IBM

ARTIC AIX Support User’s Guide

Version 1, Release 2, Modification 2

ARTIC AIX Support User’s Guide

Version 1, Release 2, Modification 2

Important

Before using this information and the product it supports, read the general information under Appendix G, “Notices.”

| ## Fifth Edition (December 1998)

This edition replaces and makes obsolete the previous edition. The technical changes for this edition are summarized under “Summary of Changes” and are indicated by a vertical bar to the left of a change. This edition applies to Version 1, Release 2, Modification 2 of Realtime Interface Co-Processor AIX Support and to all subsequent releases and modifications otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. You may address your comments to the following address:

IBM Corporation
Department GDN
1798 N.W. 40th Street
Boca Raton, Florida 33431
U.S.A.

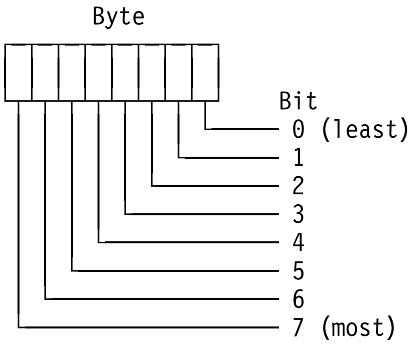
When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

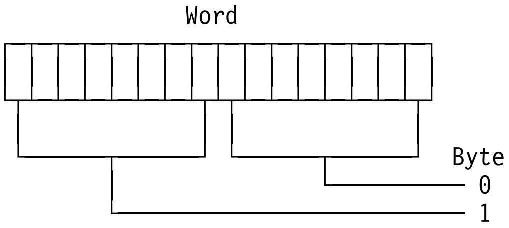
About This Book

This book describes the IBM Realtime Interface Co-Processor AIX Support. The following conventions are used in this book:

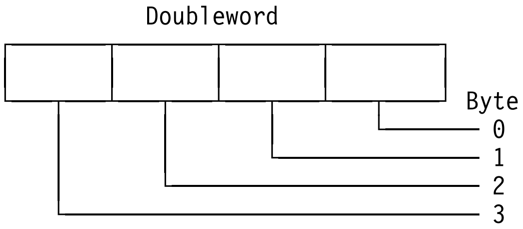
- The term *Co-Processor AIX Support* refers to the Realtime Interface Co-Processor AIX Support.
- The term *co-processor adapter* refers to an adapter card supported by Co-Processor AIX Support.
- All numbers are considered to be in decimal format unless they are immediately preceded by **0x** or immediately followed by an **h** (or **H**), in which case they are hexadecimal numbers.
- All counts in this document are assumed to start at zero.
- All bit numbering in this document conforms to the industry standard of highest order bit has the highest bit number.
- A **byte** is 8 contiguous bits and must be considered as a single object. The bits are numbered 0 to 7. Bit number 0 is the least significant bit.



- A **word** is 16 contiguous bits and must be considered as a single object. The bits are numbered 0 to 15. Bit number 0 is the least significant bit and is in byte 0, which is the low byte.



- A **doubleword** is 32 contiguous bits and must be considered as a single object. The bits are numbered 0 to 31. Bit number 0 is the least significant bit and is in byte 0, which is the low byte.



How This Book is Organized

This book is divided into the following sections:

- Chapter 1, “Overview” provides an overview of the Co-Processor AIX Support’s capabilities, hardware requirements, and software requirements.
- Chapter 2, “Installing Software” provides information needed to install and use the Realtime Interface Co-Processor AIX Support software on the system unit.
- Chapter 3, “Parameter File Description” describes the user-created parameter file, icaparm.prm.
- Chapter 4, “Device Driver Functions” describes the device driver functions and the ioctl() system functions supported by the device driver.
- Chapter 5, “Application Loader Utility” describes how the application loader loads the Realtime Control Microcode and applications to the co-processor adapter.
- Chapter 6, “Online Dump Facility” provides instructions on how to use the Online Dump Facility to obtain a dump of the co-processor adapter’s on-board memory and registers for debugging programs.
- Chapter 7, “Dump Formatter Facility” provides instructions for formatting a dump file for viewing and printing.
- Chapter 8, “C Language Interface Routines” provides a programming interface for system unit programs to the Co-Processor AIX Support and any co-processor adapter that is installed.
- Appendix A, “Output File Format for the Dump Formatter Facility” contains samples of Dump Formatter Facility output files.
- Appendix B, “Include Files” provides ioctl() function definitions, error code definitions, and ioctl() structure and union declarations. This appendix also contains function declarations and data structure definitions for the C Language Interface Library routines.
- Appendix C, “Return Codes” describes the return codes for the device driver, Application Loader Utility, and Online Dump Facility.
- Appendix D, “Messages” provides Application Loader Utility information and error messages, Online Dump Facility information and error messages, and Dump Formatter Facility error messages.
- Appendix E, “Using the Sample Programs” provides samples of a system unit program and a co-processor adapter task. It also includes instructions on compiling and linking them.
- Appendix F, “Changing the Shared Storage Window Size” describes how to change the shared storage window size using SMIT which is a part of the AIX Base Operating System.
- Appendix G, “Notices” provides book notices and trademarks.

An index is also provided.

Reference Publications

You may need to use one or more of the following IBM books that are part of the Realtime Interface Co-Processor library.

- *Realtime Interface Co-Processor C Language Support Version 1.03 (or higher) User's Guide, Volume II - Co-Processor Adapter*

  This book provides the C interface routines, and the methods of compiling and linking C tasks for the co-processor adapter.

- *Realtime Interface Co-Processor Extended Services User's Guide*

  This book:

  - Explains the installation and loading of software, event management services, intertask communications services, and asynchronous and synchronous communications support
  - Provides information necessary for Realtime Interface Co-Processor Extended Services to interface with co-processor adapters
  - Describes the functions and capabilities of Realtime Interface Co-Processor Extended Services

- *Realtime Interface Co-Processor Firmware Technical Reference*

  This book provides detailed information on the programmer interfaces to the Realtime Control Microcode for the family of Realtime Interface Co-Processor adapters. It is intended for hardware and software designers who need to understand the design and operating characteristics of the control microcode.

The following IBM books provide both hardware and software introductory and reference information, and are intended for hardware and software designers, programmers, engineers, and anyone who needs to understand the use and operation of the co-processor adapter.

- *Realtime Interface Co-Processor Multiport/2 Technical Reference*
- *X.25 Interface Co-Processor/2 Technical Reference*
- *Realtime Interface Co-Processor Portmaster Adapter/A Hardware Technical Reference*
- *Realtime Interface Co-Processor Multiport Technical Reference*
- *X.25 Interface Co-Processor Technical Reference*
- *Realtime Interface Co-Processor Multiport Model 2 Hardware Technical Reference*
- *X.25 Interface Co-Processor PCI Adapter Technical Reference*
- *ARTIC186 8-Port Adapter PCI Technical Reference*

Summary of Changes - November 1998

For this edition, support has been added for the IBM ARTIC186 8-Port PCI Adapter.

The technical changes to this edition are indicated by a vertical bar to the left of the change.

Chapter 1. Overview

This chapter provides an overview of the Realtime Interface Co-Processor AIX Support, and lists the minimum hardware and software requirements.

The Co-Processor AIX Support product is a package of program services that provides an interface between processes running under AIX (Version 4.1.x, 4.2.x, and 4.3.x, where x is any modification level) and tasks running on the co-processor adapter.

The product consists of these components:

- Device driver that allows AIX applications to interface with tasks executing on a co-processor adapter
- Application loader utility to load task files to the co-processor adapter
- Online dump facility to dump co-processor memory and hardware context
- Dump formatter facility to produce a dump report for analysis
- C Language Interface routines that provide a programming interface for system unit programs to the Co-Processor AIX Support device driver and any installed co-processor adapters
- Include file for development of AIX applications that use the device driver
- Include file for development of AIX applications that use the C Language routines to interface with the Co-Processor AIX Support device driver
- Sample programs to illustrate the device driver functions

The Co-Processor AIX Support software can be used with both IBM and non-IBM products. IBM does not exercise any control over the hardware or software of non-IBM products. The user is responsible for determining if the non-IBM products are compatible with the Co-Processor AIX Support software. IBM does not assume any responsibility for selection of any non-IBM products, nor does it provide any information on the products, or their performance, price, or maintenance.

Hardware Requirements

These are the minimum hardware requirements for the Co-Processor AIX Support product.

- One of the following co-processor adapters:
  - 4-Port Multiprotocol Communications Controller — ISA Bus, Feature Code 2701
  - Realtime Interface Co-Processor Multiport
  - X.25 Interface Co-Processor Adapter — ISA bus, Feature Code 2961
  - Realtime Interface Co-Processor Multiport Model 2
  - 4-Port Multiprotocol Communications Controller, Feature Code 2700
  - Realtime Interface Co-Processor Multiport/2, Feature Code 7002, 7004, 7022, 7024, 7026, and 7028 (depending on memory and EIB configuration)
  - X.25 Interface Co-Processor/2, Feature Code 2960
  - Realtime Interface Co-Processor Portmaster Adapter/A, Feature Code 7006, 7008, 7042, 7044, 7046, 7048 (depending on memory and EIB configuration)

- X.25 Interface Co-Processor PCI Adapter
- ARTIC186 8-Port PCI Adapter
- A system unit that supports the target ARTIC adapter and AIX.

Notes:

1. In order to compile and link tasks that will be loaded onto an ARTIC Intel 80186-based adapter, access to a Personal System/2 (PS/2) system is required to create Intel 80186 code. In lieu of the PS/2, the PC Simulator package for AIX Version 3.2 or WABI for AIX Version 4 for the system unit can be used.
2. For a complete listing of RS/6000 feature codes, refer to:

   http://wwprodsoln.bocaraton.ibm.com/artic/cfg_rs6feat.htm/

Software Requirements

The minimum software requirements for the Co-Processor AIX Support product are:

- Realtime Control Microcode, supplied with your system unit or the co-processor adapter, or downloaded from the ARTIC BBS or World Wide Web (WWW) site.

  See “Installing the AIX Support Program” on page 2-4 for the BBS and WWW site locations.
- IBM Macro Assembler/2 1.0, Microsoft C Optimizing Compiler 6.0, or Microsoft Macro Assembler 5.1, if co-processor adapter tasks are to be performed.
- Realtime Interface Co-Processor C Language Support Version 1.03 (or higher), if co-processor adapter tasks are being programmed in C language.
- For AIX Version 4, these are the additional minimum requirements:
  - IBM AIX Version 4.1.2 (or higher) for Micro Channel adapters
  - IBM AIX Version 4.1.4 (or higher) for ISA and PCI adapters
  - devices.mca.8f70.rte, Version 4.1.2.0 or higher (for Portmaster)
  - devices.mca.eff0.rte, Version 4.1.2.0 (or higher) for Multiport/2 and X.25 Co-Processor/2
  - devices.isa.mm2.rte, Version 4.1.4.0 (or higher) for Multiport Model 2
  - devices.isa.clx.rte, Version 4.1.4.0 (or higher) for Multiport and X.25 Co-Processor/1
  - devices.pci.14106100.rte for X.25 Co-Processor PCI
  - devices.pci.1410d100.rte for ARTIC186 8-Port PCI Adapter

    Note: The software package (devices.pci.14106100.rte and devices.pci.1410d100.rte) are available on the ARTIC BBS or World Wide Web site. See “Installing the AIX Support Program” on page 2-4 for the BBS and WWW site locations.
  - C for AIX or CSet++ for AIX to develop system unit applications that interface with the Realtime Interface Co-Processor AIX Support device driver.

Chapter 2. Installing Software

This chapter provides information you need to install and use the Co-Processor AIX Support software on the system unit. Review this chapter completely before installing this product, especially the information under “Configuring the AIX Support Program” on page 2-4.

Note: Before proceeding with the software installation, you need your system configured with the minimum system requirements that are listed under “Hardware Requirements” on page 1-1 and “Software Requirements” on page 1-2.

Product Contents

The Co-Processor AIX Support product files are contained on one diskette in IBM AIX **installp** format. After installation, the product files will reside in the directories listed in this section.

The following files are the device driver and the configuration methods used during driver configuration. They are loaded into the standard AIX directories **/usr/lib/drivers** and **/usr/lib/methods**.

The following files are message catalogs loaded into the directory specified by the NLSPATH environment variable:

ric_supp.cat
ric_c3x.cat
ric_stf.cat

Configuration methods:

cfgartic
ucfgartic
defartic
udefartic
chgartic

All other product files are loaded into directory **/usr/lpp/devices.artic**. The directory **/usr/lpp** is the standard AIX directory for installing Licensed Program Products (**lpp**). The ARTIC directory and all other directories within ARTIC are created by the installation procedure.

Note

mpx is the device name for the following adapters:

- ISA
  - Realtime Interface Co-Processor Multiport
  - X.25 Interface Co-Processor
- PCI
  - ARTIC186 8-Port PCI Adapter
  - X.25 Interface Co-Processor PCI Adapter
- Micro Channel
  - Realtime Interface Co-Processor Multiport/2
  - X.25 Interface Co-Processor/2

pm is the device name for the following adapters:

- ISA
  Multiport Model 2
- Micro Channel
  Realtime Interface Co-Processor Portmaster Adapter/A

The following files are information files:

lpp.doc	Product information file
copyright.master	IBM copyright notice
devices.artic.adt.mpx.odmadd	ODM database add file for mpx
devices.artic.adt.pm.odmadd	ODM database add file for pm devices
devices.artic.adt.sm_ricsupmpx.odmadd	ODM database SMIT add files for mpx devices
devices.artic.adt.sm_ricsuppm.admadd	ODM database SMIT add files for pm devices
devices.artic.adt.undo.err	Error reporting undo file
devices.artic.adt.undo.trc	Performance tracing undo file

The following files are in the **bin** subdirectory:

icaldric	Application program loader
icadpric	Online dump facility
frmtdump	Dump formatter facility

The sample programs consist of an AIX program and a co-processor adapter task. These are loaded into the directories **sample/user** and **sample/task.** The following files are in the **sample/user** directory:

readme	Sample system unit program information
icadata.h	Header file for sample programs
suioctl.c	AIX sample source code

suclib.c	Source for the sample program that uses the C Language interface routines
suclib	Executable file for the sample program that uses C Language interface routines
icadisp.c	Source file for the display sample program
makeall	Script to create samples
makefile	Makefile for compiling and linking samples

The following files for the co-processor adapter task are in the **sample/task** directory:

readme	Sample task information
ricsamp.c	Sample task source
ricsamp.exe	Sample task executable program
ricsamp.map	Task load map
build.bat	Batch program to build **ricsamp.exe**

Note: In order to compile and link the co-processor adapter task, you will need access to a system that supports Microsoft C to create Intel 80186 code. In lieu of an Intel-based operation system, use WABI for AIX Version 4.

The directories **driver** and **odm** contain files that are not for direct use by the user. They are support programs used by the adapter configuration program and should be ignored.

The directory **include** contains the header files that must be included in AIX programs that use the Co-Processor AIX Support device driver. The files in **include** are:

icaclib.h	Include file for programs that interface with the Co-Processor AIX Support device driver through the C Language Interface routines
icaioctl.h	Include file for programs that use the Co-Processor AIX Support device driver

The directory **sys** contains program message files and the application loader parameter file. The files in **sys** are:

icaldric.msg	Application loader message file
icaparm.prm	Application loader parameter file
icadpric.msg	Online dump facility message file
frmtdump.msg	Dump formatter facility message file
frmtdump.pro	Dump formatter facility profile
artic.trc	Trace facility template
ricerrlog	Error log template
ric_supp.msg	Device driver message file

The file in **lib** is:

libric.a	C Language Interface library routines

Installation Procedure

The installation procedure consists of the following steps:

- Installing the Co-Processor AIX Support software
- Configuring the Co-Processor AIX Support software
- Installing and loading the Realtime Control Microcode

Installing the AIX Support Program

The procedure for creating and installing the Co-Processor AIX Support Software is included as part of the AIX Support package on the BBS. The package can also be downloaded from the World Wide Web. Go to either repository for a copy of installation and configuration information.

1. Download the appropriate AIX package from either of the following locations:
   - BBS located in the USA — dial **(561) 443-0134**
   - WWW using URL — **http://wwprodsoln.bocaraton.ibm.com/artic/file_rep.html**
2. Follow the installation instructions supplied in the package.

See “Configuring the AIX Support Program” to configure your support software.

Configuring the AIX Support Program

Co-Processor AIX Support software can be configured to use any of the following co-processor adapters if they are installed in your system unit:

- X.25 Interface Co-Processor PCI Adapter
- ARTIC186 8-Port PCI Adapter
- Realtime Interface Co-Processor Portmaster Adapter/A
- X.25 Interface Co-Processor/2
- Realtime Interface Co-Processor Multiport/2
- Multiport Model 2
- X.25 Interface Co-Processor
- Realtime Interface Co-Processor Multiport

Different procedures are used to configure the Co-Processor AIX Support software. The appropriate procedure to use depends on the level of AIX that is installed on your system. For AIX Version 4, follow the configuration information which came with the AIX Support Program.

To configure the AIX support program:

1. Login as root, if you have not already done so.
2. Issue the command: **smitty**
3. From the SMIT menu, select **Devices**.
4. From the SMIT menu, select **Communication**.
5. From the SMIT menu, select the appropriate device.
   - Portmaster Adapter/A
   - Multiport Model 2 Adapter
   - X.25 Co-Processor/1 Adapter (for X.25 Co-Processor and Multiport)
   - X.25 Co-Processor/2 or Multiport/2 Adapter
   - X.25 Co-Processor PCI Adapter

- IBM ARTIC186 8-Port PCI Adapter

6. From the SMIT menu, select **Adapter.**

   If you have a Micro Channel or PCI adapter, go to step 8.

7. To add an adapter for an ISA card, do the following:

   a. From the SMIT menu, select the appropriate option:
      - Add an X.25 Adapter (for X.25 Co-Processor and Multiport)
      - Add a Multiport Model 2 Adapter

   b. Select the parent bus the adapter is plugged into.

   c. From the next SMIT menu, provide the following information:
      - ISA Interrupt Level = 7
      - Bus IO Address = 0x02A0
      - Bus Memory Address = 0xE0000

      Note: The values selected for ISA Interrupt Level and Bus IO Address must match the values indicated on the switch settings of the ARTIC adapter. For more information, refer to the *Guide to Operations* provided with your ARTIC adapter, which is also available on the World Wide Web at URL:

      http://wwprodsoln.bocaraton.ibm.com/artic/pubs.html

   d. Press Enter to complete adding the adapter.

   e. Press **F3** to return to the Adapter SMIT menu.

8. From the SMIT menu, select the appropriate option:

   - Manage Device Drivers for X.25 Co-Processor/2 or Multiport/2 Adapters (for X.25 Co-Processor and Multiport and X.25 Co-Processor/2 or Multiport/2).
   - Manage Device Drivers for X.25 Co-Processor PCI Adapters
   - Manage Device Drivers for IBM ARTIC186 8-Port PCI Adapters
   - Manage Device Drivers for Multiport Model 2 Adapters
   - Manage Device Drivers for Portmaster Adapter/A

9. From the SMIT menu, select **Manage RIC AIX Support Device Driver**.

10. From the next SMIT menu, select **Add a Device Driver**.

    A list of adapters is displayed.

11. Highlight the adapter you want to configure and press Enter. The following message denotes that the selected adapter was configured:

    ```
    articX Available
    ```

    where X is the logical card number (also known as the minor number).

    Note: The default shared storage window size for all supported adapters is 8K. In AIX Support Version 1.1.2 and earlier, it was 32K. To change the shared storage window size for the micro channel adapters, refer to "Appendix F. Changing the Shared Storage Window Size" in the *AIX Support User's Guide*.

12. Press **F10** to exit SMIT.

Configuration Note: After your Co-Processor AIX Support software is installed and configured, and the Realtime Control Microcode is installed and loaded, the Application Loader Utility can be used to load user tasks onto the co-processor adapter. For information on loading application tasks, see Chapter 5, "Application Loader Utility."

Installing and Loading the Realtime Control Microcode

The IBM Realtime Control Microcode (RCM) is the control program for the co-processor adapter and must be loaded on the co-processor adapter before any user tasks are loaded. The Realtime Control Microcode's file name is either **icaaim.com** (for the X.25 Interface Co-Processor/2, X.25 Interface Co-Processor PCI, X.25 Interface Co-Processor, Realtime Interface Co-Processor Multiport/2, or Realtime Interface Co-Processor Multiport, and ARTIC186 8-Port PCI Adapter) or **icarcm.com** (for the Realtime Interface Co-Processor Portmaster Adapter/A or Multiport Model 2).

RCM can be included with your adapter or it can be downloaded from an ARTIC repository. Refer to the instructions in “Installing the AIX Support Program” on page 2-4.

The application loader utility (**icaldric**) is used to load the Realtime Control Microcode onto the co-processor adapter. For more information on the application loader, see Chapter 5, “Application Loader Utility.”

Loading the Realtime Control Microcode

For the Realtime Interface Co-Processor Multiport/2, the Realtime Interface Co-Processor Multiport, X.25 Interface Co-Processor/2, X.25 Interface Co-Processor, X.25 Interface Co-Processor PCI adapter, and ARTIC 186 8-Port PCI Adapter, use the following command to load the Realtime Control Microcode onto the co-processor adapter (for AIX Version 4):

```
icaldric n /usr/lib/microcode/icaaim.com 0 reset
```

For the Realtime Interface Co-Processor Portmaster Adapter/A and Multiport Model 2, use the following command to load the Realtime Control Microcode onto the co-processor adapter (for AIX Version 4):

```
icaldric n /usr/lib/microcode/icarcm.com 0 reset
```

The logical co-processor adapter number **(n)** is determined by AIX. The AIX configuration manager scans the physical slots from low to high and defines the consecutive logical card numbers starting at zero (0) for each supported card found. If a supported adapter is added to a slot either before or after an already defined supported adapter, it is assigned the next consecutive logical number.

The **0** in the command line is the task number. Realtime Control Microcode must *always* be loaded as task 0. The parameter **-reset** on the command line causes a reset of the indicated co-processor adapter prior to loading task 0 (Realtime Control Microcode). If the Realtime Control Microcode is successfully loaded, the following message is displayed:

```
Normal termination:  Task 0 loaded on coproc n.
```

If a different message is displayed, see “Application Loader Utility Return Codes” in Appendix C, “Return Codes” for a description of the problem and the action to take to correct the problem.

Chapter 3. Parameter File Description

This chapter provides a detailed description of the user-created parameter file **icaparm.prm**. The parameter file is not required but can be created to override the MAXTASK, MAXPRI, MAXQUEUE, and MAXTIME configuration parameters for the co-processor adapter. If there is no parameter file, the default configuration parameters described under "Default System Parameters" on page 3-4 are used.

Note: The parameter file can be called by any AIX file name. The name **icaparm.prm** is used as an example name in this book.

Parameter File Definition

The **icaparm.prm** file is a user-created ASCII file that defines parameters for the co-processor adapters installed in your system unit. The file consists of one record (line) for each co-processor adapter.

These are the general rules to consider when defining the **icaparm.prm** file.

- The **icaparm.prm** file is created with a text editor as an ASCII file.
- The file's pathname is specified on the application loader command line by the **-pf <pathname>** option.

  If the **-pf** option is not used, or if the specified file cannot be found or read, the default system parameter file located in the ARTIC subdirectory **sys/icaparm.prm** is used. If neither the specified file nor the system default file can be used, the default parameters are used, as listed under "Default System Parameters" on page 3-4.
- The **icaparm.prm** file should contain one record (line) for each co-processor adapter installed on the system. Multiple co-processor adapters can be installed in a system. The *logical* co-processor adapter numbers begin at 0 and are given in the co-processor adapter record.

  If there are more co-processor adapters installed in the system than there are records in the **icaparm.prm** file, those co-processor adapters will receive the default configuration parameters described under "Default System Parameters" on page 3-4.
- Each record (line) for defining a co-processor adapter must start with a # (number sign).
- Records that do not start with # are treated as comments.
- Each record must end with a **;** (semicolon) except for the last record which must end with a $ (dollar sign).
- White space (spaces and/or tabs) separate parameter entries in a record line.
- All numbers are assumed to be hexadecimal.

- The **icaparm.prm** file must follow the format described in the following table.

Field	Name	Value/Range
0	Beginning record delimiter	#
1	Co-processor adapter number	0x00 - 0x07
2	MAXTASK	0x00 - 0xFD
3	MAXPRI	0x01 - 0xFF
4	MAXQUEUE	0x00 - 0xFE
5	MAXTIME	0x00 - 0xFE
6	Ending record delimiter	$ (last entry)
6	Ending record delimiter	; (not last)

Parameter File Examples

The following example shows the parameter file for a system unit with two co-processor adapters installed.

```
This file defines two co-processor
adapters.
# 0 11 12 13 14 ;
# 1 16 17 18 19 $
```

In the preceding example, the first two lines (a sentence) do not begin with the # character so they are considered comment lines and ignored. The third line defines parameters for adapter number 0, where 0x11 is MAXTASK, 0x12 is MAXPRI, 0x13 is MAXQUEUE, 0x14 is MAXTIME. The fourth line defines the same parameters for the second adapter, number 1.

The following parameter records are all examples of invalid parameter file entries:

00 11 12 13 14 ;	Does not begin with #; record ignored–no error
# 00 11 12 13 14	Does not end with ; or $
# 00 11 12 13 14 15 ;	Too many parameters
# 00 11 12 13 ;	Too few parameters
# ff 11 12 13 14 ;	Card number out of range
# 0 123 12 13 14 ;	MAXTASK out of range
# 0 11 -12 13 14 $	MAXPRI out of range
# 01 16 17 18 19 ;	Record after $ end-of-file

Parameter Field Definitions

This section defines each parameter field of the **icaparm.prm** file.

Beginning record delimiter

Range:	# (number sign)
Description:	The first character of each record must be a # or the record is treated as a comment and ignored.

Co-processor adapter number

Range:	0x00 - 0x07
Description:	This is the number of the co-processor adapter to which the parameter fields of this record are to be applied.

MAXTASK

Range:	0x00 - 0xF8
Description:	This is the highest task number that can be loaded on a given co-processor adapter. Task 0 is reserved for the Realtime Control Microcode. This value should be selected carefully to avoid reserving unneeded space in the Realtime Control Microcode's data area.

MAXPRI

Range:	0x01 - 0xFF
Description:	This is the highest priority level that may be assigned to a task loaded on this co-processor adapter. This value should be selected carefully to avoid reserving unneeded space in the Realtime Control Microcode’s data area.

MAXQUEUE

Range:	0x00 - 0xFE
Description:	This is the highest queue number available for the application tasks executing on the co-processor adapter. This value should be selected carefully to avoid reserving unneeded space in the Realtime Control Microcode’s data area.

MAXTIME

Range:	0x00 - 0xFE
Description:	This is the highest timer number reserved for application tasks executing on the given co-processor adapter. This value should be selected carefully to avoid reserving unneeded space in the Realtime Control Microcode’s data area.

Ending record delimiter

Range:	; or $
Description:	The last character in the definition record for a co-processor adapter should be a **;** (semicolon). If this is the last record in the file, the last character should be a $ (dollar sign). If a system does not have the $ character, the ASCII equivalent of $ (0x24) should be used.

Default System Parameters

The default system parameters are:

MAXTASK Set to a default value of 0x10 (decimal 16)
MAXPRI Set to a default value of 0x10 (decimal 16)
MAXQUEUE Set to a default value of 0x50 (decimal 80)
MAXTIME Set to a default value of 0x32 (decimal 50)

If these defaults do not meet the requirements of your system configuration, create an entry in the parameter file to override the defaults.

Chapter 4. Device Driver Functions

This chapter describes the Co-Processor AIX Support device driver functions.

Overview

| The Co-Processor AIX Support device driver:

- Handles interrupts from the co-processor adapters.
- Determines the task and co-processor adapter that interrupted the system, and signals the appropriate system unit processes.
- Provides a programming interface to the co-processor through the ioctl() device driver interface.
- Serializes calls made to the programming interface from multiple applications.

The Co-Processor AIX Support device driver supports a subset of the standard character device driver function calls defined by AIX:

open()
close()
select()
ioctl()

The Co-Processor AIX Support device driver supports the following interface calls to the co-processor adapter through the Co-Processor AIX Support device driver call **ioctl**:

Reset
Read Memory
Write Memory
Interrupt Register
Interrupt Wait
Interrupt Deregister
Issue Command
Get Parameters
Get Buffer Addresses
Get Version
Get Primary Status
Special Events Register
Special Events Wait
Special Events Deregister

Examples of these calls are in the system unit sample program **suioctl.c**, which is on your Co-Processor AIX Support program diskette. Instructions on how to run the sample programs are in Appendix E, “Using the Sample Programs.”

The C Language interface routines, described in Chapter 8, “C Language Interface Routines,” provide a higher-level programming interface to the Co-Processor AIX Support device driver and the co-processor adapter.

open()

The device driver is opened with the character device special file **/dev/artic**. This single file is used to open the device driver regardless of which co-processor adapter is to be accessed. A single call to the open() function allows an application to access all of the co-processor adapters recognized by the device driver.

The following is an example of a call to the open() routine that opens the device special file **/dev/artic**.

```
#include <fcntl.h>
#include <icaioctl.h>
#include <sys/errno.h>
        .
        .

extern int errno;

int fd;              /* File descriptor for /dev/artic   */
        .
        .
    if ((fd = open("/dev/artic", O_RDONLY)) == -1) {
        printf("open on /dev/artic failed, errno = %d\n",
                  errno);
        .
        .
    } else {
        .
        .
```

The device driver may be opened only once by each user process. The driver will return the code 0xFF26 (E_ICA_ALREADY_OPEN) in errno if an attempt is made to open the driver a second time.

close()

The device driver character device special file **/dev/artic** is closed with the close() function call.

The following is an example of a call to the close() function that closes the device special file **/dev/artic**:

```
#include <fcntl.h>
#include <icaioctl.h>
#include <sys/errno.h>
        .
        .
extern int errno;
int fd;               /* File descriptor for /dev/artic   */
        .
        .
    if ((fd = close(fd)) == -1) {
        printf("close on /dev/artic failed, errno = %d\n",
                  errno);
        .
        .
    } else {
        .
        .
```

select()

The Co-Processor AIX Support device driver supports the select() system call in the following manner:

- Read selects may never be satisfied and hang indefinitely unless a timeout value is specified.
- Write selects may never be satisfied and hang indefinitely unless a timeout value is specified.
- Exception selects are satisfied when a Co-Processor AIX Support task interrupts the system unit or the Realtime Control Microcode receives an initialize command.

Following is an example of call to the select() function that waits for a Co-Processor AIX Support task interrupt for five seconds:

```
#include <fcntl.h>
#include <icaioctl.h>
#include <sys/errno.h>
#include <sys/select.h>
#include <sys/types.h>
#include <sys/time.h>

#define MAX_FDVAL 48        /* Max descriptor value to query */

typedef struct                       /* File descriptor mask */
{
    int fdsmask[(MAX_FDVAL/32)+1];
} exceptlist;

int fd;                    /* File descriptor for /dev/artic */
int rc;                    /* Return code from select        */
struct timeval             /* Specify 5 second timeout       */
        tmout = {5L, 0L};
extern int errno;
        .
        .
    exceptlist.fdsmask[fd/32] |= (1<<(fd%32));
     if ((rc = select(MAX_FDVAL, NULL, NULL, &exceptlist, &tmout))=-1) {
         printf("select on /dev/artic failed, errno = %d\n",
                   errno);
         .
         .
    } else {
        .
        .
```

ioctl()

The Co-Processor AIX Support device driver functions are provided with the ioctl() system function call. The following pages describe the services provided, their input and output parameters, and the possible error codes that can be returned by the device driver.

See “Device Driver Return Codes” on page C-1 for a list of the error codes returned by the device driver.

The following is an example of an ioctl() call with function code ICAGETBUFADDRS (Get Buffer Addresses).

```
#include <fcntl.h>
#include <icaioctl.h> /* ioctl codes and parameter structures */
#include <sys/errno.h>

int fd;                     /* File descriptor for /dev/artic */
ARTIC_IOCTL_PARMS arg;      /* Parameter buffer for ioctl()   */
int rc;                     /* Return code from ioctl         */
        .
        .                   /* Get the buffer addresses       */
        .                   /* of task 7 on card 0            */
     arg.icagetbufaddrs.coprocnum = 0;
     arg.icagetbufaddrs.tasknum = 7;
    if ((rc = ioctl(fd,ICAGETBUFADDRS,(char *)&arg)) == -1)
    {
         printf("ioctl to /dev/artic failed, errno = %d\n",
                errno);
        .
        .
    } else {
        if (arg.icagetbufaddrs.retcode != 0) {
            printf("artic driver error, ");
            printf("driver returns %x\n",
                    arg.icagetbufaddrs.retcode);
        .
        .
```

Reset

Purpose Issues a hardware reset to the co-processor adapter.

Invocation ICARESET

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 ushort   retcode; /* Return code                 */
                 ulong   reserved; /* Reserved field              */
               } ICARESET_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter to be reset

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0B	E_ICA_TIMEOUT
0xFF33	E_ICA_RESET_FAILED

Remarks The Reset function issues a hardware reset to the co-processor adapter. The Realtime Control Microcode and all other tasks are unloaded and the co-processor adapter performs a power-on self test (POST).

The ICARESET function performs the same function as the C Language interface routine **icareset**.

A reset of the co-processor adapter may also be performed when loading the Realtime Control Microcode to the co-processor adapter by using the **-reset** application loader option. See page 5-3 for additional information on the **-reset** option.

Related Topics None

Read Memory

Purpose Reads from a co-processor adapter's memory into an application buffer.

Invocation ICAREADMEM

Format

```
typedef struct
              {
                uchar   coprocnum;  /* Co-processor adapter number */
                ulong      length;  /* Length                      */
                ushort    segpage;  /* Segment/Page                */
                ushort     offset;  /* Offset                      */
                uchar       *dest;  /* Destination buffer pointer  */
                uchar addr_format;  /* Address format              */
                ushort    retcode;  /* Return code                 */
                ulong    reserved;  /* Reserved field              */
              }ICAREADMEM_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

length The number of bytes to be read from the co-processor adapter memory.

segpage The segment or page of the co-processor adapter memory address.

offset The offset of the co-processor adapter memory address. The interpretation of this field is determined by the **addr_format** field.

dest A pointer to the application buffer where the adapter memory is to be copied.

addr_format The control field determining the address format.

Value	**Address interpretation**
0x00	The **segpage** parameter is a segment in co-processor adapter memory, and the **offset** is an offset within that segment.
0xFF	The **segpage** parameter is a page in co-processor adapter memory, and the **offset** is an offset within that page.
0x01	The **segpage** and **offset** parameters are a 32-bit physical address in adapter memory. The least significant 16-bits are in the **offset** field.

Note: All addressing formats are converted to page:offset formats by the Co-Processor AIX Support device driver prior to accessing co-processor adapter memory. When accessing the upper 1 MB on a 2 MB Realtime Interface Co-Processor Portmaster Adapter/A, page:offset format must be used because the segment:offset format can only refer to addresses in the 0–1 MB range.

Returns

retcode The return code set by the device driver:

0x0000	NO_ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF07	E_ICA_INVALID_PAGE
0xFF08	E_ICA_INVALID_OFFSET
0xFF09	E_ICA_INVALID_FORMAT
0xFF2C	E_ICA_NOMEM

Remarks The Read Memory function reads from the co-processor adapter memory into a system unit application buffer. The address in co-processor adapter memory can be specified either as a segment and offset or as a page and offset. It is the responsibility of the application to recognize that any data read from the co-processor adapter memory with this call must be in Intel (byte-swapped) format.

The ICAREADMEM function performs the same function as the C Language interface routine **icareadmem**.

Related Topics Write Memory

Write Memory

Purpose Writes to a co-processor adapter's memory from an application buffer.

Invocation ICAWRITEMEM

Format

```
typedef struct
               {
                 uchar   coprocnum; /* Co-processor adapter number */
                 ulong      length; /* Length                      */
                 ushort    segpage; /* Segment/Page                */
                 ushort     offset; /* Offset                      */
                 uchar     *source; /* Source buffer pointer       */
                 uchar addr_format; /* Address format              */
                 ushort    retcode; /* Return code                 */
                 ulong    reserved; /* Reserved field              */
               } ICAWRITEMEM_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

length The number of bytes to be written to the co-processor adapter memory.

segpage The segment or page of the co-processor adapter memory address.

offset The offset of the co-processor adapter memory address. The interpretation of this field is determined by the **addr_format** field.

source A pointer to the application buffer that contains the data to be written to adapter memory.

addr_format The control field determining the address format.

Value:	**Address interpretation:**
0x00	The **segpage** parameter is a segment in co-processor adapter memory, and the **offset** is an offset within that segment.
0xFF	The **segpage** parameter is a page in co-processor adapter memory, and the **offset** is an offset within that page.
0x01	The **segpage** and **offset** parameters are a 32-bit physical address in adapter memory. The least significant 16-bits are in the **offset** field.

Note: All addressing formats are converted to page:offset formats by the Co-Processor AIX Support device driver prior to accessing co-processor adapter memory. When accessing the upper 1 MB on a 2 MB Realtime Interface Co-Processor Portmaster Adapter/A, page:offset format must be used because the segment:offset format can refer only to addresses in the 0–1 MB range.

Returns

retcode The return code set by the device driver:

0x0000	NO_ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF07	E_ICA_INVALID_PAGE
0xFF08	E_ICA_INVALID_OFFSET
0xFF09	E_ICA_INVALID_FORMAT
0xFF2C	E_ICA_NOMEM

Remarks The Write Memory function writes to the co-processor adapter memory from a system unit application buffer. The address in adapter memory can be specified either as a segment and offset or as a page and offset. It is the responsibility of the application to recognize that any data written to the co-processor adapter memory with the Write Memory call must be in Intel (byte-swapped) format.

The ICAWRITEMEM function performs the same function as the C Language interface routine **icawritemem**.

Related Topics Read Memory

Interrupt Register

Purpose Registers an application process with the device driver for notification of a specific task interrupt.

Invocation ICAINTREG

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 uchar    tasknum; /* Task number                 */
                 ushort   retcode; /* Return code                 */
                 ulong   reserved; /* Reserved field              */
               } ICAINTREG_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF15	E_ICA_ALREADY_REG
0xFF25	E_ICA_XMALLOC_FAIL
0xFF27	E_ICA_INVALID_TASK
0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks The Interrupt Register function allows applications to be notified of task interrupts with the Interrupt Wait function. An application must first register using this function before being notified of task interrupts. Note that for the Interrupt Register function, the error task (0xFE) is always a valid task number and will not result in the E_ICA_INVALID_TASK error code being returned.

The ICAINTREG function performs the same function as the C Language interface routine **icaintreg**.

Related Topics Interrupt Deregister, Interrupt Wait

Interrupt Wait

Purpose Blocks an application process until a specific task on a co-processor adapter interrupts the system unit.

Invocation ICAINTWAIT

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 uchar    tasknum; /* Task number                */
                 ulong    timeout; /* Timeout                    */
                 ushort   retcode; /* Return code                */
                 ulong   reserved; /* Reserved field             */
               } ICAINTWAIT_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

timeout The time in milliseconds to wait for a task interrupt. If it is 0, the call returns immediately indicating whether or not a previous task interrupt has occurred.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0B	E_ICA_TIMEOUT
0xFF17	E_ICA_NOT_REG
0xFF27	E_ICA_INVALID_TASK
0xFF28	E_ICA_INTR
| 0xFF30	E_ICA_NO_MORE_RES
| 0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks The Interrupt Wait function returns immediately with no error if an application previously registered for that task interrupt and the interrupt occurred prior to this call. If the interrupt did not occur previously, this call blocks the application process until the specified task interrupts or the time specified in the **timeout** field has expired. If multiple interrupts by the same task occur prior to the Interrupt Wait call, the application is only notified once. An application must first register with the Interrupt Register before being notified of task interrupts. Note that for the Interrupt Wait function, the error task (0xFE) is always a valid task number and will not result in the E_ICA_INVALID_TASK error code being returned.

The ICAINTWAIT call performs the same function as the Co-Processor AIX Support **icaintwait** function.

Related Topics Interrupt Register

Interrupt Deregister

Purpose Cancels application process request to be notified of a specific task interrupt.

Invocation ICAINTDEREG

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 uchar    tasknum; /* Task number                 */
                 ushort   retcode; /* Return code                 */
                 ulong   reserved; /* Reserved field              */
               } ICAINTDEREG_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF17	E_ICA_NOT_REG
0xFF27	E_ICA_INVALID_TASK
0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks The Interrupt Deregister function cancels a previous application request to be notified of a specific task interrupt. Processes should cancel all requests for task interrupt notification prior to terminating.

The ICAINTDEREG function performs the same function as the C Language interface routine **icaintdereg**.

Related Topics Interrupt Register, Interrupt Wait

Issue Command

Purpose Issues a command to a task with an option to copy parameter data from an application buffer to the task's output buffer before issuing the command.

Invocation ICAISSUECMD

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 uchar    tasknum; /* Task number                 */
                 uchar    cmdcode; /* Command Code                */
                 ushort    length; /* Length of parameter buffer  */
                 ulong    timeout; /* Timeout                     */
                 uchar      *prms; /* Pointer to parameters       */
                 ushort   retcode; /* Return code                 */
                 ulong   reserved; /* Reserved field              */
               } ICAISSUECMD_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

cmdcode The command code to put in the task's Buffer Control Block (BCB).

length The number of bytes to be copied to the task's output buffer. A value of zero indicates that nothing should be written.

timeout The number of milliseconds to wait for the Realtime Control Microcode to respond to the command.

prms A pointer to the application buffer containing the data to be written to the task's output buffer. The **prms** field is ignored if the **length** field is zero.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS[1]
0xFF0B	E_ICA_TIMEOUT
0xFF11	E_ICA_BAD_PCSELECT
0xFF12	E_ICA_CMD_REJECTED
0xFF14	E_ICA_OB_SIZE
0xFF27	E_ICA_INVALID_TASK
0xFF30	E_ICA_NO_MORE_RES
0xFF31	E_ICA_BAD_OPEN_HANDLE

[1] A primary status of 0x00 is allowed if the Realtime Control Microcode is not yet loaded so applications can send commands to the ROS.

Remarks The Issue Command function issues a command to a task. The caller has the option of copying parameter information into the task's output buffer before the command is issued. It is the responsibility of the application to ensure that any parameter data to be copied to the task's output buffer is in Intel (byte-swapped) format.

The ICAISSUECMD function performs the same function as the C Language interface routine **icaissuecmd**.

Related Topics None

Get Parameters

Purpose Obtains the configuration parameter information for a co-processor adapter.

Invocation ICAGETPARMS

Format

```
typedef struct
               {
                 uchar   coprocnum;    /* Co-processor adapter number */
                 ICAPARMS cfgparms;    /* Configuration parameters    */
                 ushort    retcode;    /* Return code                 */
                 ulong    reserved;    /* Reserved field              */
               } ICAGETPARMS_PARMS;

typedef struct
               {
                ushort  io_addr;      /* Address of I/O ports        */
                uchar   maxtask;      /* Maximum task number         */
                uchar    maxpri;      /* Maximum task priorities     */
                uchar  maxqueue;      /* Maximum queues              */
                uchar   maxtime;      /* Maximum timers              */
                uchar int_level;      /* Adapter interrupt level     */
                uchar  ssw_size;      /* Shared storage window size  */
           } ICAPARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

Returns

cfgparms Structure set by the device driver with the configuration parameters for the specified co-processor adapter.

io_addr The base I/O address of the co-processor adapter's I/O ports. These ports are used by the device driver for controlling the co-processor adapter.

maxtask The highest task number that can be loaded on the co-processor adapter.

maxpri The highest value of a task's priority. The highest priority level is 1, whereas the lowest priority level has the maximum value.

maxqueue The highest queue number that can be allocated on the co-processor adapter.

maxtime The highest timer number that can be allocated on the co-processor adapter.

int_level The interrupt level on which the co-processor adapter interacts with the system unit.

ssw_size The shared storage window size. It is a code that indicates the size of the co-processor adapter's shared storage window. The following table indicates what size window each value represents:

Size Code	**Window Size (in KB)**
0	8 (default)
1	16
2	32
3	64

Notes:

1. The default changed from 32 KB to 8 KB in Version 1.1.3.
2. For information on changing the window size, refer to Appendix F, "Changing the Shared Storage Window Size."

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC

Remarks Some of the parameters returned by the function (**maxtask**, **maxpri**, **maxtime**, and **maxqueue**) can be defined in the parameter file (**icaparm.prm**), described in Chapter 3, "Parameter File Description." The device driver uses the parameters or defaults when loading the Realtime Control Microcode on a co-processor adapter.

The ICAGETPARMS function performs the same function as the C Language interface routine **icagetparms**.

Related Topics None

Get Buffer Addresses

Purpose Gets the address and length of a task’s input, output, and secondary status buffers.

Invocation ICAGETBUFADDRS

Format

```
typedef struct
               {
                 uchar   coprocnum;      /* Co-processor adapter number    */
                 uchar     tasknum;      /* Task number                     */
                 ICABUFFER      ib;      /* Input buffer information        */
                 ICABUFFER      ob;      /* Output buffer information       */
                 ICABUFFER     ssb; /* Secondary status buffer information */
                 ushort    retcode;      /* Return code                     */
                 ulong    reserved);     /* Reserved field                  */
               } ICAGETBUFADDRS_PARMS;

typedef struct
               {
                 ushort    length;       /* Length of buffer                */
                 ushort    offset;       /* Offset of buffer address        */
                 uchar       page;       /* Page of buffer address          */
               } ICABUFFER;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

Returns

ib.length The length of the task’s input buffer.

ib.offset The page offset of the task’s input buffer.

ib.page The page number of the task’s input buffer.

ob.length The length of the task’s output buffer.

ob.offset The page offset of the task’s output buffer.

ob.page The page number of the task’s output buffer.

ssb.length The length of the task’s secondary status buffer.

ssb.offset The page offset of the task’s secondary status buffer.

ssb.page The page number of the task’s secondary status buffer.

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF27	E_ICA_INVALID_TASK

Remarks The Get Buffer Addresses function returns the addresses in page:offset format only.

The ICAGETBUFADDRS function performs the same function as the C Language interface routines **icainbuf**, **icaoutbuf**, and **icasecstatbuf**.

Note: By convention, input buffer is *input/output* from the system unit to the co-processor adapter.

Related Topics None

Get Version

Purpose Gets the release level of this version of the device driver.

Invocation ICAGETVER

Format

```
typedef struct
                {
                  uchar minvc;       /* Minor version code */
                  uchar majvc;       /* Major version code */
                } ICAGETVER_PARMS;
```

Parameters

None

Returns

minvc The minor version code. The field is the fractional portion of the version number (in decimal format). In release 1.1.2, the field would have a value of 2.

majvc The major version code of the Co-Processor AIX Support device driver. The field is the integer portion of the version number. In release 1.2, the field would have a value of 1.

Remarks The ICAGETVER function performs the same function as the C Language interface routine **icagetver**.

Related Topics None

Get Primary Status

Purpose Gets the primary status byte for a task.

Invocation ICAGETPRIMSTAT

Format

```
typedef struct
               {
                 uchar  coprocnum;  /* Co-processor adapter number */
                 uchar    tasknum;  /* Task number                 */
                 uchar        psb;  /* Primary status byte         */
                 ushort   retcode;  /* Return code                 */
                 ulong   reserved;  /* Reserved field              */
               } ICAGETPRIMSTAT_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

tasknum The number of the task.

Returns

psb The value of the task's primary status byte.

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF27	E_ICA_INVALID_TASK

Remarks See the *IBM Realtime Interface Co-Processor Firmware Technical Reference* for the definition of the bits in the primary status byte.

The ICAGETPRIMSTAT function performs the same function as the C Language interface routine **icagetprimstat**.

Related Topics None

Special Events Register

Purpose Registers an application process with the device driver for notification of the Realtime Control Microcode's receipt of an Initialize command.

Invocation ICASEREG

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 uchar   ctrlflag; /* Control flag                */
                 ushort   retcode; /* Return code                 */
                 ulong   reserved; /* Reserved field              */
               } ICASEREG_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

ctrlflag A byte of control bits indicating the events for which the application should be registered. Only one bit is currently defined in the control flag:0x80. When this bit is set, it means that the application should be registered for Initialize commands issued to the Realtime Control Microcode on the co-processor adapter.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0D	E_ICA_INVALID_CONTROL
0xFF15	E_ICA_ALREADY_REG
0xFF25	E_ICA_XMALLOC_FAIL
0xFF30	E_ICA_NO_MORE_RES

Remarks This function allows applications to be notified of Initialize commands issued to the Realtime Control Microcode with the Special Events Wait function. An application must first register with this function before being notified of Initialize commands issued to the Realtime Control Microcode.

The ICASEREG function performs the same function as the C Language interface routine **icasereg**.

Related Topics Special Events Deregister, Special Events Wait

Special Events Wait

Purpose Blocks an application process until the Realtime Control Microcode on a co-processor adapter receives an Initialize command.

Invocation ICASEWAIT

Format

```
typedef struct
               {
                 uchar  coprocnum; /* Co-processor adapter number */
                 ulong    timeout; /* Timeout                     */
                 ushort   retcode; /* Return code                 */
               } ICASEWAIT_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

timeout The time in milliseconds to wait for the Realtime Control Microcode to receive an Initialize command. The **timeout** field is treated as an unsigned integer. If it is zero, the call returns immediately indicating whether or not the Realtime Control Microcode has previously received an Initialize command.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF0B	E_ICA_TIMEOUT
0xFF17	E_ICA_NOT_REG
0xFF28	E_ICA_INTR

Remarks The Special Events Wait function returns immediately with no error if an application previously registered and the Realtime Control Microcode received an Initialize command prior to this call. If the Realtime Control Microcode has not yet received the Initialize command, the call blocks the application process until the Initialize command is received by the Realtime Control Microcode or the time specified in the **timeout** parameter expires.

The ICASEWAIT function performs the same function as the C Language interface routine **icasewait**.

Related Topics Special Events Register

Special Events Deregister

Purpose Cancels an application process request to be notified of the Realtime Control Microcode's receipt of an Initialize command.

Invocation ICASEDEREG

Format typedef struct

```
{
  uchar  coprocnum; /* Co-processor adapter number */
  ushort   retcode; /* Return code                 */
  ulong   reserved; /* Reserved field              */
} ICASEDEREG_PARMS;
```

Parameters

coprocnum The logical number of the co-processor adapter.

Returns

retcode The return code set by the device driver:

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF17	E_ICA_NOT_REG

Remarks The Special Events Deregister function cancels a previous application request to be notified of the Realtime Control Microcode's receipt of an Initialize command. Processes should cancel all requests for such notification prior to terminating.

The ICASEDEREG function performs the same function as the C Language interface routine **icasedereg**.

Related Topics Special Event Register

Chapter 5. Application Loader Utility

The application loader utility (**icaldric**) is used to load the Realtime Control Microcode or tasks to the co-processor adapter. The application loader utility consists of three files:

- An executable file **icaldric** that can be invoked from the keyboard, a shell script, or an application program.
- A message file **icaldric.msg** containing messages that can be displayed to standard output or standard error by the application loader utility.
- A parameter file **icaparm.prm** that can override the default configuration parameters for the co-processor adapters.

The path and file names for the application loader are:

```
bin/icaldric  (program)
sys/icaldric.msg  (messages)
sys/icaparm.prm  (parameters)
```

The first task loaded onto a co-processor adapter must be the Realtime Control Microcode (file **icaaim.com** or **icarcm.com**), which is provided with the co-processor adapter. After the Realtime Control Microcode is loaded, the user can load applications or other tasks onto the co-processor adapter. Each task must have a unique task number that is assigned when the task is loaded. The MAXTASK field in the parameter file (**icaparm.prm**) defines the highest task number under which a task can be loaded. (For additional information on the parameter file, see Chapter 3, “Parameter File Description.”)

The following example loads the Realtime Control Microcode onto co-processor adapter 0:

```
icaldric 0 icaxxx.com 0 -reset
```

where:

xxx = **aim** for:

Realtime Interface Co-Processor Multiport/2
Realtime Interface Co-Processor Multiport
X.25 Interface Co-Processor/2
X.25 Interface Co-Processor
X.25 Interface Co-Processor PCI Adapter
ARTIC186 8-Port PCI Adapter.

xxx = **rcm** for:

Realtime Interface Co-Processor Portmaster Adapter/A
Multiport Model 2

The first 0 (zero) represents the first logical co-processor adapter. The last 0 represents the task number, which is always 0 for the Realtime Control Microcode.

The application loader can load only **.exe** or **.com** files. The maximum length of a **.com** file is 64 KB, whereas the length of an **.exe** file is restricted by the amount of free storage on the co-processor adapter at the time the load is attempted. (Refer to the *Realtime Interface Co-Processor Firmware Technical Reference* or, if you are programming in C Language, refer to the *Realtime Interface Co-Processor C Language Support Version 1.03 User's Guide, Volume II - Co-Processor Adapter*.)

The application loader sets up the initial values for the Code Segment Register, Data Segment Register, Stack Segment Register, and Stack Pointer Register in the task header.

Starting the Application Loader Utility

The application loader requires command line arguments to indicate which task to load and how it should be loaded. The command and first three parameters are required and must appear in the order in which they are shown. The remaining options are optional and can appear in any order, but each option can be specified only once. Options and parameters are separated by white space (spaces and/or tab characters).

Application Loader Facility Syntax

```
►►──icaldric──coprocnum──filename──tasknumber──────────────────────►

   ┌──────────────────┐
►──▼─┬────────────────┬┴──────────────────────────────────────────►◄
     ├─-pf──filename──┤
     ├─-ns────────────┤
     ├─-l─────────────┤
     ├─-m──boundary───┤
     ├─-q─────────────┤
     ├─-prm──string───┤
     └─-reset─────────┘
```

icaldric The name of the application loader program.

Parameter **Description**

coprocnum The logical co-processor adapter number where the task is to be loaded. This parameter is required and must be first on the command line.

filename The file name of the task to be loaded. A full pathname may be specified if the task file is not in the current working directory. This parameter is required and must be second on the command line.

tasknumber The task number to be assigned to the task to be loaded. Realtime Control Microcode must be loaded as task 0. Application tasks may be loaded in the range of 1 to the MAXTASK value, as assigned in the parameter file. This parameter is required and must be third on the command line.

-pf *filename* The name of the parameter file to be used to override the default parameters. A full pathname may be specified if the parameter file is not in the current working directory. This parameter is optional and is ignored if the Realtime Control Microcode is not being loaded.

-ns The no-start flag. If specified, the task is loaded onto the co-processor adapter only; it is not started. If not used, the task is started by default. This parameter is optional.

-l The load-low flag for loading in low co-processor adapter memory. Use **-l** (el) to load the task at the lowest possible address. Omit the **-l** flag to load the task at the highest possible address. This parameter is optional.

-m *boundary* The memory boundary in paragraphs on which the task should be loaded. The default task load boundary is one paragraph (16 bytes). If a boundary other than one paragraph is required, the required boundary can be entered using this option. The boundary must be an exact power of 2 (only one bit on in the entire word). The boundary is specified in hexadecimal format. This parameter is optional.

-q The quiet flag. If specified, **icaldric** does not display any messages to standard out or to standard error, even if an error occurs. If messages are suppressed, the application loader messages a return code upon completion. This return code is the same as the application loader message number. Therefore, if messages are suppressed and a non-zero return code is received from the application loader, the meaning of the return code can be found by looking at the application loader message with the same message number. For a description of the application loader messages, see “Application Loader Utility Information Messages” on page D-1. The default is to report the success or failure of the load to standard out or standard error. This parameter is optional.

-prm *string* The parameter string passed to the task parameter block area (offset 0x1C in the task’s header segment). The maximum length of the parameter area is 128 bytes. The parameters are passed as a NULL terminated string. To ensure that the parameter string is passed correctly, enclose multiple-word strings, special characters, and escape sequences in double quotation marks.

Note: When loading Realtime Control Microcode as task 0 on the Realtime Interface Co-Processor Portmaster Adapter/A, if a **-prm** string is not explicitly specified, the application loader uses the following parameter string to disable peer services:

```
ICARCM.COM 1
```

Peer services are not supported by Co-Processor AIX Support. They are described in the *Realtime Interface Co-Processor Firmware Technical Reference*, the *Realtime Interface Co-Processor DOS Support, Version 1.03 (or higher) User’s Guide*, the *Realtime Interface Co-Processor OS/2 Support Version 1.03 (or higher) User’s Guide*, and the *Realtime Interface Co-Processor C Language Support Version 1.03.0 User’s Guide*. (Volume II - Co-Processor Adapter)

-reset This flag causes a reset of the indicated co-processor adapter prior to loading task 0. It is ignored for tasks other than task 0. Note that the reset flag causes task 0 load time to be increased by up to 20 seconds, which is the time it takes the adapter to execute its self-test.

Examples

In the following example, the task USERTASK.EXE is loaded on co-processor adapter 1 as task 2 with messages suppressed. All other arguments have the default values.

```
icaldric 1 USERTASK.EXE 2 -q
```

The following sample loads TASK.EXE as task 1 on card 0.

```
icaldric 0 TASK.EXE 1
```

The next sample loads TASK.EXE as task 2 on card 1. The task is passed the parameter string "TASK.EXE parameter string".

```
icaldric 1 TASK.EXE 2 -prm “parameter string”
```

The parameter string is enclosed by quotation marks in order to include the spaces in the parameter string.

Application Loader Messages and Return Codes

Application loader messages are displayed to show the status of the application loader’s operation. These messages are listed in “Application Loader Utility Information Messages” on page D-1. The application loader also returns corresponding numeric values as its program return value. These return codes are described in “Application Loader Utility Return Codes” on page C-4.

Chapter 6. Online Dump Facility

The Online Dump Facility is a debugging tool that dumps the memory contents and I/O port values of a co-processor adapter to a disk file, which can then be formatted using **frmtdump**, the Dump Formatter Facility. Dump data can be obtained either interactively by the user or automatically using the AutoDump feature. The Online Dump Facility consists of two files:

- An executable file **icadpric** that can be invoked from the keyboard, a shell script, or an application program.
- A message file **icadpric.msg** that contains all the messages that can be displayed to standard output or standard error by the Online Dump Facility.

The ARTIC subdirectory and file names for the Online Dump Facility are:

```
bin/icadpric  (program)
sys/icadpric.msg  (messages)
```

Starting the Online Dump Facility

The Online Dump Facility requires command line parameters. These are shown in the following syntax diagram:

Online Dump Facility Syntax

```
►►──icadpric──coprocnum──┬─-d──┬──┬──────────────────┬──────────────────►◄
                         ├─-ea─┤  └─-dd──directory─┘
                         └─-da─┘
```

icadpric The name of the Online Dump Facility program.

Parameter **Description**

coprocnum The logical number of the co-processor adapter to be dumped. The adapter number is determined by the co-processor adapter's position relative to other co-processor adapters in the system.

-d Indicates that a dump of the co-processor adapter identified by **coprocnum** should be done immediately. If AutoDump has previously been enabled (armed), the dump is done immediately. After the dump is complete, AutoDump is no longer be enabled.

-dd *directory* Specifies the directory where the dump files are to be located. If this parameter is omitted, the dump directory will be the current directory.

-ea Indicates that the co-processor adapter identified by **coprocnum** should be dumped whenever a level 1 error occurs on the co-processor adapter. This is referred to as the *AutoDump feature.*

-da Cancels a previous request for an Auto Dump on the co-processor identified by **coprocnum**.

Note: When the user selects which co-processor adapter to arm for AutoDump, the Online Dump Facility acquires the breakpoint and watchdog timer vectors on the co-processor. It also sets the watchdog timer to a timeout length of approximately 12 milliseconds. When the watchdog timer expires or a breakpoint is reached (level 1 error), the co-processor adapter is automatically dumped without user intervention. The user should verify that the dump drive has enough free storage before arming a co-processor adapter for AutoDump. The dump drive will be the drive determined at the time the co-processor adapter is armed for the dump.

Output Files

A dump creates two files:

ICAME*N*.DMP The file that contains the memory image of the co-processor adapter.

ICASYS*N*.DMP A file containing system information (in binary format) such as the adapter software version number, the adapter register contents, I/O ports, the free memory listing, and task information.

The names of the files produced by the Online Dump Utility are based upon the adapter number where *N* is the logical number of the co-processor adapter that was dumped.

Examples

The following example enables AutoDump on co-processor adapter 2 and directs the output of any resulting dump to be written to the **/tmp** directory:

```
icadpric 2 -ea -dd /tmp
```

The following command can be used to cancel a previously enabled AutoDump on co-processor adapter 2:

```
icadpric 2 -da
```

This example performs an immediate dump of co-processor adapter 0 placing the resulting dump files in the current directory:

```
icadpric 0 -d
```

Online Dump Facility Messages and Return Codes

The Online Dump displays messages to show the status of the dump program operation. These messages are listed in “Online Dump Facility Information Messages” on page D-6. The Online Dump program also returns corresponding numeric values as its program return value. These return codes are described in “Online Dump Facility Return Codes” on page C-7.

Chapter 7. Dump Formatter Facility

The Dump Formatter Facility converts the machine-readable images generated by the Online Dump Facility into a format that can be viewed or printed, or both. The Formatter organizes the dump data into an easy-to-read format, using headers and blocks to group related information. The Dump Formatter Facility consists of three files:

- An executable module **frmtdump** that can be invoked from the keyboard, a shell script, or an application program
- A message file **frmtdump.msg** that contains all the messages that can be displayed to standard output or standard error by the Dump Formatter Facility program
- A profile **frmtdump.pro** that can be used to tailor the output of the formatter for different printers and cause the display of select areas of memory when the memory image file is generated

The ARTIC subdirectory and file names for the Dump Formatter Facility are:

```
bin/frmtdump  (program)
sys/frmtdump.msg  (messages)
sys/frmtdump.pro  (printer profile)
```

Starting the Dump Formatter Facility

The Dump Formatter Facility requires command line parameters. These are shown in the following syntax diagram:

Dump Formatter Facility Syntax

```
►►──frmtdump──(1)──coprocnum──┬──────┬──-s──┬──────────────────┬──────────►◄
                              └──-m──┘      └──-fd──directory──┘
```

Note:

[1] The corresponding dump files **ICAME*N*.DMP** and **ICASYS*N*.DMP** must be in the current directory, where *N* is the coprocnum.

frmtdump The name of the Dump Formatter Facility program.

Parameter **Description**

coprocnum The logical card number *N* of the adapter that produced the dump files **ICAME*N*.DMP** and **ICASYS*N*.DMP**.

-m Generates the memory image file from the file **ICAME*N*.DMP**, where *N* is the logical card number of the card that produced the memory dump file. The file **ICAME*N*.DMP** must be in the current directory.

If both this and the **-s** flag are omitted, the Dump Formatter Facility produces both a memory image file and a system information file.

-s Generates the system information file from the file **ICASYS*N*.DMP.** where *N* is the logical card number of the card that produced the system file. The file **ICASYS*N*.DMP** must be in the current directory. If both this and the **-m** flag are omitted, the Dump

Formatter Facility produces both a memory image file and a system information file.

-fd *directory* Specifies the directory where the formatted files are to be located. If this parameter is omitted, the files will be in the current directory.

Example

The following example formats the system information dump file for co-processor adapter 1 and places the formatted output in the **/u/me/my_formatter_output_dir** directory.

```
frmtdump 1 -s -fd /u/me/my_formatter_output_dir
```

Output Files

The Dump Formatter Facility creates either a memory image file or a system information file, or both. The memory image file is an ASCII representation of an Online Dump Formatter Utility memory image file. The system information file is an ASCII representation of a Dump Utility system file. See Appendix A, "Output File Format for the Dump Formatter Facility" for more details on the format of the files produced by the Dump Formatter Facility.

The names of the files produced by the Dump Formatter Facility are:

- **MEMORY***N***.PRT**, the memory image file, where *N* is the logical number of the co-processor adapter that produced the memory image dump file.
- **SYSINFO***N***.PRT**, the system information file, where *N* is the logical number of the co-processor adapter that produced the system information dump file.

Profile

The output of the Dump Formatter Facility can be tailored for different printers and select portions of co-processor memory can be displayed by setting parameters in the Dump Formatter Facility profile **frmtdump.pro**. If no profile exists, the default parameters are used. The default printer-specific parameters are for the IBM Proprinter. The default parameters are listed under "Default Profile" on page 7-7.

The following conventions apply to parameters in **frmtdump.pro**:

- Each parameter must begin on a new line.
- Numbers can be entered in decimal or hexadecimal format. Hex numbers must be immediately followed by an uppercase or lowercase **h**. Unless otherwise specified, any number outside the proper range is ignored.
- Commas or blanks can be used as delimiters in lists of integers.

Profile Parameters

In listing the parameters, the following assumptions are made:

- An integer in the range 0h through FFh (0 through 255 decimal) inclusive is represented by **nn**.
- An integer in the range 0h through FFFFh (0 through 65535 decimal) inclusive is represented by **NN**.

- Brackets ([ ]) indicate an optional parameter.
- The vertical bar (|) represents a choice. One of the options separated by a vertical bar can be chosen.

Following is a list of the Dump Formatter Facility profile parameters:

BOXCHARS - Printer codes for box characters.

The printer codes for generating box characters in the output files can be specified by adding the following line to the profile:

```
BOXCHARS  = nn [,] nn [,] nn [,] nn [,] nn [,] nn [,]
            nn [,] nn [,] nn [,] nn [,] nn
```

Each **nn** represents the ASCII character code of a box character. The 11 codes are assigned in the order listed in the following table. The printer codes for the IBM Graphics Printer are also listed in the table. If the codes are not set, or if there is an error in the list of codes, they default to the box characters of the IBM Proprinter.

│ Vertical Bar	Default value = B3h (179 decimal)
─ Horizontal Bar	Default value = C4h (196 decimal)
┌ Upper Left Corner	Default value = DAh (218 decimal)
┐ Upper Right Corner	Default value = BFh (191 decimal)
┘ Lower Right Corner	Default value = D9h (217 decimal)
└ Lower Left Corner	Default value = C0h (192 decimal)
┤ Horizontal Line on Vertical Bar (left)	Default value = B4h (180 decimal)
┬ Horizontal Line on Vertical Bar (right)	Default value = C3h (195 decimal)
┬ T Junction	Default value = C2h (194 decimal)
┴ Inverted T Junction	Default value = C1h (193 decimal)
┼ Cross	Default value = C5h (197 decimal)

FORM_FEED - Printer formfeed sequence.

This is the printer sequence for generating a formfeed. To set this parameter, add the following line to the profile:

```
    FORM_FEED = NONE | nn[[,]nn]...
```

A list of up to 16 ASCII character codes can be entered for the formfeed sequence. This allows the user to set the profile for whatever printer is being used. If more than 16 codes are specified, or if any code exceeds 255, the default value 0Ch (12 decimal) is used. 0Ch is the standard ASCII code for formfeed.

Specifying FORM_FEED = NONE indicates that no formfeed character exists for the printer being used. Instead, blank lines are printed in place of the formfeed character to bring the printer to the top of the next page. The PAGE_LINES parameter indicates how many blank lines to print.

LONG - Complete memory listings

If the keyword **LONG** is included in the profile, every line of memory is displayed, regardless of the contents of memory. Omitting the keyword **LONG** can make the memory file **MEMORY*N*.PRT** shorter because redundant lines of memory are not redisplayed.

MEMLIST - Memory dumped by location

This parameter affects the contents of **MEMORY*N*.PRT**, but not of **SYSINFO*N*.PRT**. It specifies which blocks of memory are to be included in the output.

To set this parameter, add the following line to the profile:

```
MEMLIST = [ ALL ] [ NONE ] [ (NN [,] [+]NN) [,] ] ...
```

Specifying **ALL** means that all co-processor adapter memory locations will be displayed in **MEMORY*N*.PRT**. Specifying **NONE** means that none of co-processor adapter memory will be displayed in **MEMORY*N*.PRT**.

Ranges of co-processor adapter memory can be specified in terms of paragraphs (16 bytes). Ranges can be either a lower and upper boundary or a lower boundary and a length. Specifying **(NN1 NN2)** gives memory contents from paragraphs **NN1** to **NN2**, inclusive. **NN1** and **NN2** specify paragraph boundaries and are numbers with the same range as described earlier. Either can be the smaller number, and they do not need to be ordered.

Specifying **(NN1 +NN2)** gives memory contents starting at paragraph **NN1** and continuing for **NN2** paragraphs. For example,

```
MEMLIST = (1000h, +20h)
```

designates a total of 20h consecutive paragraphs, starting at paragraph 1000h (start address = 1000:0000).

```
MEMLIST = ALL
```

gives the contents of all installed co-processor adapter memory.

```
MEMLIST = (1000h, 1FFFh) (3000h, 3FFFh)
```

gives the memory contents of memory addresses 1000:000 through 1FFF:000F and addresses 3000:0000 through 3FFF:000F.

```
MEMLIST = (1000h, +1000h) (3000h, +1000h)
```

gives the same results as the previous example.

Note: For example, memory addresses 2000:0000 through 2FFF:0 00F are not generated.

The default value for MEMLIST is **NONE**. The keywords ALL or NONE override parameters to their left. If conflicting keywords are found, the last (right-most) one overrides the others.

PAGE_LINES - The number of lines per page

If a formfeed code is not available on the target printer, the Dump Formatter Facility will compute how many blank lines to generate.

This parameter is set by adding the following line to the profile:

```
PAGE_LINES = nn
```

The default value is 66 lines per page.

POSTSTRING - Printer postfix sequence

This sequence comes last in the **MEMORY*N*.PRT** and **SYSINFO*N*.PRT** files. Use it to return the printer to a desired state (for example, 80 character-per-line mode and 8 lines per inch). This parameter is set by adding the following line to the profile:

```
POSTSTRING = NONE | [nn[,]...]
```

Specifying **NONE** results in no string being generated at the end of **MEMORY*N*.PRT** or **SYSINFO*N*.PRT**.

The length of this string cannot exceed 256 bytes. If more than 256 integers are entered, or if any integer exceeds a value of 255, the default sequence of POSTSTRING is used.

The default sequence is as follows:

```
POSTSTRING = 12h
```

This character sequence stops compressed character mode printing on the IBM Graphics Printer.

PRESTRING - Printer prefix sequence

This is the sequence that comes first in the **MEMORY*N*.PRT** and **SYSINFO*N*.PRT** files. It is used to force the printer into a desired state.

This parameter is set by adding the following line to the profile:

```
PRESTRING = NONE | [ nn [,] ...]
```

Specifying **NONE** results in no string being presented at the start of **MEMORY*N*.PRT** or **SYSINFO*N*.PRT**. The file then begins with formatted output, instead of printer-specific information.

The length of this string cannot exceed 256 bytes, that is, no more than 256 integers can be presented on the line in **frmtdump.pro**. If more than 256 integers are presented, or if any integer exceeds a value of 255, the default sequence of PRESTRING is used.

The Dump Formatter Facility defaults to the following:

```
PRESTRING = 18h, 0Fh, 1Bh, 41h, 0Ch, 1Bh, 32h,
            1Bh, 36h, 1Bh, 39h, 1Bh, 43h, 42h, 1Bh,
            46h, 1Bh, 48h, 1Bh, 54h, 1Bh, 55h, 0
```

This IBM Proprinter character sequence does the following in this order:

1. Clears the printer buffer
2. Shifts the printer to *compressed character* mode (132 characters per line)
3. Sets line spacing to six lines per inch
4. Selects character set 2 on the IBM Proprinter
5. Cancels any *ignore paper end* command
6. Sets the page length to 66 lines per page
7. Turns off printing in *emphasized* mode
8. Turns off printing in *double strike* mode
9. Turns off printing in *superscript* mode or *subscript* mode
10. Sets the printer for bidirectional printing

Printer codes are explained in depth in your printer's guide to operation.

PRINT_LINES - The number of lines to print per page

This parameter allows you to set the actual number of lines you want printed on a page. This parameter applies to **SYSINFO*N*.PRT** and **MEMORY*N*.PRT**. It cannot exceed the value of the PAGE_LINES parameter.

This parameter is set by adding the following line to the profile:

```
PRINT_LINES = nn
```

The default value is 60 lines per page.

REPCHARSET - Representable character set

Use this parameter to control characters.

Characters are entered as a series of ASCII character code ranges or individual ASCII character codes. ASCII character code ranges are represented as two integers separated by a comma or a space, inside parentheses. All the characters within the range, including the lower and upper boundaries, are added to the representable character set. Integers represent individual ASCII character codes.

This parameter is set by adding the following line to the profile:

```
REPCHARSET = [(nn[,]nn) | nn[,]...]
```

For example, a representable character set of 30 and the range 32 through 255 might be entered as follows:

```
REPCHARSET = 30,(32,255)
```

The period (.) cannot be removed from the set, even if it is omitted from the REPCHARSET parameter.

TASKLIST - Task memory dumped

This parameter affects only the contents of **MEMORY*N*.PRT**. It specifies a list of the tasks to be included in the output. If a task is included in the list, all of its memory is stored in formatted form in **MEMORY*N*.PRT**.

This parameter is set by adding the following line to the profile:

```
TASKLIST = [ ALL ] [NONE ] [[-|+] nn ] [,] ...
```

where *nn* is a task number.

Specifying **ALL** adds all tasks and their associated memory to **MEMORY*N*.PRT**. Specifying **NONE** subtracts all task-related output from **MEMORY*N*.PRT**. The list of tasks to be included can be modified by specifying task numbers individually. A minus sign (-) in front of a task number removes it from the list of tasks being printed. A plus sign (+) in front of a task number adds it to the list of tasks being printed.

Note: If a task number does not have a "+" or a "-" preceding it, "+" is assumed.

Some examples of the TASKLIST line follow. The first example shows how to display all tasks except task 210.

```
TASKLIST = ALL -210
```

The second example shows how to display tasks 15h and 16h.

```
TASKLIST = 15h 16h
```

The default setting for TASKLIST is **NONE**.

TITLE - Title for formatted output

This parameter assigns a title to the Dump Formatter Facility output files. This title is printed at the top of each page in the **MEMORY*N*.PRT** and **SYSINFO*N*.PRT** files.

To set this parameter, add the following line to the profile:

```
TITLE = title_string
```

The title string ends with a carriage return, that is, it must fit on a single line.

The default setting for TITLE is the character string "Dump Information".

USER_SEG - User-Selected Segment

This parameter sets a special field indicating memory addresses with an offset from the beginning of a user-selected segment.

An address falling within the 64 KB block of memory starting at this selected segment appears in the form segment:offset. The physical address and page:offset addresses will be displayed for memory outside the 64 KB block. Memory addresses outside the 64 KB block are not displayed in segment:offset format, since different segment values are required to represent these addresses.

To set this parameter, add the following line to the profile:

```
USER_SEG = NN
```

The default for this parameter is segment 0044h, which is the start of the co-processor adapter Interface Block (IB).

Default Profile

The following profile contains default values and is supplied under file name **frmtdump.pro**. The values are assumed if **frmtdump.pro** cannot be found. The profile need not be present for the Dump Formatter Facility to work.

```
BOXCHARS = B3H,C4H,DAH,BFH,D9H,C0H,C3H,B4H,C2H,C1H,C5H
FORM_FEED = 0CH
MEMLIST = NONE
PAGE_LINES = 66
POSTSTRING =  12H
PRESTRING = 18H, 0FH, 1BH, 41H, 0CH, 1BH, 32H, 1BH, 36H, 1BH, 39H,
            1BH, 43H, 42H, 1BH, 46H, 1BH, 48H, 1BH, 54H, 1BH, 55H, 0
PRINT_LINES= 60
REPCHARSET =  (32, 255)
TASKLIST =  NONE
TITLE = Untitled
USER_SEG = 44H
```

Note: The preceding printer-specific parameters are for the IBM Proprinter.

Dump Formatter Messages and Return Codes

The messages and a brief explanation of each one displayed by the Dump Formatter Facility are listed in “Dump Formatter Facility Information Messages” on page D-9. “Dump Formatter Facility Return Codes” on page C-9 lists the Dump Formatter return codes and a brief explanation of each.

Chapter 8. C Language Interface Routines

The C Language interface routines provide a programming interface for system unit programs to the Co-Processor AIX Support device driver and any installed co-processor adapters. Applications linked to the Co-Processor AIX Support library (**libric.a**) can issue commands and access co-processor adapter memory and task parameters. The file **icaclib.h** contains declarations for the C Language interface routines in **libric.a**. Include **icaclib.h** in the source for your application programs.

Call Example

Following is an example of a call to the C Language interface library routine **icasecstatbuf**:

```
#include <icaclib.h>     /* C Language interface routine declarations */
#include <fcntl.h>

int fd;                            /* Device driver handle           */
ushort rc;                         /* Return code                    */
ICABUFFER ssb;                     /* Secondary status buffer        */
      .
      .

   if ((fd = open("/dev/artic",O_RDONLY)) == -1) {
      printf("open on /dev/artic failed, errno = %d\n", errno);
      .
      .

   }
   else {
      .
      .
                                 /* Get the secondary status buffer    */
                                 /* address, length for task 7, card 0 */
      if ((rc = icasecstatbuf(fd,0,7,&ssb)) != 0) {
         printf("call to icasecstatbuf failed, return code = %d\n",rc);
      .
      .
      }
      else {
      .
      .
```

Declarations

The following declarations define the function calls and parameter types used by the C Language interface routines and are prototyped in **icaclib.h**. Error codes are passed back as function values:

icareset
icareadmem
icawritemem
icaintreg
icaintwait
icaintdereg
icaissuecmd
icagetparms
icagetprimstat
icainbuf
icaoutbuf
icasecstatbuf
icagetbuffers
icagetver
icasereg
icasewait
icasedereg

Examples of these routines are in the sample system unit program **suclib.c**, which is on your Co-Processor AIX Support program diskette. Instructions on how to run the sample programs are in Appendix E, “Using the Sample Programs.”

icareset

Purpose Issues a hardware reset to the co-processor adapter.

Format

```
ushort icareset(int         fd,    /* File descriptor             */
                uchar coprocnum);  /* Co-processor adapter number */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter to reset.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0B	E_ICA_TIMEOUT
0xFF23	E_ICA_INVALID_FD
0xFF33	E_ICA_RESET_FAILED

Remarks The **icareset** function issues a hardware reset to the co-processor adapter. The Realtime Control Microcode and all other tasks are unloaded and the adapter performs a power-on self test (POST).

A reset of the co-processor adapter may also be performed when loading the Realtime Control Microcode to the co-processor adapter by using the **-reset** application loader option.

Related Topics None

icareadmem

Purpose Reads from a co-processor adapter's memory into an application buffer.

Format

```
ushort icareadmem(int    fd,          /* File descriptor            */
                  uchar  coprocnum,   /* Co-processor adapter number */
                  ulong  length,      /* Length                     */
                  ushort segpage,     /* Segment/Page               */
                  ushort offset,      /* Offset                     */
                  uchar  addr_format, /* Address format             */
                  uchar  *buffptr);   /* Destination buffer pointer  */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

length The number of bytes to read from co-processor adapter memory.

segpage The segment or page of the adapter memory address. The interpretation of this field is determined by the **addr_format** field.

offset The offset of the adapter memory address. The interpretation of this field is determined by the **addr_format** field.

addr_format The control field determining the address format.

Value:	**Address interpretation:**
0x00	The **segpage** parameter is a segment in co-processor adapter memory, and the **offset** is an offset within that segment.
0xFF	The **segpage** parameter is a page in co-processor adapter memory, and the **offset** is an offset within that page.
0x01	The **segpage** and **offset** parameters are a 32-bit physical address in adapter memory. The least significant 16-bits are in the **offset** field.

buffptr A pointer to the application buffer where the adapter memory is to be copied.

Note: All addressing formats are converted to page:offset formats by the Co-Processor AIX Support device driver prior to accessing co-processor adapter memory. When accessing the upper 1 MB on a 2 MB Realtime Interface Co-Processor Portmaster Adapter/A, page:offset format must be used because the segment: format can refer only to addresses in the 0–1 MB range.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF07	E_ICA_INVALID_PAGE
0xFF08	E_ICA_INVALID_OFFSET
0xFF09	E_ICA_INVALID_FORMAT
0xFF23	E_ICA_INVALID_FD
0xFF2C	E_ICA_NOMEM

Remarks The **icareadmem** function reads from co-processor adapter memory into a system unit application buffer. The address in adapter memory can be specified either as a segment and offset or as a page and offset. It is the responsibility of the application to recognize that any data read from adapter memory with the **icareadmem** call will be in Intel (byte-swapped) format.

Related Topics icawritemem

icawritemem

Purpose Writes to a co-processor adapter's memory from an application buffer.

Format

```
ushort icawritemem(int    fd,           /* File descriptor            */
                   uchar  coprocnum,    /* Co-processor adapter number */
                   ulong  length,       /* Length                     */
                   ushort segpage,      /* Segment/Page               */
                   ushort offset,       /* Offset                     */
                   uchar addr_format,   /* Address format             */
                   uchar  *buffptr);    /* Source buffer pointer      */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

length The number of bytes to write from co-processor adapter memory. A value of 0 indicates that 64 KB should be written.

segpage The segment or page of the adapter memory address. The interpretation of this field is determined by the **addr_format** field.

offset The offset of the adapter memory address. The interpretation of this field is determined by the **addr_format** field.

addr_format The control field determining the address format.

Value:	Address interpretation:
0x00	The **segpage** parameter is a segment in co-processor adapter memory, and the **offset** is an offset within that segment.
0xFF	The **segpage** parameter is a page in co-processor adapter memory, and the **offset** is an offset within that page.
0x01	The **segpage** and **offset** parameters are a 32-bit physical address in adapter memory. The least significant 16-bits are in the **Offset** field.

buffptr A pointer to the application buffer that contains the data to be written to adapter memory.

Note: All addressing formats are converted to page:offset formats by the Co-Processor AIX Support device driver prior to accessing co-processor adapter memory. When accessing the upper 1 MB on a 2 MB Realtime Interface Co-Processor Portmaster Adapter/A, page:offset format must be used because the segment: format can only refer to addresses in the 0–1 MB range.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF07	E_ICA_INVALID_PAGE
0xFF08	E_ICA_INVALID_OFFSET
0xFF09	E_ICA_INVALID_FORMAT
0xFF23	E_ICA_INVALID_FD
0xFF2C	E_ICA_NOMEM

Remarks The **icawritemem** function writes to co-processor adapter memory from a system unit application buffer. The address in adapter memory can be specified either as a segment and offset or as a page and offset. It is the responsibility of the application to ensure that any data to be copied to adapter memory with the **icawritemem** call is in Intel (byte-swapped) format.

Related Topics icareadmem

icaintreg

Purpose Registers an application process with the device driver for notification of a specific task interrupt.

Format

```
ushort icaintreg(int   fd,         /* File descriptor             */
                 uchar coprocnum,  /* Co-processor adapter number */
                 uchar tasknum);   /* Task number                 */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF15	E_ICA_ALREADY_REG
0xFF23	E_ICA_INVALID_FD
0xFF25	E_ICA_XMALLOC_FAIL
0xFF27	E_ICA_INVALID_TASK
0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks This function allows applications to be notified of task interrupts by way of the **icaintwait** function. An application must first register with **icaintreg** before being notified of task interrupts.

Related Topics icaintdereg, icaintwait

icaintwait

Purpose Blocks an application process until a specific task on a co-processor adapter interrupts the system unit.

Format

```
ushort icaintwait(int   fd,        /* File descriptor             */
                  uchar coprocnum, /* Co-processor adapter number */
                  uchar tasknum,   /* Task number                 */
                  ulong timeout);  /* Timeout                     */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

timeout The time in milliseconds to wait for a task interrupt. If this parameter is 0, the call returns immediately, indicating whether or not a previous task interrupt has occurred.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0B	E_ICA_TIMEOUT
0xFF17	E_ICA_NOT_REG
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK
0xFF28	E_ICA_INTR
0xFF30	E_ICA_NO_MORE_RES
0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks This call returns immediately with no error if an application previously registered for the task interrupt and the interrupt occurred prior to this call. If the interrupt did not occur previously, **icaintwait** blocks the application process until the specified task interrupts or the time specified in the **timeout** parameter expires. If multiple interrupts by the same task occur prior to this call, the application is notified only once. An application must first register with **icaintreg** before being notified of task interrupts. For the Interrupt Wait routine, the error task (0xFE) is always a valid task number and will not result in the E_ICA_INVALID_TASK error code being returned.

Related Topics icaintreg

icaintdereg

Purpose Cancels the request by the application process to be notified of a specific task interrupt.

Format

```
ushort icaintdereg(int   fd,         /* File descriptor            */
                   uchar coprocnum,  /* Co-processor adapter number */
                   uchar tasknum);   /* Task number                */
```

Parameters

fd	The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.
coprocnum	The logical number of the adapter.
tasknum	The task number.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF17	E_ICA_NOT_REG
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK
0xFF31	E_ICA_BAD_OPEN_HANDLE

Remarks This function cancels a previous application request to be notified of a specific task interrupt. Processes should cancel all requests for task interrupt notification prior to terminating.

Related Topics icaintreg, icaintwait

icaissuecmd

Purpose Issues a command to a task with an option to copy parameter data from an application buffer to the task's output buffer before issuing the command.

Format

```
ushort icaissuecmd(int          fd,  /* File descriptor             */
                   uchar coprocnum,  /* Co-processor adapter number */
                   uchar   tasknum,  /* Task number                 */
                   uchar   cmdcode,  /* Command code                */
                   ushort   length,  /* Length of parameter buffer  */
                   ulong   timeout,  /* Timeout                     */
                   uchar  *prmptr);  /* Pointer to parameters       */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

cmdcode The command code to put in the task's Buffer Control Block (BCB).

length The number of bytes in the parameter block to be copied to the task's output buffer. A value of zero indicates that nothing should be written to the task's output buffer.

timeout The number of milliseconds to wait for the Realtime Control Microcode to respond to the command.

prmptr A pointer to the application buffer containing the data to be written to the task's output buffer. This field is ignored if the length parameter is zero.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS[1]
0xFF0B	E_ICA_TIMEOUT
0xFF11	E_ICA_BAD_PCSELECT
0xFF12	E_ICA_CMD_REJECTED
0xFF14	E_ICA_OB_SIZE
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK
0xFF30	E_ICA_NO_MORE_RES
0xFF31	E_ICA_BAD_OPEN_HANDLE

[1] A primary status of 0x00 is allowed if the Realtime Control Microcode is not yet loaded so applications can send commands to the ROS.

Remarks The **icaissuecmd** function issues a command to a task. The caller has the option of copying parameter information into the task's output buffer before the command is issued. It is the responsibility of the application to ensure that any parameter data to be copied to the task's output buffer is in Intel (byte-swapped) format.

Related Topics None

icagetparms

Purpose Obtains configuration parameter information for a co-processor adapter.

Format

```
ushort icagetparms(int      fd,          /* Co-processor adapter number */
                   uchar    coprocnum,   /* Task number                 */
                   ICAPARMS *prmbuf);    /* Parameter buffer            */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

prmbuf The address of a structure to receive the parameter information. The structure has the following format:

```
typedef struct {
          ushort io_addr;  /* Address of I/O ports       */
          uchar maxtask;   /* Maximum task number        */
          uchar maxpri;    /* Maximum task priorities    */
          uchar maxqueue;  /* Maximum queues             */
          uchar maxtime;   /* Maximum timers             */
          uchar int_level; /* Adapter interrupt level    */
          uchar ssw_size;  /* Shared storage window size */
}ICAPARMS;
```

where the fields are defined as follows:

io_addr The base address of the adapter's I/O ports.

maxtask The highest task number that can be loaded on the adapter.

maxpri The highest value of a task's priority. The highest priority level is 1, whereas the lowest priority level has the maximum value.

maxqueue The highest queue number that can be allocated on the adapter.

maxtime The highest timer number that can be allocated on the adapter.

int_level The interrupt level on which the adapter interacts with the system unit.

ssw_size A code indicating the size of the shared storage window.

The following table indicates what size window each value represents:

Size Code	Window Size (in KB)
0	8 (default)
1	16
2	32
3	64

Notes:

1. The default changed from 32 KB to 8 KB in Version 1.1.3.
2. For information on changing the window size, refer to Appendix F, “Changing the Shared Storage Window Size.”

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF23	E_ICA_INVALID_FD

Remarks Some of the parameters returned by the function (MAXTASK, MAXPRI, MAXTIME and MAXQUEUE) can be defined in the parameter file (**icaparm.prm**), described in Chapter 3, “Parameter File Description.” The device driver uses the parameters or defaults when loading the Realtime Control Microcode onto a co-processor adapter.

Related Topics None

icagetprimstat

Purpose Obtains the primary status byte for a task.

Format

```
ushort icagetprimstat(int   fd,           /* File descriptor             */
                      uchar coprocnum,    /* Co-processor adapter number */
                      uchar tasknum,      /* Task number                 */
                      uchar *primstat);   /* Primary status byte         */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

primstat The returned value of the task's primary status byte.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK

Remarks See the *Realtime Interface Co-Processor Firmware Technical Reference* for the definition of the bits in the primary status byte.

Related Topics None

icainbuf

Purpose Gets the address and length of a task's input buffer.

Format

```
ushort icainbuf(int       fd,          /* File descriptor               */
                uchar     coprocnum,   /* Co-processor adapter number   */
                uchar     tasknum,     /* Task number                   */
                ICABUFFER *ib);        /* Input buffer information      */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

ib The address of a structure to receive the input buffer's address and length. The structure has the following format:

```
typedef struct {
   ushort length;
   ushort offset;
   uchar  page;
}ICABUFFER;
```

where the fields are defined as follows:

length The input buffer's length

offset The input buffer's offset (page:offset format)

page The input buffer's page number

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK

Remarks The **icainbuf** function returns the address in page:offset format only.

Related Topics None

icaoutbuf

Purpose Gets the address and length of a task's output buffer.

Format

```
ushort icaoutbuf(int   fd,             /* File descriptor               */
                 uchar coprocnum,      /* Co-processor adapter number   */
                 uchar tasknum,        /* Task number                   */
                 ICABUFFER *ob);       /* Output buffer information     */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

ob The address of a structure to receive the output buffer's address and length. The structure has the following format:

```
typedef struct {
   ushort length;
   ushort offset;
   uchar  page;
}ICABUFFER;
```

where the fields are defined as follows:

length The output buffer's length

offset The output buffer's offset (page:offset format)

page The output buffer's page number

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK

Remarks The **icaoutbuf** function returns the address in page:offset format only.

Related Topics None

icasecstatbuf

Purpose Gets the address and length of a task's secondary status buffer.

Format

```
ushort icasecstatbuf(int       fd,         /* File descriptor            */
                     uchar     coprocnum,  /* Co-processor adapter number */
                     uchar     tasknum,    /* Task number                */
                     ICABUFFER *ssb);      /* Secondary status buffer    */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

ssb The address of a structure to receive the secondary status buffer's address and length. The structure has the following format:

```
typedef struct {
   ushort length;      /* Length of buffer             */
   ushort offset;      /* Offset of buffer address     */
   uchar  page;        /* Page of buffer address       */
}ICABUFFER;
```

where the fields are defined as follows:

length The output buffer's length

offset The output buffer's offset (page:offset format)

page The output buffer's page number

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF0C	E_ICA_STATUS_NOT_READY
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK

Remarks The **icasecstatbuf** function returns the address in page:offset format only.

Related Topics None

icagetbuffers

Purpose Gets the address and length of a task's input buffer, output buffer, and secondary status buffer.

Format

```
ushort icagetbuffers(int       fd,          /* File descriptor             */
                     uchar     coprocnum,   /* Co-processor adapter number */
                     uchar     tasknum,     /* Task number                 */
                     ICABUFFER *ib,         /* Input buffer                */
                     ICABUFFER *ob,         /* Output buffer               */
                     ICABUFFER *ssb);       /* Secondary status buffer     */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

tasknum The task number.

ib, ob, and ssb The addresses of three structures to receive the input, output, and secondary status buffers' address and length. Each structure has the following format:

```
  typedef struct {
    ushort length;          /* Length of buffer         */
    ushort offset;          /* Offset of buffer address */
    uchar  page;            /* Page of buffer address   */
 }ICABUFFER;
```

where the fields are defined as follows:

ib.length The input buffer's length.

ib.offset The input buffer's offset (page:offset format)

ib.page The input buffer's page number

ob.length The output buffer's length

ob.offset The output buffer's offset (page:offset format)

ob.page The output buffer's page number

ssb.length The secondary status buffer's length

ssb.offset The secondary status buffer's offset (page:offset format)

ssb.page The secondary status buffer's page number

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF23	E_ICA_INVALID_FD
0xFF27	E_ICA_INVALID_TASK

Remarks The **icagetbuffers** function returns the address in page:offset format only.

Related Topics None

icagetver

Purpose Gets the release level of this version of the device driver.

Format

```
ushort icagetver(int    fd,             /* File descriptor                   */
                 ushort *vernum);       /* Version and release number        */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

vernum The major and minor portions of the device driver. The most significant byte is an unsigned character and represents the minor version code (such as 0x02 if the version was 1.2). The least significant byte is the major version code and is also meant to be interpreted as an unsigned character.

Returns

0x0000	NO ERROR
0xFF23	E_ICA_INVALID_FD

Remarks This function allows an application to know which version of the Co-Processor AIX Support device driver is installed.

Related Topics None

icasereg

Purpose Registers an application process with the device driver for notification of the Realtime Control Microcode's receipt of an Initialize command.

Format

```
ushort icasereg(int   fd,         /* File descriptor            */
                uchar coprocnum,  /* Co-processor adapter number */
                uchar ctlflag);   /* Control flag               */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

ctlflag Control bits indicating the events for which the application should be registered. At this time, one bit is defined in the **ctlflag** parameter: 0x80. When this bit is set, it means the application should be registered for Initialize commands issued to the Realtime Control Microcode on the co-processor adapter.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF0D	E_ICA_INVALID_CONTROL
0xFF15	E_ICA_ALREADY_REG
0xFF23	E_ICA_INVALID_FD
0xFF25	E_ICA_XMALLOC_FAIL
0xFF30	E_ICA_NO_MORE_RES

Remarks This function allows applications to be notified of Initialize commands issued to the Realtime Control Microcode by way of the **icasewait** function. An application must first register with **icasereg** before being notified of Initialize commands issued to the Realtime Control Microcode.

Related Topics icasedereg, icasewait

icasewait

Purpose Blocks the application process until the Realtime Control Microcode on a specified co-processor adapter receives an Initialize command.

Format

```
ushort icasewait(int   fd,           /* File descriptor              */
                 uchar coprocnum,    /* Co-processor adapter number */
                 ulong timeout);     /* Timeout                      */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

timeout The time in milliseconds to wait for the Realtime Control Microcode to receive an Initialize command. If this parameter is 0, the call returns immediately indicating whether or not the Realtime Control Microcode has previously received an Initialize command.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF0B	E_ICA_TIMEOUT
0xFF17	E_ICA_NOT_REG
0xFF23	E_ICA_INVALID_FD
0xFF28	E_ICA_INTR

Remarks This call returns immediately with no error if an application previously registered with the **icasereg** call, and the Realtime Control Microcode received an Initialize command prior to this call. If the Realtime Control Microcode has not yet received the Initialize command, this call blocks the application process until the Initialize command is received by the Realtime Control Microcode or the time specified in the **timeout** parameter has expired.

Related Topics icasereg

icasedereg

Purpose Cancels the request by the application process to be notified of the Realtime Control Microcode's receipt of an Initialize command.

Format

```
ushort icasedereg(int   fd,           /* File descriptor            */
                  uchar coprocnum);   /* Co-processor adapter number */
```

Parameters

fd The file descriptor for the Co-Processor AIX Support device driver returned by a previous call to the **open** system subroutine.

coprocnum The logical number of the adapter.

Returns

0x0000	NO ERROR
0xFF05	E_ICA_INVALID_COPROC
0xFF06	E_ICA_INVALID_TASK_STATUS
0xFF17	E_ICA_NOT_REG
0xFF23	E_ICA_INVALID_FD

Remarks This function cancels a previous application request to be notified of the Realtime Control Microcode's receipt of an Initialize command. Processes should cancel all requests for such notification prior to terminating.

Related Topics icasereg

Appendix A. Output File Format for the Dump Formatter Facility

This appendix shows samples of the output files generated by the Dump Formatter Facility described in Chapter 7, "Dump Formatter Facility," which formats a co-processor adapter's dump file for viewing and printing.

Memory Image File

Following is a sample of the formatted output file **MEMORY*N*.PRT** that was produced from the dump of a Multiport/2 adapter with 512 KB of memory. Each page of the output listing is preceded by a form feed character.

```
CO-PROCESSOR ADAPTER 0 MEMORY DUMP                                   12:05:01     Fri Feb 22, 1991     PAGE 1
Dump Information

USER-SET |PHY.  |  8K   |                 MEMORY CONTENTS                                     |    CHARACTER REPRESENTATION
 SEGMENT |ADDR. | PAGES |+00      +04      +08      +0C     |+10      +14      +18      +1C      | 0123456789ABCDEF 0123456789ABCDEF
-----------------------------------------------------------------------------------------------------------------------------------
         |000000|00:0000|001BCE78 56312E34 00000000 00000000|00000000 00000000 00000000 00000000| ...xV1.4........ ................
         |000020|00:0020|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
Addresses 00040 through 00070:                 Same As The Line Above
         |000080|00:0080|FF00FF00 7000C800 00000000 00000000|00000000 00000000 00000000 00000000| ....p........... ................
         |0000A0|00:00A0|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
Addresses 000C0 through 000F0:                 Same As The Line Above
         |000100|00:0100|54000700 26000000 54000700 26000000|74000000 0000C000 0A000000 0000C000| T...&...T...&... t...............
         |000120|00:0120|54000700 20000000 54000700 20000000|74000000 0000C000 0A000000 0000C000| T.......T....... t...............
         |000140|00:0140|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
Addresses 00160 through 00170:                 Same As The Line Above
         |000180|00:0180|9C007000 30004000 5600FF00 FA00F000|00000000 00000000 00002A00 2A00FB00| ..p.0.@.V....... ..........*.*...
         |0001A0|00:01A0|06003000 00000000 00000000 FF00FF00|FF00FF00 FF00FF00 05000500 2500FF00| ..0............. ............%...
         |0001C0|00:01C0|06000000 FF003B00 00000000 00000000|06000000 FF003B00 00000000 00000000| ......;......... ......;.........
         |0001E0|00:01E0|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
         |000200|00:0200|C1000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
         |000220|00:0220|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
Addresses 00240 through 00270:                 Same As The Line Above
         |000280|00:0280|C1000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
         |0002A0|00:02A0|60000000 0A00F000 00005000 21000400|0000FF00 07000000 C0000000 00000000| ..........P.!... ................
         |0002C0|00:02C0|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
         |0002E0|00:02E0|00000000 00000000 00000000 00FC0000|00000000 00000000 00000000 00000000| ................ ................
         |000300|00:0300|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF| ................ ................
         |000320|00:0320|3000FF80 12801280 00000700 00000100|02800100 03000300 02003000 07000700| 0............... ..........0.....
         |000340|00:0340|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF|00000200 00002820 00000200 000029A0| ................ ......(.......).
         |000360|00:0360|0A060024 FFFF2180 00000000 00000000|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF| ...$..!......... ................
         |000380|00:0380|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF| ................ ................
         |0003A0|00:03A0|38F0F83F 7900F841 B8C0FFFF FFFFFFFF|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF| ...?y........... ................
         |0003C0|00:03C0|0000F0FF 0000F0FF 00000000 00000000|0000F0FF 0000F0FF 00000000 00000000| ................ ................
         |0003E0|00:03E0|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF|FFFFFFFF FFFFFFFF FFFFFFFF FFFFFF20| ................ ................
         |000400|00:0400|440707FC 862000B0 448007C1 86AB0002|30000000 0000C047 0A1D0074 0000C000| D.... ..D....... 0......G...t....
         |000420|00:0420|44230723 802300FC 44050700 8000007A|3007007D 0000C000 0A040008 0084C000| D#.#.#..D......z 0..}............
0044:0000|000440|00:0440|FFFF0700 20202020 C500CE78 8703CE78|90005B02 00000F00 00009D04 6200CE78| ...........x...x ..[.........b..x
0044:0020|000460|00:0460|5505CE78 C903CE78 00000000 00000000|00000000 00000000 0080C1C1 03000303| U..x...x........ ................
                                                .
                                                .
                                                .

         |07F660|3F:1660|EB098B16 264D8A87 0A4DEEF8 C38AD88B|162A4DEC 3CC07405 B402EB17 9032FFD1| ....&M...M...... .*M.<.t......2..
         |07F680|3F:1680|E38B1628 4DED2287 224D7505 B401EB03|90B400B0 198866FF F9C3B018 F9C30000| ...(M."."Mu..... ......f.........
         |07F6A0|3F:16A0|00000000 00000000 00000000 00000000|00000000 00000000 00000000 00000000| ................ ................
Addresses 7F6C0 through 7FFB0:                 Same As The Line Above
         |07FFC0|3F:1FC0|00000000 00000000 00000000 00000000|00000000 00000000 00000000 9E13E707| ................ ................
         |07FFE0|3F:1FE0|00000000 87032500 E2560000 58FF0000|00000000 CE788F06 917A46F2 00000000| ......%..V..X... .....x...zF.....
```

System Information File

Following is a sample of the formatted output file **SYSINFO*N*.PRT** that was produced from the dump of a Realtime Interface Co-Processor Multiport/2 adapter with 512 KB of memory. The adapter had four RS-232 ports and four RS-422 ports and had Realtime Control Microcode (**icaaim.com**) and Realtime Interface Co-Processor Extended Services (**ricps.com** and **riccs.com**) loaded in the following configuration:

Task name	Task number
icaaim.com	0
ricps.com	2
riccs.com	3

Each page of the output listing is preceded by a form feed character.

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                          12:09:25    Fri Feb 22, 1991    PAGE 1
Dump Information

┌──────────────────────────────────────────────────────────────┐
│                     GENERAL INFORMATION                      │
├─────────────────────┬──────────┬───────────────────┬─────────┤
│ Dump Date           │ 02/22/91 │ Dump Time         │ 12:02:28│
├─────────────────────┼──────────┼───────────────────┼─────────┤
│ Co-Proc. Logical #  │    0     │ RAM Size          │   512K  │
├─────────────────────┼──────────┼───────────────────┼─────────┤
│ AIX Version         │   3.1    │ ROS/ROM Version   │   01.4  │
├─────────────────────┼──────────┼───────────────────┼─────────┤
│ Dump Version        │   1.00   │ Formatter Version │   1.00  │
├─────────────────────┼──────────┼───────────────────┼─────────┤
│ I/O Base Address    │   02A0   │                   │         │
└─────────────────────┴──────────┴───────────────────┴─────────┘


CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                          12:09:25    Fri Feb 22, 1991    PAGE 2
Dump Information

┌───────────────────────────────────────────────────────────────────────────────────────┐
│                    CO-PROCESSOR ADAPTER 80186 CPU REGISTER VALUES                     │
├───────────────────────────────────────────────────────────────┬───────────────────────┤
│ AX=0000  BX=0000  CS=FC00  SS=0000  DS=0000  ES=0000  BP=03C9 │       FLAGS=F046      │
│ CX=0000  DX=0280  IP=151E  SP=02F0  SI=005B  DI=0008          │ OF DF IF SF ZF AF PF CF│
│  CS:IP=FD51E SS:SP=002F0 SS:BP=003C9 DS:SI=0005B ES:DI=00008  │  0  0  0  0  1  0  1  0│
└───────────────────────────────────────────────────────────────┴───────────────────────┘
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                   12:09:25    Fri Feb 22, 1991    PAGE 3
Dump Information

+---------------------------------------------------------------------------------------------------------------------+
|                      CO-PROCESSOR ADAPTER 80186 PERIPHERAL CONTROL BLOCK (BASE I/O ADDRESS = FF00H)                   |
+------+--------------------------------------------+-----+------+--------------------------------------------+-----+
|OFFSET|                                            |     |OFFSET|                                            |     |
| INTO |              REGISTER NAME                 |VALUE| INTO |              REGISTER NAME                 |VALUE|
| PCB  |                                            |     | PCB  |                                            |     |
+------+--------------------------------------------+-----+------+--------------------------------------------+-----+
|  50  | Timer 0 Count..............................| 0000|  A8  | MPCS.......................................| C0B8|
|  52  | Timer 0 Max Count A                        | 0002|  C0  | DMA Channel 0 Source Pointer               | 0000|
|  54  | Timer 0 Max Count B........................| 0000|  C2  | DMA Channel 0 Srce Pointer (upper 4 bits)..| FFF0|
|  56  | Timer 0 Mode/Control Word                  | 2028|  C4  | DMA Channel 0 Destination Pointer          | 0000|
|  58  | Timer 1 Count..............................| 0001|  C6  | DMA Channel 0 Dest. Pointer (upper 4 bits).| FFF0|
|  5A  | Timer 1 Max Count A                        | 0002|  C8  | DMA Channel 0 Transfer Count               | 0000|
|  5C  | Timer 1 Max Count B........................| 0000|  CA  | DMA Channel 0 Control Word.................| 0000|
|  5E  | Timer 1 Mode/Control Word                  | A029|  D0  | DMA Channel 1 Source Pointer               | 0000|
|  60  | Timer 2 Count..............................| 0811|  D2  | DMA Channel 1 Source Pointer (upper 4 bits)| FFF0|
|  62  | Timer 2 Max Count A                        | 2400|  D4  | DMA Channel 1 Destination Pointer          | 0000|
|  66  | Timer 2 Mode/Control Word..................| 8021|  D6  | DMA Channel 1 Dest. Pointer (upper 4 bits).| FFF0|
|  A0  | UMCS                                       | F038|  D8  | DMA Channel 1 Transfer Count               | 0000|
|  A2  | LMCS.......................................| 3FF8|  DA  | DMA Channel 1 Control Word.................| 0000|
|  A4  | PACS                                       | 0079|  FE  | Relocation Register                        | 20FF|
|  A6  | MMCS.......................................| 41F8|      |                                            |     |
+------+--------------------------------------------+-----+------+--------------------------------------------+-----+
|                                   INTERRUPT CONTROLLER REGISTERS (MASTER MODE)                                      |
+------+--------------------------------------------+-----+------+--------------------------------------------+-----+
|  22  | EOI........................................| 80FF|  32  | Timer Control..............................| 0001|
|  24  | Poll                                       | 0000|  34  | DMA 0 Control                              | 0003|
|  26  | Poll Status................................| 0000|  36  | DMA 1 Control..............................| 0003|
|  28  | Mask                                       | 0000|  38  | INT0 Control                               | 0002|
|  2A  | Priority Mask..............................| 0007|  3A  | INT1 Control...............................| 0030|
|  2C  | In-Service                                 | 0001|  3C  | INT2 Control                               | 0007|
|  2E  | Interrupt Request..........................| 0001|  3E  | INT3 Control...............................| 0007|
|  30  | Interrupt Controller Status                | 8002|      |                                            |     |
+------+--------------------------------------------+-----+------+--------------------------------------------+-----+


CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                   12:09:25    Fri Feb 22, 1991    PAGE 4
Dump Information

+-------------------------------------------------------------------------------------+
|                   SPECIAL CO-PROCESSOR ADAPTER 80186 I/O PORTS/REGISTERS              |
+-------------------------------------------------------------------------------------+
|               CONFIGURATION SWITCHES (L1,L2,L4,BN,XT,BW,M1,M2) = 11111111            |
+--------------------+-------------+-------+--------------------+-------------+-------+
| PORT/REGISTER NAME | I/O ADDRESS | VALUE | PORT/REGISTER NAME | I/O ADDRESS | VALUE |
+--------------------+-------------+-------+--------------------+-------------+-------+
| Initialization Even|    0004     |  0A   | Parity 1           |    000E     |  04   |
+--------------------+-------------+-------+--------------------+-------------+-------+
| Initialization Odd |    0006     |  F0   | Parity 2           |    0010     |  00   |
+--------------------+-------------+-------+--------------------+-------------+-------+
| NMI Mask           |    0008     |  00   | Daughter Board 0 ID|    0200     |  C1   |
+--------------------+-------------+-------+--------------------+-------------+-------+
| NMI Status         |    000A     |  50   | Daughter Board 1 ID|    0280     |  C1   |
+--------------------+-------------+-------+--------------------+-------------+-------+
| Parity 0           |    000C     |  21   | Extended Interface |    0086     |  C8   |
+--------------------+-------------+-------+--------------------+-------------+-------+
| Window Size        |    001A     |  00   | Clocking opt. 0 & 1|    0880     | 0000  |
+--------------------+-------------+-------+--------------------+-------------+-------+


CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                   12:09:25    Fri Feb 22, 1991    PAGE 5
Dump Information

+---------------------------------------------------------------------------------------------------------+
|                                 SPECIAL SYSTEM UNIT I/O PORT/REGISTER VALUES                              |
+-------------+-------------+-------+-------------+-------------+-------+-------------+-------------+-------+
|    NAME     | I/O ADDRESS | VALUE |    NAME     | I/O ADDRESS | VALUE |    NAME     | I/O ADDRESS | VALUE |
+-------------+-------------+-------+-------------+-------------+-------+-------------+-------------+-------+
|Page Location|    02A0     |  60   |Data Register|    02A3     |  00   |Command Reg. |    02A6     |  10   |
+-------------+-------------+-------+-------------+-------------+-------+-------------+-------------+-------+
|Meg. Location|    02A1     |  00   |Task Register|    02A4     |  FF   |             |             |       |
+-------------+-------------+-------+-------------+-------------+-------+-------------+-------------+-------+
|Pointer Reg. |    02A2     |  09   |CPU Page Reg.|    02A5     |  07   |             |             |       |
+-------------+-------------+-------+-------------+-------------+-------+-------------+-------------+-------+
|                                                                                                         |
|                              DATA REGISTER Values Indexed By POINTER REGISTER                             |
|                                                                                                         |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
| POINTER |         REGISTER ACCESSED         | VALUE | POINTER |         REGISTER ACCESSED         | VALUE |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
|   08    | Interrupt Level                   |  02   |   0C    | Degate Compare 0                  |  10   |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
|   09    | Interrupt Co-Processor            |  00   |   0D    | Degate Compare 1                  |  E0   |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
|   0A    | Parity Address Low                |  00   |   0E    | Degate Compare 2                  |  0F   |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
|   0B    | Parity Address High And Status    |  00   |   0F    | Gate Array ID (SSTIC)             |  C0   |
+---------+-----------------------------------+-------+---------+-----------------------------------+-------+
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                    12:09:25    Fri Feb 22, 1991    PAGE 6
Dump Information

                    8030 SCC 00 REGISTER VALUES

   80186       REGISTER   VALUE    80186       REGISTER   VALUE
 I/O ADDRESS     NAME            I/O ADDRESS     NAME

    0100       RR0B.....   54        0120      RR0A.....   54
    0102       RR1B        07        0122      RR1A        07
    0104       RR2B.....   26        0124      RR2A.....   20
    0106       RR3B        00        0126      RR3A        00
    0110       RR8B.....   74        0130      RR8A.....   74
    0114       RR10B       00        0134      RR10A       00
    0118       RR12B....   0A        0138      RR12A....   0A
    011A       RR13B       00        013A      RR13A       00
    011E       RR15B....   C0        013E      RR15A....   C0
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                    12:09:25    Fri Feb 22, 1991    PAGE 7
Dump Information

                    8030 SCC 01 REGISTER VALUES

   80186       REGISTER   VALUE    80186       REGISTER   VALUE
 I/O ADDRESS     NAME            I/O ADDRESS     NAME

    0400       RR0B.....   44        0420      RR0A.....   44
    0402       RR1B        07        0422      RR1A        07
    0404       RR2B.....   86        0424      RR2A.....   80
    0406       RR3B        00        0426      RR3A        00
    0410       RR8B.....   30        0430      RR8A.....   30
    0414       RR10B       00        0434      RR10A       00
    0418       RR12B....   0A        0438      RR12A....   0A
    041A       RR13B       00        043A      RR13A       00
    041E       RR15B....   C0        043E      RR15A....   C0
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                    12:09:25    Fri Feb 22, 1991    PAGE 8
Dump Information

                    8030 SCC 02 REGISTER VALUES

   80186       REGISTER   VALUE    80186       REGISTER   VALUE
 I/O ADDRESS     NAME            I/O ADDRESS     NAME

    0600       RR0B.....   5C        0620      RR0A.....   5C
    0602       RR1B        07        0622      RR1A        07
    0604       RR2B.....   96        0624      RR2A.....   90
    0606       RR3B        00        0626      RR3A        00
    0610       RR8B.....   30        0630      RR8A.....   30
    0614       RR10B       40        0634      RR10A       00
    0618       RR12B....   0A        0638      RR12A....   0A
    061A       RR13B       00        063A      RR13A       00
    061E       RR15B....   C0        063E      RR15A....   C0
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                    12:09:25    Fri Feb 22, 1991    PAGE 9
Dump Information

                    8030 SCC 03 REGISTER VALUES

   80186       REGISTER   VALUE    80186       REGISTER   VALUE
 I/O ADDRESS     NAME            I/O ADDRESS     NAME

    0700       RR0B.....   5C        0720      RR0A.....   54
    0702       RR1B        07        0722      RR1A        07
    0704       RR2B.....   E6        0724      RR2A.....   E0
    0706       RR3B        00        0726      RR3A        00
    0710       RR8B.....   30        0730      RR8A.....   30
    0714       RR10B       40        0734      RR10A       40
    0718       RR12B....   0A        0738      RR12A....   0A
    071A       RR13B       00        073A      RR13A       00
    071E       RR15B....   C0        073E      RR15A....   C0
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                        12:09:25    Fri Feb 22, 1991    PAGE 10
Dump Information

  80186       8036 CIO 0 PORT/REGISTER NAME       VALUE       80186       8036 CIO 0 PORT/REGISTER NAME       VALUE
 I/O ADDRESS                                                 I/O ADDRESS

    0180     Master Interrupt Control...........   9C          01B0     Counter/Timer 2 Time Constant MSB..   FF
    0182     Master Configuration Control          70          01B2     Counter/Timer 2 Time Constant LSB     FF
    0184     Port A Interrupt Vector............   30          01B4     Counter/Timer 3 Time Constant MSB..   FF
    0186     Port B Interrupt Vector               40          01B6     Counter/Timer 3 Time Constant LSB     FF
    0188     Counter/Timer Interrupt Vector.....   56          01B8     Counter/Timer 1 Mode Specification.   05
    018A     Port C Data Path Polarity             FF          01BA     Counter/Timer 2 Mode Specification    05
    018C     Port C Data Direction..............   FA          01BC     Counter/Timer 3 Mode Specification.   25
    018E     Port C Special I/O Control            F0          01BE     Current Vector                        FF
    0190     Port A Command and Status..........   00          01C0     Port A Mode Specification..........   06
    0192     Port B Command and Status             00          01C2     Port A Handshake Specification        00
    0194     Counter/Timer 1 Command and Status.   00          01C4     Port A Data Path Polarity..........   FF
    0196     Counter/Timer 2 Command and Status    00          01C6     Port A Data Direction                 3B
    0198     Counter/Timer 3 Command and Status.   00          01C8     Port A Special I/O Control.........   00
    019A     Port A Data                           2A          01CA     Port A Pattern Polarity               00
    019C     Port B Data........................   2A          01CC     Port A Pattern Transition..........   00
    019E     Port C Data                           FB          01CE     Port A Pattern Mask                   00
    01A0     Counter/Timer 1 Current Count MSB..   06          01D0     Port B Mode Specification..........   06
    01A2     Counter/Timer 1 Current Count LSB     30          01D2     Port B Handshake Specification        00
    01A4     Counter/Timer 2 Current Count MSB..   00          01D4     Port B Data Path Polarity..........   FF
    01A6     Counter/Timer 2 Current Count LSB     00          01D6     Port B Data Direction                 3B
    01A8     Counter/Timer 3 Current Count MSB..   00          01D8     Port B Special I/O Control.........   00
    01AA     Counter/Timer 3 Current Count LSB     00          01DA     Port B Pattern Polarity               00
    01AC     Counter/Timer 1 Time Constant MSB..   FF          01DC     Port B Pattern Transition..........   00
    01AE     Counter/Timer 1 Time Constant LSB     FF          01DE     Port B Pattern Mask                   00
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                        12:09:25    Fri Feb 22, 1991    PAGE 11
Dump Information

  80186       8036 CIO 1 PORT/REGISTER NAME       VALUE       80186       8036 CIO 1 PORT/REGISTER NAME       VALUE
 I/O ADDRESS                                                 I/O ADDRESS

    0500     Master Interrupt Control...........   84          0530     Counter/Timer 2 Time Constant MSB..   FF
    0502     Master Configuration Control          70          0532     Counter/Timer 2 Time Constant LSB     FF
    0504     Port A Interrupt Vector............   F0          0534     Counter/Timer 3 Time Constant MSB..   00
    0506     Port B Interrupt Vector               F8          0536     Counter/Timer 3 Time Constant LSB     01
    0508     Counter/Timer Interrupt Vector.....   F6          0538     Counter/Timer 1 Mode Specification.   05
    050A     Port C Data Path Polarity             F0          053A     Counter/Timer 2 Mode Specification    05
    050C     Port C Data Direction..............   F0          053C     Counter/Timer 3 Mode Specification.   00
    050E     Port C Special I/O Control            F0          053E     Current Vector                        FF
    0510     Port A Command and Status..........   00          0540     Port A Mode Specification..........   06
    0512     Port B Command and Status             08          0542     Port A Handshake Specification        00
    0514     Counter/Timer 1 Command and Status.   00          0544     Port A Data Path Polarity..........   FF
    0516     Counter/Timer 2 Command and Status    00          0546     Port A Data Direction                 FF
    0518     Counter/Timer 3 Command and Status.   00          0548     Port A Special I/O Control.........   00
    051A     Port A Data                           FC          054A     Port A Pattern Polarity               00
    051C     Port B Data........................   00          054C     Port A Pattern Transition..........   00
    051E     Port C Data                           F0          054E     Port A Pattern Mask                   00
    0520     Counter/Timer 1 Current Count MSB..   05          0550     Port B Mode Specification..........   00
    0522     Counter/Timer 1 Current Count LSB     FB          0552     Port B Handshake Specification        00
    0524     Counter/Timer 2 Current Count MSB..   00          0554     Port B Data Path Polarity..........   FF
    0526     Counter/Timer 2 Current Count LSB     00          0556     Port B Data Direction                 C0
    0528     Counter/Timer 3 Current Count MSB..   00          0558     Port B Special I/O Control.........   00
    052A     Counter/Timer 3 Current Count LSB     00          055A     Port B Pattern Polarity               00
    052C     Counter/Timer 1 Time Constant MSB..   FF          055C     Port B Pattern Transition..........   00
    052E     Counter/Timer 1 Time Constant LSB     FF          055E     Port B Pattern Mask                   00
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                        12:09:25    Fri Feb 22, 1991    PAGE 12
Dump Information

                                       CO-PROCESSOR ADAPTER FREE MEMORY LIST

 (START,+SIZE)  (START,+SIZE)  (START,+SIZE)  (START,+SIZE)  (START,+SIZE)  (START,+SIZE)  (START,+SIZE)  (START,+SIZE)

 (0090, +0170)  (025B, +6CFF)  (    ,  +   )  (    ,  +   )  (    ,  +   )  (    ,  +   )  (    ,  +   )  (    ,  +   )
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                        12:09:25    Fri Feb 22, 1991    PAGE 13
Dump Information

                                              TASK-RELATED INFORMATION FOR TASK 00

 TASK HEADER BEGINS AT 7A91:0000 = 3D:0910 (8K PG) = 7A910                                TASK IN MEMORY FROM 7A910 TO 7FFFC

                                                          TASK HEADER

Task ID = 0002          |Module Length = 000056EC Bytes|      STACK       | COMMAND VECTOR   |  INITIAL ENTRY    |  DATA SEGMENT
                                                       |Seg.  Off.        |Seg.  Off.        |Seg.  Off.         |Seg.  Off.
 Task |Priority|Debug|Extension|Resource Request|Sixth|7A91:56EC         |7A91:0000         |7A91:0380          |7A91:0000
Number|        |Flag | Offset  | Block Pointer  |Byte |  3F:1FFC (8K PG) |  3D:0910 (8K PG) |  3D:0C90  (8K PG) |  3D:0910 (8K PG)
  00  |   00   | 00  |  0000   |     0000       | 00  |    7FFFC         |    7A910         |    7AC90          |    7A910

                                                    RESOURCES OWNED BY TASK

 SCC/CIO's OWNED:                                                                  | CPU DMA's OWNED:

 CIO Timer's OWNED:

 SCC's OWNED:


 CIO's OWNED:

 RS232's OWNED:                                                | RS422's OWNED:
 V.35's  OWNED:                                                | EIA-530 OWNED:
 X.21's  OWNED:                                                | OTHER        :

                                                    SOFTWARE TIMERS OWNED:


                                                        VECTORS OWNED:


                                                         QUEUES OWNED:


                                                      MEMORY BLOCKS OWNED

 (START,+SIZE) | (START,+SIZE) | (START,+SIZE) | (START,+SIZE) | (START,+SIZE) | (START,+SIZE) | (START,+SIZE) | (START,+SIZE)
 (     , +    )| (     , +    )| (     , +    )| (     , +    )| (     , +    )| (     , +    )| (     , +    )| (     , +    )

                                                       SEMAPHORES OWNED

   HANDLE     COUNT  |   HANDLE     COUNT  |   HANDLE     COUNT  |   HANDLE     COUNT  |   HANDLE     COUNT  |   HANDLE     COUNT


                                                    NAMING RESOURCES OWNED

   DEVICE TYPE     | DEVICE NO. | NAME POINTER |     NAME      |   DEVICE TYPE     | DEVICE NO. | NAME POINTER |     NAME


                                                 EXTENDED MEMORY BLOCKS OWNED

CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                        12:09:25    Fri Feb 22, 1991    PAGE 14
Dump Information

 HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES  | HANDLE,PAGES
 (     ,      )| (     ,      )| (     ,      )| (     ,      )| (     ,      )| (     ,      )| (     ,      )| (     ,      )
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                         12:09:25    Fri Feb 22, 1991    PAGE 15
Dump Information

                                             TASK-RELATED INFORMATION FOR TASK 02

  TASK HEADER BEGINS AT 764C:0000 = 3B:04C0 (8K PG) = 764C0                                  TASK IN MEMORY FROM 764C0 TO 78CDF

                                                           TASK HEADER

 Task ID = 5053          Module Length = 0000281F Bytes      STACK        COMMAND VECTOR     INITIAL ENTRY       DATA SEGMENT
                                                        Seg.  Off.        Seg.  Off.        Seg.  Off.          Seg.  Off.
  Task  Priority Debug Extension Resource Request Sixth 764C:281F         764C:07D0         764C:031F           764C:0000
 Number          Flag   Offset    Block Pointer   Byte    3C:0CDF (8K PG)   3B:0C90 (8K PG)   3B:07DF  (8K PG)    3B:04C0 (8K PG)
  02        02    00     0000          0086        00       78CDF             76C90             767DF               764C0

                                                     RESOURCES OWNED BY TASK

  SCC/CIO's OWNED:                                                                    CPU DMA's OWNED:

  CIO Timer's OWNED:

  SCC's OWNED:



  CIO's OWNED:

  RS232's OWNED:                                                  RS422's OWNED:
  V.35's  OWNED:                                                  EIA-530 OWNED:
  X.21's  OWNED:                                                  OTHER        :

                                                     SOFTWARE TIMERS OWNED:
  20

                                                         VECTORS OWNED:


                                                          QUEUES OWNED:


                                                       MEMORY BLOCKS OWNED

  (START,+SIZE)   (START,+SIZE)   (START,+SIZE)   (START,+SIZE)   (START,+SIZE)   (START,+SIZE)   (START,+SIZE)   (START,+SIZE)
  (0200, +005B)   (    , +    )   (    , +    )   (    , +    )   (    , +    )   (    , +    )   (    , +    )   (    , +    )

                                                       SEMAPHORES OWNED

    HANDLE    COUNT       HANDLE    COUNT       HANDLE    COUNT       HANDLE    COUNT       HANDLE    COUNT      HANDLE    COUNT


                                                    NAMING RESOURCES OWNED

    DEVICE TYPE      DEVICE NO.  NAME POINTER        NAME           DEVICE TYPE      DEVICE NO.  NAME POINTER        NAME


                                                EXTENDED MEMORY BLOCKS OWNED

CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                         12:09:25    Fri Feb 22, 1991    PAGE 16
Dump Information

  HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES    HANDLE,PAGES
  (    ,      )   (    ,      )   (    ,      )   (    ,      )   (    ,      )   (    ,      )   (    ,      )   (    ,      )
```

```
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                12:09:25    Fri Feb 22, 1991    PAGE 17
Dump Information
┌───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│                                                                                                                       │
│                                        TASK-RELATED INFORMATION FOR TASK 03                                           │
│                                                                                                                       │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ TASK HEADER BEGINS AT 6F5A:0000 = 37:15A0 (8K PG) = 6F5A0                           TASK IN MEMORY FROM 6F5A0 TO 764BD │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                                  TASK HEADER                                                          │
├──────────────────────┬───────────────────────────────┬────────────────┬────────────────┬────────────────┬─────────────┤
│Task ID = 0000        │Module Length = 00006F1D Bytes │     STACK      │ COMMAND VECTOR │ INITIAL ENTRY  │ DATA SEGMENT│
├──────┬────────┬─────┬┴────────┬───────────────┬──────┤Seg.  Off.      │Seg.  Off.      │Seg.  Off.      │Seg.  Off.   │
│ Task │Priority│Debug│Extension│Resource Request│Sixth│6F5A:6F1D       │6F5A:3255       │6F5A:177C       │6F5A:0000    │
│Number│        │Flag │ Offset  │ Block Pointer │Byte  │  3B:04BD (8K PG)│  39:07F5 (8K PG)│  38:0D1C  (8K PG)│  37:15A0 (8K PG)│
│  03  │   03   │ 00  │  0000   │     0168      │ 00   │    764BD       │    727F5       │    70D1C       │   6F5A0     │
├──────┴────────┴─────┴─────────┴───────────────┴──────┴────────────────┴────────────────┴────────────────┴─────────────┤
│                                          RESOURCES OWNED BY TASK                                                      │
├───────────────────────────────────────────────────────────────────────────┬───────────────────────────────────────────┤
│ SCC/CIO's OWNED:                                                          │ CPU DMA's OWNED:                          │
├───────────────────────────────────────────────────────────────────────────┴───────────────────────────────────────────┤
│ CIO Timer's OWNED:                                                                                                    │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ SCC's OWNED:                                                                                                          │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                                                                                                       │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ CIO's OWNED:                                                                                                          │
├─────────────────────────────────────────────────────────┬─────────────────────────────────────────────────────────────┤
│ RS232's OWNED:                                          │ RS422's OWNED:                                              │
│ V.35's  OWNED:                                          │ EIA-530 OWNED:                                              │
│ X.21's  OWNED:                                          │ OTHER        :                                              │
├─────────────────────────────────────────────────────────┴─────────────────────────────────────────────────────────────┤
│                                          SOFTWARE TIMERS OWNED:                                                       │
│ 00                                                                                                                    │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                              VECTORS OWNED:                                                           │
│ 75  71                                                                                                                │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                              QUEUES OWNED:                                                            │
│                                                                                                                       │
├───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                            MEMORY BLOCKS OWNED                                                        │
├──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┤
│(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │(START,+SIZE) │
│(     , +   ) │(     , +   ) │(     , +   ) │(     , +   ) │(     , +   ) │(     , +   ) │(     , +   ) │(     , +   ) │
├──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┤
│                                            SEMAPHORES OWNED                                                           │
├──────────────────┬───────────────────┬───────────────────┬───────────────────┬───────────────────┬────────────────────┤
│  HANDLE    COUNT │    HANDLE    COUNT│    HANDLE    COUNT│    HANDLE    COUNT│    HANDLE    COUNT│   HANDLE    COUNT  │
│                  │                   │                   │                   │                   │                    │
├──────────────────┴───────────────────┴───────────────────┴───────────────────┴───────────────────┴────────────────────┤
│                                          NAMING RESOURCES OWNED                                                       │
├────────────────┬──────────┬────────────┬────────────────┬────────────────┬──────────┬────────────┬────────────────────┤
│  DEVICE TYPE   │DEVICE NO.│NAME POINTER│     NAME       │  DEVICE TYPE   │DEVICE NO.│NAME POINTER│        NAME        │
│                │          │            │                │                │          │            │                    │
├────────────────┴──────────┴────────────┴────────────────┴────────────────┴──────────┴────────────┴────────────────────┤
│                                        EXTENDED MEMORY BLOCKS OWNED                                                   │
├──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┬──────────────┤
CO-PROCESSOR ADAPTER 0 SYSTEM INFORMATION DUMP                                12:09:25    Fri Feb 22, 1991    PAGE 18
Dump Information

│ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │ HANDLE,PAGES │
│ (    ,     ) │ (    ,     ) │ (    ,     ) │ (    ,     ) │ (    ,     ) │ (    ,     ) │ (    ,     ) │ (    ,     ) │
└──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┴──────────────┘
```

Appendix B. Include Files

This appendix describes the Device Driver include file and the C Language interface include file.

Device Driver Include File

The include file **icaioctl.h** for the Co-Processor AIX Support device driver functions contains the necessary declarations for applications using the device driver. Include **icaioctl.h** in your application program by putting the following line at the beginning of your application program:

```
#include <icaioctl.h>
```

The include file consists of these major sections:

- Parameter structure declarations for each Co-Processor AIX Support device driver function and a union of all such structures
- Function code definitions for the Co-Processor AIX Support device driver
- Error code definitions
- Definitions that allow easier access by system unit applications to common data structures used by the Realtime Control Microcode and other co-processor adapter tasks
- Miscellaneous definitions of commonly used constants

C Language Interface Include File

The C Language Interface include file **icaclib.h** contains the necessary declarations for applications using the Co-Processor AIX Support device driver and the C Language Interface Library. Include **icaclib.h** in your application program by putting the following line at the beginning of your application program:

```
#include <icaclib.h>
```

The include file consists of these major sections:

- Function declarations for C Language Interface Library Routines
- Error code definitions
- Data structure definitions for C Language Interface Library Routines
- Definitions that allow easier access by system unit applications to common data structures used by the Realtime Control Microcode and other co-processor adapter tasks
- Miscellaneous definitions of commonly used constants

Appendix C. Return Codes

This appendix explains the return codes for the device driver routines, the C Language Interface routines, the application loader, the online dump facility, and the dump formatter facility.

Device Driver Return Codes

The following messages may be returned for the Co-Processor AIX Support device driver routines and the C Language Interface routines, unless noted otherwise:

0x0000 NO_ERROR

The requested driver function was completed successfully.

0xFF05 E_ICA_INVALID_COPROC

The co-processor adapter number is out of range, or the referenced adapter is not installed.

0xFF06 E_ICA_INVALID_TASK_STATUS

The task referenced is not loaded, or there is an error indicated in the task's primary status byte. This error code is also returned if the Co-Processor AIX Support device driver routine ICAISSUECMD is called with parameters and the referenced task's output buffer is busy.

0xFF07 E_ICA_INVALID_PAGE

The page number passed to the device driver is out of range. The maximum allowable page number depends on the amount of memory installed on the co-processor adapter. Following are the maximum page values based on the adapter type and selected window size.

	Window Size			
Memory on Adapter	**8 KB**	**16 KB**	**32 KB**	**64 KB**
Adapters with 512 KB	0x3F	0x1F	0x0F	0x07
Adapters with 1 MB except PortMaster/A	0x77	0x3B	0x1D	0x0E
Portmaster/A with 512 KB	0x7F	0x3F	0x1F	0x07
Portmaster/A with 1 MB	0x7F	0x3F	0x1F	0x0F
Portmaster/A with 2 MB	0xFF	0x7F	0x3F	0x1F

For a discussion of changing the shared storage window size, see Appendix F, "Changing the Shared Storage Window Size."

0xFF08 E_ICA_INVALID_OFFSET

The page offset is out of range. For an 8 KB page, the valid page offsets are in the range 0x0 through 0x1FFFF. For a 16 KB page, the valid numbers are 0x0 through 0x3FFFF. For a 32 KB page, the range is 0x0 through 0x7FFFF. For a 64 KB page, all page offsets are valid.

0xFF09 E_ICA_INVALID_FORMAT

The address format byte had a value other than 0x00, 0x01, or 0xFF.

0xFF0B E_ICA_TIMEOUT

The call timed out and may have failed. If this code was returned after issuing a command, the Realtime Control Microcode did not respond to the command in time. If it was returned after an Interrupt Wait or Special Events Wait call, the task did not interrupt, or the Realtime Control Microcode did not receive an Initialize command within the specified time. If it was returned after issuing a reset, the PROM did not complete initialization successfully.

0xFF0D E_ICA_INVALID_CONTROL

The control flag used when registering to be notified of the Realtime Control Microcode's receipt of an Initialize command was not valid.

0xFF11 E_ICA_BAD_PCSELECT

The command could not be issued because the PC select byte in the interface block was not valid. See the *Realtime Interface Co-Processor Firmware Technical Reference* for more information on the PC select byte.

0xFF12 E_ICA_CMD_REJECTED

The Realtime Control Microcode returned an error after a command was issued.

0xFF13 E_ICA_NO_CMD_RESPONSE

The co-processor did not respond after a command was issued.

0xFF14 E_ICA_OB_SIZE

The number of bytes in the parameter buffer was greater than the size of the task's output buffer when issuing a command. The command is aborted.

0xFF15 E_ICA_ALREADY_REG

An attempt was made to register for a task interrupt or for notification of the Realtime Control Microcode's receipt of an Initialize command, but the application process has already done so.

0xFF17 E_ICA_NOT_REG

An attempt was made to wait for a task interrupt or for notification of the Realtime Control Microcode's receipt of an Initialize command, but the application process is not registered with the device driver.

0xFF23 E_ICA_INVALID_FD

The ioctl returned -1. Check the errno value to get the error value from the ioctl.

0xFF25 E_ICA_XMALLOC_FAIL

The device driver was unable to allocate memory for use in the kernel.

0xFF26 E_ICA_ALREADY_OPEN

An attempt was made to open the device driver a second time in the context of a single process. Only one open is allowed per process.

0xFF27 E_ICA_INVALID_TASK

The task number passed to the device driver was out of range. The task number must be greater than or equal to 0, and less than or equal to the value for MAXTASK specified in the application loader parameter file (**icaparm.prm**). The only exception to this is task number 0xFE, which is used to access the error task, and does not return this error.

0xFF28 E_ICA_INTR

The system call was interrupted by a signal.

OxFF2C E_ICA_NOMEM

The device driver was unable to pin real memory. Install more memory SIMMs or memory adapters, or decrease the system's use of real memory.

0xFF30 E_ICA_NO_MORE_RES

The device driver was unable to allocate a system timer from the AIX operating system.

0xFF31 E_ICA_BAD_OPEN_HANDLE

This is returned by the device driver when a child user of the device driver handle calls the **Interrupt Register, Interrupt Deregister, Interrupt Wait,** or **Issue Command** function. These functions can be called only by the process opening the device driver directly. The child process needs to do an open call before issuing any of the above mentioned functions.

0xFF33 E_ICA_RESET_OPEN_HANDLE

The reset of the adapter failed because the PROM could not complete initialization and POST successfully. Run diagnostics to determine the cause of the problem.

Application Loader Utility Return Codes

The following are the application loader return codes that correspond to the Application Loader Utility messages listed in “Application Loader Utility Information Messages” on page D-1.

00 **Normal Termination**.

The application loader loaded the requested task. No errors were found.

01 **Parameter file open error.**[1]

The loader was unable to open the parameter file.

02 **Parameter file read error.**[1]

The loader was unable to read the parameter file.

03 **Parameter file close error.**[1]

The loader was unable to close the parameter file.

04 **Invalid parameter file entry.**[1]

One or more parameter file entries are not in the correct format.

06 **Error opening task, driver, or message file.**

The loader was unable to open the task file, the loader message file, or the device entry **/dev/artic**.

07 **Error reading task or messages file.**

The loader was unable to read a task file or the loader message file.

08 **Error closing task or messages file.**

The loader was unable to close a task file or the loader message file.

09 **Illegal flag.**

An illegal flag was entered on the loader command line.

10 **Invalid task number.**

The specified task number is greater than the MAXTASK value specified in the parameter file (or the default value if a parameter file is not used).

12 **Task 0 already loaded.**[1]

The Realtime Control Microcode is already loaded on the specified co-processor adapter.

[1] This message can be returned only when Realtime Control Microcode is being loaded.

13 Task 0 status invalid.

The error bit in the Realtime Control Microcode's primary status byte is set.

14 Task 0 not loaded and initialized.

An attempt was made to load a task before the Realtime Control Microcode was loaded on the specified co-processor adapter.

15 Task already loaded.

The specified task is already loaded.

17 Task 0 output buffer size invalid.

The output buffer length field in the Realtime Control Microcode’s Buffer Control Block has been overwritten and is invalid.

18 Command not accepted.

The Realtime Control Microcode has rejected a command because it RAM-resident code and/or data has been inadvertently modified.

19 Cannot start task - task not loaded.

The specified task was not correctly assigned the "loaded" state in its primary status byte following the Load Task command. This indicates that the task’s RAM-resident code and/or data have been modified.

20 File relocation error.

An error occurred while the application loader attempted to relocate a task on the co-processor.

21 No device response.

The loader did not receive an interrupt from the co-processor adapter in the allocated time. This situation could be caused by a software error on the system unit or by a co-processor adapter hardware error.

22 Invalid PC select byte.

The command could not be issued because the PC select byte in the task interface block was invalid. This signals that there was an error in previous communication between the system unit and the co-processor adapter or this storage area was inadvertently overwritten.

23 ARTIC device driver is not installed.

The Co-Processor AIX Support device driver is not installed so the loader cannot execute. The device entry **/dev/artic** does not exist.

25 **Invalid co-processor adapter number.**

The specified adapter was not initialized by the device driver and is not recognized as being installed.

26 **The device driver returned an error code.**

A catch-all message for error codes returned to the loader by the device driver which are not addressed by any of the other loader error messages.

27 **Invalid or missing command line argument(s):**

The command line was found to be in error.

31 **Driver ioctl error on specified adapter.**

A device driver ioctl failed on the adapter.

Online Dump Facility Return Codes

| The following are the online dump facility return codes that correspond to the Online Dump Facility messages in “Online Dump Facility Information Messages” on page D-6:

0 **No error**.

The dump completed successfully.

01 **Invalid co-processor adapter number.**

Adapter was not initialized by the device driver and is not recognized as being installed.

02 **Co-processor already enabled for AutoDump.**

An attempt was made to enable AutoDump on a co-processor adapter that has already been enabled for AutoDump.

03 **Co-processor not enabled for AutoDump.**

An attempt was made to disable AutoDump on a co-processor adapter that has not been enabled for AutoDump.

04 **Illegal flag**.

An illegal flag was entered on the command line.

05 **Cannot access directory.**

The directory (specified with the -dd flag) does not exist or cannot be accessed.

06 **Unable to perform dump. Coproc not responding.**

The co-processor adapter is not responding to commands from the Online Dump Facility.

07 **AutoDump not enabled on coproc. Coproc not responding.**

The co-processor adapter is not responding to commands from the Online Dump Facility.

08 **Dump data will not fit on file system. Dump of coproc canceled.**

The Online Dump Facility detected that the file system where the dump data is to be stored does not have enough free space.

09 **AutoDump data will not fit on file system. AutoDump of coproc canceled.**

The Online Dump Facility detected that the file system where the dump data is to be stored does not have enough free space.

10 **ARTIC device driver is not installed.**

The Co-Processor AIX Support device driver is not installed, so the Online Dump Facility cannot execute.

11 **Error opening a file.**

The Online Dump Facility encountered an error while attempting to open a file.

12 **Error reading a file.**

The Online Dump Facility encountered an error while attempting to read a file.

13 **Error closing a file.**

The Online Dump Facility encountered an error while attempting to close a file.

14 **Cannot create AutoDump tag file.**

The Online Dump Facility could not create the AutoDump tag file.

15 **Cannot allocate memory for dump.**

The Online Dump Facility could not allocate enough memory for the dump.

16 **Error writing to system or memory dump files.**

The Online Dump Facility could not write to the System or Memory dump files.

17 **Device Driver error.**

The Online Dump Facility encountered an error during a device driver call.

Dump Formatter Facility Return Codes

The following codes are returned by the Dump Formatter Facility and correspond to the error messages in “Online Dump Facility Error Messages” on page D-6.

00 No errors.

01 **Invalid Co-Processor adapter specified**

The co-processor adapter number does not fit in the range 0–7 of valid co-processor adapter numbers.

02 **Cannot access file xxxxxxxx.xxx**

The file **xxxxxxxx.xxx** could not be accessed by the Dump Formatter Facility.

The dump files **ICAME*N*.DMP** and/or **ICASYS*N*.DMP** could not be opened by the Dump Formatter Facility.

03 **Illegal command option(s)**

An illegal option was entered on the command line.

Appendix D. Messages

The messages in this appendix are returned by the Application Loader Utility, the Online Dump Facility, and the Dump Formatter Facility.

Application Loader Utility Information Messages

Normal Termination. Task yy loaded on coproc xx.

Explanation: The application loader loaded a task onto co-processor xx as task number yy. No errors were found.

Action: None

Application Loader Utility Error Messages

ICALDR01E: Parameter file *filename* open error. Error code = nnnn.[1]

Explanation: The loader was unable to open the parameter file *filename*. The system unit error code returned by AIX is **nnnn**.

Action: Check that the file name is correctly spelled. Check the file **/usr/include/sys/errno.h** to help resolve the error.

ICALDR02E: Parameter file *filename* read error. Error code = nnnn.[1]

Explanation: The loader was unable to read the parameter file **filename**. The system unit error code returned by AIX is **nnnn**.

Action: Check that you have read file permission. Check the file **/usr/include/sys/errno.h** to help resolve the error.

ICALDR03E: Parameter file *filename* close error. Error code = nnnn.[1]

Explanation: The loader was unable to close the parameter file **filename**. The errno code returned by AIX is **nnnn**.

Action: Check the file **/usr/include/sys/errno.h** to help resolve the error.

ICALDR04E: Invalid parameter file entry.[1]

Explanation: One or more parameter file entries are not in the correct format.

Action: Check the format of the parameter file. Correct the file as required.

[1] This message can be returned only when Realtime Control Microcode is being loaded.

ICALDR06E: Error opening *filename*. Return code = nnnn.

Explanation: The loader was unable to open **filename** which is a task file, the loader message file, or the device entry **/dev/artic**. Check the file **/usr/include/sys/errno.h** to help resolve the error.

| Action: Check that the file name is correctly spelled. Check the file
| **/usr/include/errno.h** to help resolve the error.

ICALDR07E: Error reading *filename*. Return code = nnnn.

Explanation: The loader was unable to read **filename** which is a task file or the loader message file. nnnn is the errno code returned by AIX.

| Action: Check that you have read file permission. Check the file
| **/usr/include/sys/errno.h** to help resolve the error.

ICALDR08E: Error closing *filename*. Return code = nnnn.

Explanation: The loader was unable to close **filename** which is a task file, or the loader message file. The errno code returned by AIX is **nnnn**.

| Action: Check the file **/usr/include/sys/errno.h** to help resolve the error.

ICALDR09E: Illegal flag "-xxx".

Explanation: An illegal flag (**-xxx**) was entered on the loader command line.

Action: Correct the error and retry the command.

ICALDR10E: Invalid task number. Task number = nn.

Explanation: The specified task number, nn, is greater than the MAXTASK value specified in the parameter file (or the default value if a parameter file is not used).

Action: Correct the task number and retry the command.

ICALDR12E: Task 0 already loaded. Status = nn.[1]

Explanation: The Realtime Control Microcode is already loaded on the specified co-processor adapter. Task 0's primary status byte is **nn**.

Action: If the Realtime Control Microcode appears to be functioning properly, do not reload it. If reload of the Realtime Control Microcode is required, use the **-reset** application loader option.

ICALDR13E: Task 0 invalid status. Status = nn.

Explanation: The error bit in the Realtime Control Microcode's primary status byte is set. Task 0's primary status byte is **nn**.

Action: If a dump has not occurred, you can use the Online Dump Facility to dump the adapter to attempt to locate the cause of the error. Reload the Realtime Control Microcode on the failed co-processor adapter using the **-reset** application loader option.

ICALDR14E: Task 0 not loaded and initialized. Status = nn.

Explanation: An attempt was made to load a task before the Realtime Control Microcode was loaded on the specified co-processor adapter. Task 0's primary status byte is **nn**.

Action: Load the Realtime Control Microcode.

ICALDR15E: Task already loaded. Status = nn.

Explanation: The specified task is already loaded. The previously loaded task's primary status byte is **nn**.

Action: Ensure that the task number is correct.

ICALDR17E: Task 0 output buffer size invalid. Status = nn.

Explanation: The output buffer length field in the Realtime Control Microcode's Buffer Control Block has been overwritten and is invalid. Task 0's primary status byte is **nn**.

Action: You can use the Online Dump Facility to dump the co-processor adapter to attempt to find the cause of the error. Reload the Realtime Control Microcode using the **-reset** application loader option.

ICALDR18E: Command not accepted. Status = nn.

Explanation: The Realtime Control Microcode has rejected a command because its RAM-resident code and/or data has been inadvertently modified. Task 0's primary status byte is **nn**.

Action: Use the Online Dump Facility to dump the co-processor adapter. Reload the Realtime Control Microcode using the **-reset** application loader option.

ICALDR19E: Cannot start task - task not loaded.

Explanation: The specified task was not correctly assigned the *loaded* state in its primary status byte following the Load Task command. The task's RAM-resident code and/or data have been modified.

Action: Use the Online Dump Facility to dump the adapter. Reload the Realtime Control Microcode using the **-reset** application loader option. Retry the command to load the task.

ICALDR20E: File relocation error.

Explanation: An error occurred while the application loader attempted to relocate a task on the co-processor.

Action: Verify that the task file is of proper format, either **.com** or **.exe**.

ICALDR21E: No device response. Coproc = nn. Status = nn.

Explanation: The loader did not receive an interrupt from co-processor **nn** in the allocated time. This situation could be caused by a software error on the system unit or by a co-processor adapter hardware error.

Action: Run diagnostics on the failing adapter.

ICALDR22E: Invalid PC Select Byte. PC Select = nn

Explanation: The command could not be issued because the PC select byte in the task interface block was invalid. This signals that there was an error in some previous communication between the system unit and the co-processor adapter, or that this area of storage was inadvertently overwritten.

Action: First, isolate the error. Then issue a hardware reset to the co-processor adapter to clear the condition.

ICALDR23E: Co-Processor AIX Support device driver is not installed.

Explanation: The Co-Processor AIX Support device driver is not installed so the application loader cannot execute. The device entry **/dev/artic** does not exist.

Action: Run the co-processor adapter configuration program to determine if the driver is installed. Install the Co-Processor AIX Support driver if required.

ICALDR25E: Invalid co-processor adapter number nnnn.

Explanation: Adapter number **nnnn** was not initialized by the device driver and is not recognized as being installed.

Action: Verify that the adapter number is correct. Run the adapter configuration program to determine if the driver is installed. Install the driver if required.

ICALDR26E: The device driver returned error code nn.

Explanation: A catch-all message for error codes returned to the loader by the device driver which are not addressed by any of the other loader error messages. The error code returned by the device driver is **nnnn**.

Action: Check "Application Loader Utility Return Codes" on page C-4 for the source of the problem.

ICALDR27E: Invalid or missing command line argument(s):

Explanation: The startup command line was found to be in error. The correct usage format is displayed.

Usage: icaldric cardnumber filename tasknumber [-pf filename] [-ns] [-l] [-m boundary] [-q] [-prm string]

Action: Correct the command and retry.

ICALDR31E: Device driver ioctl error on "devname". Error code = nnnn.

Explanation: A device driver function failed on the device **devname**. Check the file **/usr/include/errno.h** to help resolve the error.

Action: Check “Application Loader Utility Return Codes” on page C-4 for the source of the problem.

ICALDR32E: PriStatus: nn. SecStatus: nn nn nn nn nn nn ...

Explanation: This message is displayed following error messages ICALDR19E, ICALDR21E, and ICALDR33E - no response, cannot start task, and task not initialized. The primary and secondary status bytes are those of Task 0.

Action: Decode the status bytes to aid in problem determination.

Online Dump Facility Information Messages

ICADPR30I: Dump Completed.

Explanation: The Online Dump Facility completed dumping a co-processor adapter's memory and I/O ports to disk.

Action: The output files can be formatted using the Dump Formatter Facility (described in Chapter 7, “Dump Formatter Facility”).

ICADPR31I: Writing dump data....

Explanation: The Online Dump Facility is writing dump data to specified file(s).

Action: Wait for the Online Dump Facility to finish dumping the co-processor adapter’s memory and I/O ports.

ICADPR32I: AutoDump beginning....

Explanation: A co-processor adapter requested that its memory and I/O ports be dumped to disk when a Level 1 error occurs on the adapter. The write to disk is beginning.

Action: Wait for the Online Dump Facility to finish dumping the co-processor adapter’s memory and I/O ports. The output files can be formatted using the Dump Formatter Facility.

ICADPR33I: usage: icadpric cardnumber -[d | ea | da][-dd directory]

Explanation: Invalid entry on the command line.

Action: Re-enter correct information using the indicated format.

Online Dump Facility Error Messages

ICADPR01E: Invalid co-processor adapter number: nn.

Explanation: Co-Processor adapter number **nn** was not initialized by the device driver and is not recognized as being installed.

Action: Check the co-processor adapter number to make sure that it matches the co-processor adapter to be dumped.

ICADPR02E: Co-processor nn already enabled for AutoDump.

Explanation: An attempt was made to enable AutoDump on a co-processor adapter (**nn**) that has already been enabled for AutoDump.

Action: Verify that the co-processor adapter is enabled for AutoDump by checking to see if the process identification (PID) number contained in the AutoDump tag file (/tmp/AUTODUMP.X, where X is the co-processor adapter number) is an active **icadpric** process.

Example: If PID number 6789 is contained in the tag file AUTODUMP.1 (for co-processor adapter 1), use the "ps 6789" command to check the status of process 6789. If the output of the "ps" command contains "icadpric 1 -ea", the process is already enabled for AutoDump. If PID 6789 was not found, remove the tag file and attempt to enable the co-processor adapter again.

ICADPR03E: Co-processor nn not enabled for AutoDump.

Explanation: An attempt was made to disable AutoDump on a co-processor adapter (**nn**) that has not been enabled for AutoDump.

Action: Verify that the co-processor adapter is enabled for AutoDump by checking to see if the AutoDump tag file (/tmp/AUTODUMP.X, where X is the co-processor adapter number) exists. If it does, then the co-processor adapter is enabled for AutoDump. If it doesn't, then it was never enabled.

ICADPR04E: Illegal flag -xxx

Explanation: An illegal flag (**-xxx**) was entered on the command line.

Action: Invoke the Online Dump Facility by using the indicated format.

usage: icadpric coprocnum -[d | ea | da] [-dd directory].

ICADPR05E: Cannot access directory xxx.

Explanation: The directory **xxx** (specified with the **-dd** flag) does not exist or cannot be accessed.

Action: If the directory does not exist, create it or specify a different path. If the directory cannot be accessed, obtain write permission for that directory or specify a different path.

ICADPR06E: Unable to perform dump. Coproc nn not responding.

Explanation: Co-processor **nn** is not responding to commands from the Online Dump Facility.

Action: Make sure that the Realtime Control Microcode is loaded and running on the co-processor adapter.

ICADPR07E: AutoDump not enabled on coproc nn. Coproc not responding.

Explanation: Co-processor **nn** is not responding to commands from the Online Dump Facility.

Action: Make sure that the Realtime Control Microcode is loaded and running on the co-processor adapter.

ICADPR08E: Dump data will not fit on file system. Dump of coproc nn canceled.

Explanation: The Online Dump Facility detected that the file system where the dump data is to be stored does not have enough free space.

Action: Make room on the target file system by removing files, or choose a different file system for the dump data.

ICADPR09E: AutoDump data will not fit on file system. AutoDump of coproc nn canceled.

Explanation: The Online Dump Facility detected that the file system where the dump data is to be stored does not have enough free space.

Action: Make room on the target file system by removing files, or choose a different file system for the dump data.

ICADPR10E: ARTIC device driver is not installed.

Explanation: The Co-Processor AIX Support device driver is not installed so the Online Dump Facility cannot execute.

Action: Install the Co-Processor AIX Support device driver.

ICADPR11E: Error opening xxx. Return code = nn.

Explanation: The Online Dump Facility encountered an error while attempting to open file **xxx**. Return code **nn** from the **open** system call is also displayed.

Action: Verify that file **xxx** exists and has read permission.

ICADPR12E: Error reading xxx. Return code = nn.

Explanation: The Online Dump Facility encountered an error while attempting to read file **xxx**. Return code **nn** from the **read** system call is also displayed.

Action: Verify that file **xxx** exists, has read permission, and contains data to be read.

ICADPR13E: Error closing xxx. Return code = nn.

Explanation: The Online Dump Facility encountered an error while attempting to close file **xxx**. Return code **nn** from the **close** system call is also displayed.

Action: Verify that file **xxx** exists.

ICADPR14E: Cannot create AutoDump tag file xxx.

Explanation: The Online Dump Facility could not create the AutoDump tag file.

Action: Verify directory **/tmp** has write permission, and that the file system where /tmp resides has enough space.

ICADPR15E: Cannot allocate memory for dump.

Explanation: The Online Dump Facility could not allocate enough memory for the dump.

Action: Reduce system load and try again.

ICADPR16E: Error writing to system or memory dump files.

Explanation: The Online Dump Facility could not write to the system or memory dump files.

Action: Check to make sure that the system or memory dump files were not removed before the dump was completed. Also check to verify that the file system did not run out of space for the dump files.

ICADPR17E: Device Driver error, return code = nn.

Explanation: The Online Dump Facility encountered an error during a device driver call. The return code from the device driver is **nn**.

Action: Refer to Appendix C, “Return Codes” for the source of the problem.

Dump Formatter Facility Information Messages

Message File missing or invalid.

Explanation: The message file **frmtdump.msg** could not be accessed. The default messages (U.S. English) are being used.

File already exists: xxxxxxxx.xxx
Press:
0 To Abort
1 To Overwrite

Explanation: The Dump Formatter Facility output file **xxxxxxxx.xxx** already exists. User is being prompted to decide whether to abort the formatting process or continue and overwrite the old output file.

Dump Formatter Facility Error Messages

ICAFMT01E: Cannot open input files.

The dump files **ICAME***N* and/or **ICASYS***N* could not be opened by the Dump Formatter Facility.

ICAFMT02E: Cannot write to output directory xxx.

The output directory **xxx** (specified with the **-fd** flag) cannot be found or the Dump Formatter Facility does not have write access to it.

ICAFMT03E: Illegal flag -xxx

Explanation: An illegal flag (**-xxx**) was entered on the command line.

Action: Correct the error and retry the command.

usage: frmtdump *card number* **-[m | s] [-fd]**

Appendix E. Using the Sample Programs

This appendix describes the sample programs that are included on the Realtime Interface Co-Processor AIX Support program diskette. After installation, the files are located in two directories, **sample/user** and **sample/task**.

Following is a list of the files and a short description of each:

sample/user:

Filename	Description
readme	Text file that lists all system unit sample files
suioctl.c	Sample system unit program source—uses ioctl calls
suioctl	Sample system unit program executable file - uses C Language interface routines
suclib.c	Sample system unit program source - uses C Language interface routines
suclib	Sample system unit program executable file - uses C Language interface routines
icadisp.c	Sample system unit display program source—displays co-processor adapter information file
icadisp	Sample system unit program executable file
icadata.h	Include file for sample system unit display program
makeall	Shell script to make samples
makefile	Makefile for all samples

sample/task:

Filename	**Description**
readme	Text file that lists all co-processor adapter sample files
ricsamp.c	Sample co-processor adapter program source.[1]
ricsamp	Co-processor adapter sample program executable file.
ricsamp.map	Co-processor adapter sample program executable file.

The sample programs provided with the Co-Processor AIX Support product are used together in two ways:

- The system unit sample programs **suioctl** (ioctl calls) and **suclib** (C Language interface routines) run with the sample co-processor adapter task **ricsamp** and demonstrate that the device driver is working. See "Running the Demonstration Sample Programs" on page E-5 for instructions on running the programs.
- The system unit sample **icadisp** is used to display task parameters for the co-processor sample task **ricsamp**. See "Running the Display Sample Program" on page E-6 for instructions on running the programs.

[1] To compile and link the co-processor adapter sample program **ricsamp.c**, you will need IBM Realtime Interface Co-Processor C Language Support Version 1.03 and the Microsoft C 6.0 Optimizing Compiler. The **ricsamp** executable file may be used without compiling and linking.

Compiling and Linking System Unit Sample Programs

The sample system unit programs are provided on the Realtime Interface Co-Processor AIX Support program diskette in both source and executable form.

The source files (**suioctl.c**, **suclib.c**, and **icadisp.c**) are written in C Language and compiled using the file **makefile**. To create the executable files for the system unit:

Type: **makeall**

If using AIX Version 4.0, refer to the *IBM C for AIX User's Guide*. If using AIX Version 3.0, refer to the *IBM XL C Compiler User's Guide for AIX Version 3* for more information about writing C Language programs for AIX. Refer to “Running the Demonstration Sample Programs” on page E-5 in this appendix for instructions on how to invoke the system unit and co-processor adapter samples.

Compiling and Linking the Co-Processor Adapter Sample Program

The sample co-processor adapter task **ricsamp** demonstrates that the device driver is working. This sample task also writes values into the CIO 0 port A for use with the Online Dump Facility.

To compile and link **ricsamp**, you must have a system unit that supports either IBM DOS or OS/2.

See the instructions in "Running the Demonstration Sample Programs" on page E-5 in this appendix for information on how **ricsamp** works.

To compile and link the **ricsamp.c** task using the Microsoft C Optimizing Compiler Version 6.0, do the following:

```
cl /AL /G1 /Zp /c ricsamp.c
link icaheadc+ricsamp/NOD /NOE,ricsamp,ricsamp,icams60l+icatskl+llibce /M
```

The files **icaheadc.obj, icams60l.lib,** and **icatskl.lib** are part of the Realtime Interface Co-Processor C Language Support Version 1.03 (or higher).

Use the following compiler options when compiling C tasks:

/G1 This option causes the compiler to generate code for the 80186, the CPU on the co-processor adapter.

/Zp This option causes the supplied structures to be packed. If they are not packed, the generated programs do not access the structures correctly.

When linking C object modules for the co-processor adapter, certain link parameters and options must be used. Consider the following when linking C tasks.

- The **/NOE** option must be used. This option prevents the linker from searching the extended dictionary (an internal list of intermodule dependencies). Use this option to suppress error messages about multiple symbol definitions.
- The **/NOD** option must be used. This is the option for no default library search. All libraries used by a task must be specified in the link command.
- The task header file **icaheadc.obj** must be the first file to link.
- The C Language Support interface routine library **icatskl.lib** must be specified before the standard C library in the link line.

For more information on creating C Language co-processor adapter tasks, refer to the *Realtime Interface Co-Processor C Language Support, Version 1.03 (or higher) User's Guide, Volume II - Co-Processor Adapter*.

You can also write co-processor adapter tasks using the IBM Macro Assembler/2, Version 1.0, or Microsoft Macro Assembler 5.1. For more information, refer to the *Realtime Interface Co-Processor Firmware Technical Reference*.

Running the Demonstration Sample Programs

To run the sample programs, follow these steps:

1. Load the Realtime Control Microcode (this example runs on co-processor adapter 0):

   - For a Realtime Interface Co-Processor Portmaster Adapter/A or Multiport Model 2, type:

     icaldric 0 icarcm.com 0 -reset

   - For an X.25 Interface Co-Processor/2, X.25 Interface Co-Processor, X.25 Interface Co-Processor PCI Adapter, Realtime Interface Co-Processor Multiport, Realtime Interface Co-Processor Multiport/2, or ARTIC186 8-Port PCI Adapter, type:

     icaload 0 icaaim.com 0 -reset

2. Load the co-processor adapter task as task 1:

   icaload 0 ricsamp.exe 1

3. Invoke a system unit sample program by typing one of the following:

   - **suioctl adapternumber tasknumber [number_of_iterations]**

   - **suclib adapternumber tasknumber [number_of_iterations]**

   where

   adapternumber = Co-processor adapter number.

   tasknumber = Number of the co-processor adapter task

   number_of_iterations = (Optional) Number of times the system unit program loops through its tests. The default is 10.

As the sample system unit program runs, information is displayed about the various system unit function calls that are being made.

Note: The sample program **suioctl** uses the ioctl system unit calls, while the sample program **suclib** uses the C Language interface calls.

Running the Display Sample Program

To run the display sample program, follow these steps:

1. Load the Realtime Control Microcode (this example runs on co-processor adapter 0):

   For a Realtime Interface Co-Processor Portmaster Adapter/A or Multiport Model 2, type:

   icaldric 0 icarcm.com 0 -reset

   For an X.25 Interface Co-Processor/2, X.25 Interface Co-Processor, X.25 Interface Co-Processor PCI Adapter, Realtime Interface Co-Processor Multiport/2, Realtime Interface Co-Processor Multiport/2, or ARTIC186 8-Port PCI, type:

   icaload 0 icaaim.com 0 -reset

2. Load the sample co-processor adapter task as task 1:

   Type: **icaldric 0 ricsamp 1**

3. Invoke the sample system unit display program **icadisp**

   Type: **icadisp 0 1**

   where:

   icadisp = Sample system unit program that displays information about the co-processor adapter

   0 = Number of the co-processor adapter

   1 = Number of the co-processor adapter task

The sample display program provides a menu of co-processor adapter data structures that can be displayed.

Appendix F. Changing the Shared Storage Window Size

The following procedure describes how to change the shared window size.

The default shared storage size is 8 KB. The shared storage window size can be changed only for the Micro Channel adapters.

To change the shared storage window size:

1. Log on as **root** (or obtain superuser privileges).
2. Execute the command: **smitty**

   Note: In the following steps, use the up arrow and down arrow keys to highlight an entry, and press **Enter** to select it. Each time you make a selection, the SMIT program automatically advances to the next menu.
3. From the first SMIT menu, select **Devices**.
4. From the next SMIT menu, select **Communications**.
5. From the next SMIT menu, made a selection to change the shared window size of **one** type of adapter:
   - To change the shared window size on a Portmaster Adapter/A, select **Postmaster Adapter/A**.
   - To change the shared window size on an X.25 Co-Processor or a Multiport/2 Adapter, select **X.25 Co-Processor or Multiport/2 Adapter**.
6. From the first SMIT menu, select **Adapter**.
7. From the next SMIT menu, make a selection to manage the appropriate device driver for the type of adapter you previously chose.
   - To change the shared window size on a Portmaster Adapter/A, select:

     Manage Device Drivers for Portmaster Adapter/A
   - To change the shared window size on an X.25 Co-Processor or a Multiport/2 Adapter, select:

     Manage Device Drivers for X.25 Co-Processor or Multiport/2 Adapter
8. From the next SMIT menu, select:

   Manage RIC AIX Support Device Driver
9. From the next SMIT menu, select:

   Change/Show Characteristics of a Device Driver
10. On the next SMIT menu, a list of adapters is displayed. Highlight the adapter on which you want to change the shared window size and press **Enter**.
11. To select a window size, type *one* of the following binary patterns, and press **Enter**. For example, for a window size of 32 KB, type **0X8000**.

Binary Pattern	Desired Window Size
0X2000	8 KB
0X4000	16 KB
0X8000	32 KB
0X10000	64 KB

12. If you have other adapters on which you would like to change the window size, press **F3** to return to Step 9 on page F-1. Otherwise, press **F10** to exit SMIT.

Appendix G. Notices

This information was developed for products and services offered in the U.S.A. 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 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Trademarks and Service Marks

The following terms are trademarks of the IBM Corporation in the United States or other countries, or both:

AIX
IBM
Macro Assembler/2
Micro Channel
Personal System/2
Portmaster
Proprinter
PS/2

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United states and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.