping type	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Container Based Processor absolute capping value	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Group Profile attributes	
Capacity	Available for Defined CPCs version 2.9.2 or later.
Image Activation Profile attributes	
Group profile name	Available for Defined CPCs version 2.9.2 or later.
Number of dedicated Application Assist Processor (zAAP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved dedicated Application Assist Processors (zAAP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of dedicated Integrated Facility for Linux (IFL) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved dedicated Integrated Facility for Linux (IFL) processors	Available for Defined CPCs version 2.9.2 or later.
Number of dedicated Internal Coupling Facility (ICF) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved dedicated Internal Coupling Facility (ICF) processors	Available for Defined CPCs version 2.9.2 or later.
Number of dedicated z Integrated Information Processors (zIIP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved dedicated z Integrated Information Processors (zIIP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of dedicated Container Based Processor (CBP) processors	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Number of reserved dedicated Container Based Processor (CBP) processors	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Number of shared general purpose processors	Available for Defined CPCs version 2.9.2 or later.

Table 2. Level of objects required on attributes (continued)	
Attribute	**Availability**
Number of reserved shared general purpose processors	Available for Defined CPCs version 2.9.2 or later.
Number of shared Application Assist Processor (zAAP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved shared Application Assist Processor (zAAP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of shared Integrated Facility for Linux (IFL) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved shared Integrated Facility for Linux (IFL) processors	Available for Defined CPCs version 2.9.2 or later.
Number of shared Internal Coupling Facility (ICF) processors	Available for Defined CPCs version 2.9.2 or later.
Number of reserved shared Internal Coupling Facility (ICF) processors	Available for Defined CPCs version 2.9.2 or later.
Number of shared z Integrated Information Processors (zIIP) processors	Available for Defined CPCs version 2.9.2 or later.
Number of shared Container Based Processor (CBP) processors	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Number of reserved shared Container Based Processor (CBP) processors	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Number of reserved shared z Integrated Information Processors (zIIP) processors	Available for Defined CPCs version 2.9.2 or later.
Initial Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Minimum Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Minimum Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Maximum Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Maximum Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Current Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Current Container Based Processor processing weight capped	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Container Based Processor absolute capping type	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Container Based Processor absolute capping value	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.
Initial Container Based Processor processing weight	Available for Defined CPCs version 2.14.0 or later with the latest available microcode.

Table 2. Level of objects required on attributes (continued)	
Attribute	**Availability**
Capacity Record attributes	
All attributes	Available for Defined CPCs version 2.10.0 or later.

Appendix F. XML descriptions

XML strings are used in several places throughout this document. This appendix defines the format of this XML and provides examples of each. It is important to keep in mind that while the XML returned and provided on input must be well formed and syntactically correct, this XML is not a complete document, but rather an XML fragment as illustrated with the examples. XML is used in the following areas of the Console Application Programming Interfaces.

Add capacity command

The input parameters for the HWMCA_ADD_CAPACITY_COMMAND on page "HWMCA_ADD_CAPACITY_COMMAND" on page 47 is specified using XML. Following is an example of this input; refer to the XML schema at the end of this appendix for the complete syntax definition of this XML.

```
<!--
 This example XML document illustrate the markup used to perform the following
 addition of temporary capacity :
 - change the general processors to a model A99
 - add 3 AAP processors
 - indicate the activation should take not priority
 - indicate the activation is not a "test", but a "real" activation
 -->
<add>
   <recordid>12345</recordid>
   <softwaremodel>A99</softwaremodel>
   <processorinfo>
      <type>AAP</type>
      <procstep>3</procstep>
   </processorinfo>
   <priority>false</priority>
   <test>false</test>
</add>
```

Remove capacity command

The input parameters for the HWMCA_REMOVE_CAPACITY_COMMAND on page "HWMCA_REMOVE_CAPACITY_COMMAND" on page 47 is specified using XML. Following is an example of this input; refer to the XML schema at the end of this appendix for the complete syntax definition of this XML.

```
<!--
 This example XML document illustrate the markup used to perform the following
 removal of temporary capacity :
 - change the general processors to a model A99
 - remove 3 AAP processors
 -->
<remove>
   <recordid>12345</recordid>
   <softwaremodel>A99</softwaremodel>
   <processorinfo>
      <type>AAP</type>
      <procstep>3</procstep>
   </processorinfo>
</remove>
```

Capacity record query

The output of a Get operation for a Capacity Record Object is an XML string that describes the record. Following is an example of this output; refer to the XML schema at the end of this appendix for the complete syntax definition of this XML.

```
<!--
 This example XML document illustrates the markup used to describe
 a capacity record that allows for:
   - 2 additional CPs with 2 additional speed steps
   - currently 1 CP with 1 speed step are active
 -->
<record xmlns="http://www.ibm.com/hwmca/api"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="file:///C:/Documents%20and%20Settings/Administrator/My%20Documents/
                            Test%20Java%20Code/xml ebod.xsd">
   <!-- Record id -->
   <recordid>12345</recordid>
   <!-- Offering type -->
   <recordtype>OOCOD</recordtype>
   <!-- Activation status -->
   <status>Real</status>
   <!-- Activation processor information. -->
   <processorinfo>
      <!-- Processor type. -->
      <type>CP</type>
      <!-- Processor count. -->
      <procstep>+1</procstep>
      <!-- Speed count. -->
      <speedstep>+1</speedstep>
      <!-- Maximum number of processors. -->
      <max>2</max>
      <!-- Remaining processor days. (-1 means unlimited) -->
      <remainingprocdays>99</remainingprocdays>
      <!-- Remaining MSU days. (-1 means unlimited) -->
      <remainingmsudays>99</remainingmsudays>
   </processorinfo>
   <processorinfo>
      <!-- Processor type. -->
      <type>AAP</type>
      <!-- Processor count. -->
      <procstep>+1</procstep>
      <!-- Maximum number of processors. -->
      <max>2</max>
      <!-- Remaining processor days. (-1 means unlimited) -->
      <remainingprocdays>99</remainingprocdays>
      <!-- Remaining MSU days. (-1 means unlimited) -->
      <remainingmsudays>99</remainingmsudays>
   </processorinfo>
   <!-- Activation start date (UTC). -->
   <activationstart>2006-07-04T11:11:11Z</activationstart>
   <!-- Activation expiration date. -->
   <activationexpiration>2006-08-04T11:11:11Z</activationexpiration>
   <!-- Record expiration date. -->
   <recordexpiration>2006-12-31T23:59:59Z</recordexpiration>
   <!-- Maximum real activation days. (-1 means unlimited) -->
   <maxrealdays>22</maxrealdays>
   <!-- Maximum test activation days. (-1 means unlimited) -->
   <maxtestdays>-1</maxtestdays>
   <!-- Remaining real activation days. (-1 means unlimited) -->
   <remainingrealdays>15</remainingrealdays>
   <!-- Remaining test activation days. (-1 means unlimited) -->
   <remainingtestdays>-1</remainingtestdays>
   <!-- Target information. -->
   <target>
      <!-- Processor count. -->
      <procstep>-1</procstep>
      <!-- Speed count. -->
      <speedstep>-1</speedstep>
      <!-- Software model. -->
      <softwaremodel>A100</softwaremodel>
      <!-- Billable MSU cost. -->
      <billablemsucost>100</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>-10</billablemsudelta>
   </target>
   <target>
      <!-- Processor count. -->
```

```
      <procstep>-1</procstep>
      <!-- Software model. -->
      <softwaremodel>A104</softwaremodel>
      <!-- Billable MSU cost. -->
      <billablemsucost>104</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>-6</billablemsudelta>
   </target>
   <target>
      <!-- Speed count. -->
      <speedstep>-1</speedstep>
      <!-- Software model. -->
      <softwaremodel>A105</softwaremodel>
      <!-- Real MSU cost. -->
      <billablemsucost>105</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>-5</billablemsudelta>
   </target>
   <target>
      <!-- Speed count. -->
      <speedstep>+1</speedstep>
      <!-- Software model. -->
      <softwaremodel>A115</softwaremodel>
      <!-- Billable MSU cost. -->
      <billablemsucost>115</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>5</billablemsudelta>
   </target>
   <target>
      <!-- Processor count. -->
      <procstep>+1</procstep>
      <!-- Software model. -->
      <softwaremodel>A116</softwaremodel>
      <!-- Billable MSU cost. -->
      <billablemsucost>116</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>6</billablemsudelta>
   </target>
   <target>
      <!-- Processor count. -->
      <procstep>+1</procstep>
      <!-- Speed count. -->
      <speedstep>+1</speedstep>
      <!-- Software model. -->
      <softwaremodel>A120</softwaremodel>
      <!-- Billable MSU cost. -->
      <billablemsucost>120</billablemsucost>
      <!-- Billable MSU delta -->
      <billablemsudelta>10</billablemsudelta>
   </target>
</record>
```

Engineering Change (EC)/Microcode Level (MCL) query

The output of a Get operation for the Engineering Change (EC)/Microcode Level (MCL) attribute of a Defined CPC or Console Object is an XML string. Following is an example of this output; refer to the XML schema at the end of this appendix for the complete syntax definition of this XML.

```
<!--
 This example XML document illustrates the markup used to describe
 EC/MCL information with one EC having only retrieved MCLs.
 -->
<sysinfo>
  <ec>
    <number>A12345</number>
    <partnumber>098765432</partnumber>
    <type>Console Application</type>
    <description>Console Application code</description>
    <mcl>
      <type>retrieved</type>
      <level>009</level>
      <lastupdate>2007-01-01T11:59:00Z</lastupdate>
    </mcl>
  </ec>
</sysinfo>
```

STP configuration information

The output of a Get operation for the STP Information attribute of a Defined CPC object as well as the input required for the HWMCA_SYSPLEX_TIME_SET_STP_CONFIG_COMMAND is an XML string. Following is an example of this data; refer to the XML schema at the end of this appendix for the complete syntax definition of this XML.

```
<STPConfiguration>
  <CTNID>
    <STPID>stpTst1</STPID>
  </CTNID>
  <Preferred>
    <NodeName>T25A</NodeName>
    <NodeID>
      <Type>002094</Type>
      <Model>S18</Model>
      <Manuf>IBM</Manuf>
      <PoManuf>00</PoManuf>
      <SeqNum>00000000T25A</SeqNum>
    </NodeID>
  </Preferred>
  <Backup>
    <NodeName>POLLUX</NodeName>
  </Backup>
  <Arbiter>
    <NodeID>
      <Type>002086</Type>
      <Model>A04</Model>
      <Manuf>IBM</Manuf>
      <PoManuf>00</PoManuf>
      <SeqNum>000000000T03</SeqNum>
    </NodeID>
  </Arbiter>
  <CurrentTimeServer>Backup</CurrentTimeServer>
</STPConfiguration>
```

XML schema

Following is the XML schema used to define the syntax of the XML used as input and output of the Console Application Programming Interfaces.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.ibm.com/hwmca/api"
            xmlns="http://www.ibm.com/hwmca/api"
            elementFormDefault="qualified">
<xsd:annotation>
    <xsd:documentation>
        This is the first version of the XML schema
        used to describe the XML that can be used as
        input or returned as output from the Console
        Application Programming Interfaces.  As future
        additions are made to this schema the intent is
        that it will remain compatible with the earlier
        version of the schema.
    </xsd:documentation>
</xsd:annotation>

<!--
Temporary Capacity related XML definitions.
-->
<!--
 Used to define the type for a processor.  The currently valid values for
 this element are:
    AAP - Application Assist Processor
    IFL - Integrated Facility for Linux processor
    ICF - Internal Coupling Facility processor
    IIP - Integrated Information Processors processor
    SAP - System Assist processor
    CBP - Container Based Processor
 -->
<xsd:element name="type" type="xsd:string"/>
<!--
```

```
 Used to define the identifier for a capacity record.
 -->
<xsd:element name="recordid" type="xsd:string"/>
<!--
 Used to define the number of processor steps for a specific type of
 processor compared to some base point.
 -->
<xsd:element name="procstep" type="xsd:integer"/>
<!--
 Used to define the number of processor speed steps for a specific type of
 processor compared to some base point.
 -->
<xsd:element name="speedstep" type="xsd:integer"/>
<!--
 Used to define the software model for a specific processor configuration.
 -->
<xsd:element name="softwaremodel" type="xsd:string"/>

<!--
 Used to define if a capacity request has priority.
 -->
<xsd:element name="priority" type="xsd:boolean" default="false"/>
<!--
 Used to define if a capacity request is for test or real purposes.
 -->
<xsd:element name="test" type="xsd:boolean"/>
<!--
 The "processorinfo" element define information about a processor.
 The information that can be specified with this tag includes:
    - processor type (required)
    - number of processor steps
 -->
<xsd:element name="processorinfo">
   <xsd:complexType>
      <xsd:all>
         <xsd:element ref="type"/>
         <xsd:element ref="procstep" minOccurs="0"/>
      </xsd:all>
   </xsd:complexType>
</xsd:element>
<!--
 Used to define information specific to a capacity record.
 -->
<xsd:complexType name="recordinfo">
   <xsd:sequence>
      <xsd:element ref="recordid" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="softwaremodel" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="processorinfo" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
<!--
 Used to define the parameters to use for the addition of temporary
 capacity.  The processor steps/software model specified in the element are
 specified with relation to the current configuration of the system.
 -->
<xsd:complexType name="addtype">
   <xsd:complexContent>
      <xsd:extension base="recordinfo">
         <xsd:sequence>
            <xsd:element ref="priority"/>
            <xsd:element ref="test"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name="add" type="addtype"/>
<!--
 Used to define the parameters to use for removal of temporary
 capacity.  The processor steps/software model specified in the element are
 specified with relation to the current configuration of the system.
 -->
<xsd:element name="remove" type="recordinfo"/>

<!--
 Used to define the processor information for a capacity record.
-->
<xsd:complexType name="recordprocessors">
   <xsd:all>
      <!-- Processor type. -->
      <xsd:element ref="type"/>
      <!-- Processor count. -->
      <xsd:element ref="procstep" minOccurs="0"/>
```

```
      <!-- Speed count. -->
      <xsd:element ref="speedstep" minOccurs="0"/>
      <!-- Maximum number of processors. -->
      <xsd:element name="max" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Remaining processor days. (-1 means unlimited) -->
      <xsd:element name="remainingprocdays" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Remaining MSU days. (-1 means unlimited) -->
      <xsd:element name="remainingmsudays" type="xsd:integer" minOccurs="0" maxOccurs="1"/>
   </xsd:all>
</xsd:complexType>
<!--
 Used to define information for an activation/deactivation target of a capacity record.
-->
<xsd:complexType name="target">
    <xsd:all>
       <!-- Processor count. -->
       <xsd:element ref="procstep" minOccurs="0"/>
       <!-- Speed count. -->
       <xsd:element ref="speedstep" minOccurs="0"/>
       <!-- Software model. -->
       <xsd:element ref="softwaremodel" minOccurs="1"/>
       <!-- Billable MSU cost. -->
       <xsd:element name="billablemsucost" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
       <!-- Billable MSU delta -->
       <xsd:element name="billablemsudelta" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
    </xsd:all>
</xsd:complexType>
<!--
 Used to define a capacity record.
-->
<xsd:complexType name="recordtype">
   <xsd:sequence>
      <!-- Record id -->
      <xsd:element ref="recordid" minOccurs="1" maxOccurs="1"/>
      <!-- Offering type -->
      <xsd:element name="recordtype" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <!-- Activation status -->
      <xsd:element name="status" minOccurs="1" maxOccurs="1">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="Test"/>
              <xsd:enumeration value="Real"/>
              <xsd:enumeration value="None"/>
              <xsd:enumeration value="Available"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <!-- Activation processor information. -->
      <xsd:element name="processorinfo" type="recordprocessors" minOccurs="1"
maxOccurs="unbounded"/>
      <!-- Activation start date. -->
      <xsd:element name="activationstart" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
      <!-- Activation expiration date. -->
      <xsd:element name="activationexpiration" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
      <!-- Record expiration date. -->
      <xsd:element name="recordexpiration" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
      <!-- Maximum real activation days. (-1 means unlimited) -->
      <xsd:element name="maxrealdays" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Maximum test activation days. (-1 means unlimited) -->
      <xsd:element name="maxtestdays" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Remaining real activation days. (-1 means unlimited) -->
      <xsd:element name="remainingrealdays" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Remaining test activation days. (-1 means unlimited) -->
      <xsd:element name="remainingtestdays" type="xsd:integer" minOccurs="1" maxOccurs="1"/>
      <!-- Target information. -->
      <xsd:element name="target" type="target" minOccurs="1" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
<xsd:element name="record" type="recordtype"/>

<!--
EC/MCL Information XML definitions.
-->
<!--
Used to define MicroCode Level (MCL) information.
-->
<xsd:complexType name="mcl">
  <xsd:sequence>
    <xsd:element name="type" minOccurs="1" maxOccurs="1">
       <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="retrieved"/>
```

```
            <xsd:enumeration value="installed"/>
            <xsd:enumeration value="activated"/>
            <xsd:enumeration value="accepted"/>
            <xsd:enumeration value="removed"/>
            <xsd:enumeration value="installableconcurrent"/>
            <xsd:enumeration value="removableconcurrent"/>
          </xsd:restriction>
       </xsd:simpleType>
    </xsd:element>
    <xsd:element name="level" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="lastupdate" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
Used to define pending action information.
-->
<xsd:simpleType name="actiontype">
   <xsd:restriction base="xsd:string">
       <xsd:enumeration value="ChannelConfig"/>
       <xsd:enumeration value="CouplingFacilityReactivation"/>
       <xsd:enumeration value="PoweronResetTracking"/>
   </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="actionactivation">
   <xsd:restriction base="xsd:string">
       <xsd:enumeration value="current"/>
       <xsd:enumeration value="next"/>
   </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="action">
    <xsd:attribute name="type" use="required" type="actiontype"/>
    <xsd:attribute name="activation" use="required" type="actionactivation"/>
    <xsd:attribute name="pending" type="xsd:boolean" use="required"/>
</xsd:complexType>
<xsd:complexType name="pending">
    <xsd:sequence>
        <xsd:element name="action" type="action" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<!--
Used to define Engineering Change (EC) information.
-->
<xsd:complexType name="ec">
  <xsd:sequence>
    <xsd:element name="number" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="partnumber" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="mcl" type="mcl" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<!--
Used to define Engineering Change (EC) / MicroCode Level (MCL) information.
-->
<xsd:element name="sysinfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pending" type="pending" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="ec" type="ec" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--
Program Status Word (PSW) Information XML definitions.
-->
<!--
Used to define Program Status Word (PSW) information for a single processor.
-->
<xsd:complexType name="pswinfo">
  <xsd:sequence>
    <xsd:element name="psw" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="cpid" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
Used to define CPC Image Program Status Word (PSW) information.
-->
<xsd:element name="imagepswinfo">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="pswinfo" type="pswinfo" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--
STP related XML definitions.
-->

<!--
Used to define the STP ID.
Limited to 8 alpha-numeric (plus '_' & '-') characters.
-->
<xsd:simpleType name="STPID">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="0"/>
    <xsd:maxLength value="8"/>
    <xsd:pattern value="([0-9a-zA-Z_\-])*"/>
  </xsd:restriction>
</xsd:simpleType>

<!--
Used to define the ETR ID.
Limited to 2 numeric characters. (0-31)
-->
<xsd:simpleType name="ETRID">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="31"/>
  </xsd:restriction>
</xsd:simpleType><!--
Used to define the CTN ID.

For output, if not in an STP-only or Mixed CTN, the STP ID will be empty
-->
<xsd:complexType name="CTNID">
  <xsd:all>
    <!-- Used to define the STP ID of the CTN. -->
    <xsd:element name="STPID" type="STPID" minOccurs="1" maxOccurs="1"/>
    <!-- Used to define the ETR ID of the CTN. -->
    <xsd:element name="ETRID" type="ETRID" minOccurs="0" maxOccurs="1"/>
  </xsd:all>
</xsd:complexType>

<!--
Used to define the type of processor.
Limited to 6 characters.
-->
<xsd:simpleType name="Type">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="6"/>
    <xsd:maxLength value="6"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
Used to define the model of the processor.
Limited to 3 characters.
-->
<xsd:simpleType name="Model">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="3"/>
    <xsd:maxLength value="3"/>
  </xsd:restriction>
</xsd:simpleType>

<!--
Used to define the manufacturer of the processor.
Limited to 3 characters.
-->
<xsd:simpleType name="Manuf">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="3"/>
    <xsd:maxLength value="3"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
Used to define the plant of manufacturer of the processor.
Limited to 2 characters.
-->
<xsd:simpleType name="PoManuf">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="2"/>
    <xsd:maxLength value="2"/>
```

```
  </xsd:restriction>
</xsd:simpleType>

<!--
Used to define the sequence number of the processor.
Limited to 12 characters.
-->
<xsd:simpleType name="SeqNum">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="12"/>
    <xsd:maxLength value="12"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
Used to define the Node ID of the processor.
-->
<xsd:complexType name="NodeID">
  <xsd:all>
    <!-- Used to define the type of the CPC. -->
    <xsd:element name="Type" type="Type" minOccurs="1" maxOccurs="1"/>
    <!-- Used to define the model of the CPC. -->
    <xsd:element name="Model" type="Model" minOccurs="1" maxOccurs="1"/>
    <!-- Used to define the manufacturer of the CPC. -->
    <xsd:element name="Manuf" type="Manuf" minOccurs="1" maxOccurs="1"/>
    <!-- Used to define the plant of manufacturer of the CPC. -->
    <xsd:element name="PoManuf" type="PoManuf" minOccurs="1" maxOccurs="1"/>
    <!-- Used to define the sequence number of the CPC. -->
    <xsd:element name="SeqNum" type="SeqNum" minOccurs="1" maxOccurs="1"/>
  </xsd:all>
</xsd:complexType>

<!--
Used to define the name of the processor.
Limited to 8 characters.
-->
<xsd:simpleType name="NodeName">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="8"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
Used to define the Node of the processor.

For input, there must be at least one of these specified.
If both are specified, the NodeID will take precedence.

For output, at least one is guaranteed.
-->
<xsd:complexType name="NodeDef">
  <xsd:all>
    <!-- Used to define Node ID to identify the CPC. -->
    <xsd:element name="NodeID" type="NodeID" minOccurs="0" maxOccurs="1"/>
    <!-- Used to define Node Name to identify the CPC. -->
    <xsd:element name="NodeName" type="NodeName" minOccurs="0" maxOccurs="1"/>
  </xsd:all>
</xsd:complexType>

<!--
Used to define the Current Time Server of the CTN.
Limited to "Preferred" and "Backup".
-->
<xsd:simpleType name="CurrentTimeServer">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Preferred"/>
    <xsd:enumeration value="Backup"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
Used to define STP configuration information.

For input:
Only STP-only configurations are allowed to be configured.
The CTNID is not required unless changing the CTNID of the STP-only CTN.
The Preferred node definition is required.
-->
<xsd:element name="STPConfiguration">
  <xsd:complexType>
    <xsd:all>
      <!-- Used to define new CTN ID.  -->
      <xsd:element name="CTNID" type="CTNID" minOccurs="0" maxOccurs="1"/>
      <!-- Used to define new Preferred CPC.  -->
```

```
      <xsd:element name="Preferred" type="NodeDef" minOccurs="0" maxOccurs="1"/>
      <!-- Used to define new Backup CPC.  -->
      <xsd:element name="Backup" type="NodeDef" minOccurs="0" maxOccurs="1"/>
      <!-- Used to define new Arbiter CPC.  -->
      <xsd:element name="Arbiter" type="NodeDef" minOccurs="0" maxOccurs="1"/>
      <!-- Used to define which system is going to be the Current Time Server CPC.  -->
      <xsd:element name="CurrentTimeServer" type="CurrentTimeServer" minOccurs="1"
maxOccurs="1"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

Appendix G. Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names

might be trademarks of IBM or other companies. A current list of IBM trademarks is available on http://www.ibm.com/trademark.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Class A Notices

The following Class A statements apply to this IBM product. The statement for other IBM products intended for use with this product will appear in their accompanying manuals.

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. IBM is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Industry Canada Compliance Statement

This Class A digital apparatus complies with Canadian ICES-003.

Avis de conformité à la réglementation d'Industrie Canada

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

European Community Compliance Statement

This product is in conformity with the protection requirements of EU Council Directive 2014/30/EU on the approximation of the laws of the Member States relating to electromagnetic compatibility. IBM cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of the product, including the fitting of non-IBM option cards.

This product has been tested and found to comply with the limits for Class A Information Technology Equipment according to European Standard EN 55032. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

European Community contact:
IBM Deutschland GmbH
Technical Regulations, Department M372

IBM-Allee 1, 71139 Ehningen, Germany
Tele: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Warning: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may be required to take adequate measures.

VCCI Statement - Japan

この装置は、クラス A 情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。 VCCI-A

The following is a summary of the Japanese VCCI statement above:

This is a Class A product based on the standard of the VCCI Council. If this equipment is used in a domestic environment, radio interference may occur, in which case the user may be required to take corrective actions.

Japan JIS C 61000-3-2 Compliance

（一社）電子情報技術産業協会 高調波電流抑制対策実施
要領に基づく定格入力電力値： Knowledge Centerの各製品の
仕様ページ参照

For products less than or equal to 20 A per phase, the following statement applies:

高調波電流規格 JIS C 61000-3-2 適合品

For products greater than 20 A, single-phase, the following statements apply:

高調波電流規格 JIS C 61000-3-2 準用品

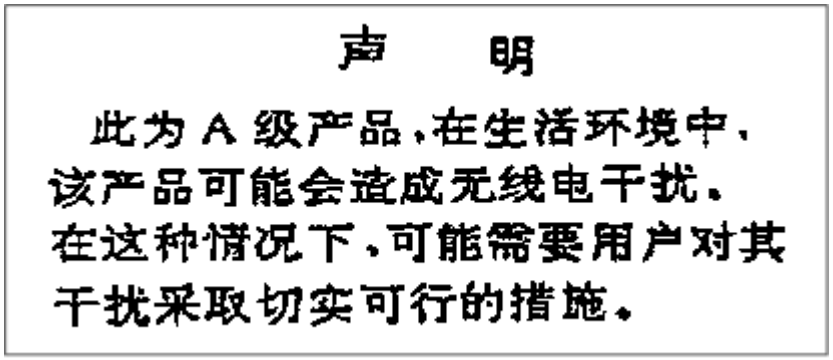
本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。
回路分類：6（単相、ＰＦＣ回路付）
換算係数：0

For products greater than 20 A per phase, three-phase, the following statements apply:

高調波電流規格 JIS C 61000-3-2 準用品

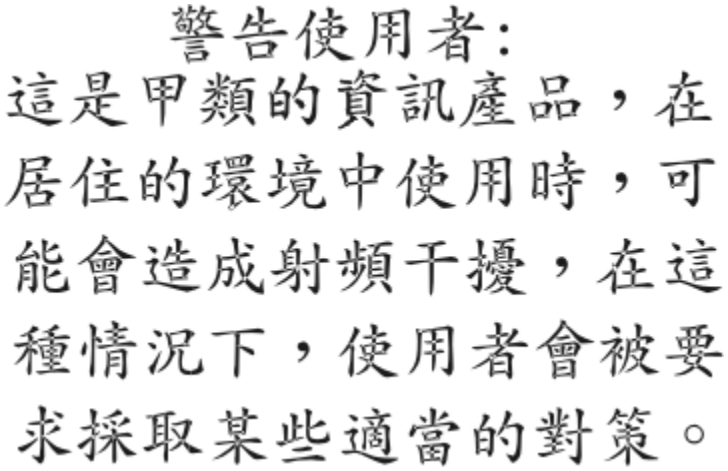
本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。
回路分類 ：5（3相、ＰＦＣ回路付）
換算係数 ：0

Electromagnetic Interference (EMI) Statement - People's Republic of China



Declaration: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may need to perform practical action.

Electromagnetic Interference (EMI) Statement - Taiwan



The following is a summary of the Taiwan EMI statement above:

Warning: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user will be required to take adequate measures.

IBM Taiwan Contact Information:

Electromagnetic Interference (EMI) Statement - Korea

이 기기는 업무용(A급)으로 전자파적합등록을 한 기기이오니
판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

Germany Compliance Statement

Deutschsprachiger EU Hinweis: Hinweis für Geräte der Klasse A EU-Richtlinie zur Elektromagnetischen Verträglichkeit

Dieses Produkt entspricht den Schutzanforderungen der EU-Richtlinie 2014/30/EU zur Angleichung der Rechtsvorschriften über die elektromagnetische Verträglichkeit in den EU-Mitgliedsstaaten und hält die Grenzwerte der EN 55032 Klasse A ein.

Um dieses sicherzustellen, sind die Geräte wie in den Handbüchern beschrieben zu installieren und zu betreiben. Des Weiteren dürfen auch nur von der IBM empfohlene Kabel angeschlossen werden. IBM übernimmt keine Verantwortung für die Einhaltung der Schutzanforderungen, wenn das Produkt ohne Zustimmung von IBM verändert bzw. wenn Erweiterungskomponenten von Fremdherstellern ohne Empfehlung von IBM gesteckt/eingebaut werden.

EN 55032 Klasse A Geräte müssen mit folgendem Warnhinweis versehen werden:

"Warnung: Dieses ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funk-Störungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen zu ergreifen und dafür aufzukommen."

Deutschland: Einhaltung des Gesetzes über die elektromagnetische Verträglichkeit von Geräten

Dieses Produkt entspricht dem "Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG)". Dies ist die Umsetzung der EU-Richtlinie 2014/30/EU in der Bundesrepublik Deutschland.

Zulassungsbescheinigung laut dem Deutschen Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG) (bzw. der EMC EG Richtlinie 2014/30/EU) für Geräte der Klasse A

Dieses Gerät ist berechtigt, in Übereinstimmung mit dem Deutschen EMVG das EG-Konformitätszeichen - CE - zu führen.

Verantwortlich für die Einhaltung der EMV Vorschriften ist der Hersteller:
International Business Machines Corp.
New Orchard Road
Armonk, New York 10504
Tel: 914-499-1900

Der verantwortliche Ansprechpartner des Herstellers in der EU ist:
IBM Deutschland GmbH
Technical Regulations, Abteilung M372
IBM-Allee 1, 71139 Ehningen, Germany
Tel: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Generelle Informationen:

Das Gerät erfüllt die Schutzanforderungen nach EN 55024 und EN 55032 Klasse A.

Electromagnetic Interference (EMI) Statement - Russia

ВНИМАНИЕ! Настоящее изделие относится к классу А.
В жилых помещениях оно может создавать радиопомехи, для снижения которых необходимы дополнительные меры

SB10-7171-06