



p/n YAL304NT

AcroLIB API User's Guide

Effective: October 7, 2002

AcroLIB API – User's Guide

© Copyright 1996-2002 ACROLOOP MOTION CONTROL SYSTEMS INC.

All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement.

ACROLOOP MOTION CONTROL SYSTEMS, INC. SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATING TO THE USE OF THIS PRODUCT.

Microsoft and Windows are registered trademarks and Visual C++ and Visual Basic are trademarks of Microsoft Corporation.

This guide is intended to be used for the following software product(s).

AcroLIB For NT Version 3.04 (Part # YAL304 NT)

If you have any comments about the software or documentation, please contact

ACROLOOP MOTION CONTROL SYSTEMS INC.
3650 CHESTNUT STREET NORTH
CHASKA, MN 55318
USA

Telephone: (952)-448-9800 Fax: (952)-448-9321

<http://www.acroloop.com>

Table of Contents

CHAPTER 1 OVERVIEW	1
THE ACROLIB INTERFACE.....	2
REQUIREMENTS FOR USING ACROLIB INTERFACE.....	2
INSTALLING THE API INTERFACE	2
DISTRIBUTING YOUR APPLICATION	2
CHAPTER 2 USING ACROLIB API.....	3
ESTABLISHING COMMUNICATIONS WITH THE CARD	4
SENDING A TEXT FILE TO THE CARD.....	4
READING PARAMETERS FROM THE CARD.....	4
READING USER-DEFINED PARAMETERS	5
READING SYSTEM-DEFINED PARAMETERS.....	5
READING LOCAL USER PARAMETERS	5
SETTING PARAMETERS ON THE CARD.....	6
SETTING USER-DEFINED PARAMETERS	6
SETTING SYSTEM-DEFINED PARAMETERS	6
SETTING LOCAL-USER PARAMETERS	7
SETTING/CLEARING BITS ON THE CARD.....	7
SENDING MOVES TO THE CARD.....	7
SYNCHRONIZING WITH MOVES ON THE CARD	8
DYNAMICALLY CHANGING MOVE PROFILES.....	8
SENDING ACROBASIC COMMANDS TO THE CARD.....	8
SETTING UP A TERMINAL TO THE CARD	8
CHAPTER 3 FUNCTION REFERENCE.....	9
A8_BIN_ADDRESS	10
A8_BIN_CLR.....	11
A8_BIN_FOV	12
A16_BIN_FOV	13
A8_BIN_GET_STATUS	14
A8_BIN_GETIEEE	15
A8_BIN_GETLONG	16
A8_BIN_GROUP_GETIEEE	17
A8_BIN_GROUP_GETLONG.....	18
A8_BIN_INIT_STATUS.....	19
A8_BIN_MASK.....	22
A8_BIN_MOVE_IEEE	23
A16_BIN_MOVE_IEEE	25
A8_BIN_MOVE_LONG.....	27
A16_BIN_MOVE_LONG.....	29
A8_BIN_PEEK_IEEE.....	31
A8_BIN_PEEK_LONG	32
A8_BIN_POKE_IEEE	33
A8_BIN_POKE_LONG	34
A8_BIN_ROV	35
A16_BIN_ROV	36
A8_BIN_SET.....	37
A8_BIN_SETLONG	38
A8_BIN_SETIEEE	39
A8_BIN_SYS_MASK	40
ACR8K_FAST_STATUS.....	41
ACR8K_FAST_STATUS_REQ.....	42
ACR8K_IEEE_MOVE	43
A16_IEEE_MOVE	45
ACR8K_LONG_MOVE	47
A16_LONG_MOVE.....	49

ACROBINARYBUFFEREMPTY.....	51
ACROBINARYBUFFERNUMBEROFELEMENTS	52
ACROCARDPRESENT	53
ACROGETBINARYTIMEOUT.....	54
ACROGETBINARYTIMERDELTA	55
ACROGETBINTRANSTHRESHOLD	56
ACROGETDRBINARYBUFFERSTATE	57
ACROGETDRBINARYBUFFLENGTH.....	58
ACROGETDRBINARYBUFFSIZE	59
ACROGETDRREADBUFFLENGTH	60
ACROGETDRREADBUFFSIZE.....	61
ACROGETDRWRITEBUFFLENGTH	62
ACROGETDRWRITEBUFFSIZE	63
ACROGETERROR	64
ACROGETFASTSTATUSTIMERDELTA.....	65
ACROGETMAXBINTRANSPERFSINTR	66
ACROGETMAXFASTSTATUSDELTA	67
ACROGETMOVECOUNTER.....	68
ACROGETPERFORMANCE	69
ACROGETREADTIMEOUT	72
ACROGETREADTIMERDELTA.....	73
ACROGETSTATUSTIMERDELTA.....	74
ACROGETTIMEMILLI.....	75
ACROGETTRANSACTIONSTATE.....	76
ACROGETVERSION	77
ACROGETWRIETIMEOUT.....	78
ACROGETWRIETIMERDELTA	79
ACROHIDEERRORDIALOGS.....	80
ACROINITIALIZE	81
ACROINITPERFORMANCE.....	82
ACRORECEIVETEXTSTRING.....	83
ACROSENDESCAPE.....	84
ACROSENDSTRING	85
ACROSENDTEXTSTRING.....	86
ACROSETBINARYTIMEOUT	87
ACROSETBINARYTIMERDELTA.....	88
ACROSETBINTRANSTHRESHOLD	89
ACROSETERROR	90
ACROSETFASTSTATUSTIMERDELTA	91
ACROSETINITIALWAIT	92
ACROSETMAXBINTRANSPERFSINTR.....	93
ACROSETMAXFASTSTATUSDELTA.....	94
ACROSETMAXSIZEOFINPUTBINARYBUFFER	95
ACROSETMOVECOUNTER	96
ACROSETREADTIMEOUT	97
ACROSETREADTIMERDELTA	98
ACROSETRETRYBEFOREBINREADTIMEOUT.....	99
ACROSETRETRYBEFOREBINWRIETIMEOUT	100
ACROSETRETRYBEFOREREADTIMEOUT	101
ACROSETRETRYBEFOREWRIETIMEOUT	102
ACROSETSTATUSTIMERDELTA	103
ACROSETTIMEOUTFORBINREADMUTEX	104
ACROSETTIMEOUTFORBINTRANSMUTEX	105
ACROSETTIMEOUTFORBINTRANSWAIT	106
ACROSETTIMEOUTFORBINWRITEMUTEX.....	107
ACROSETTIMEOUTFORREADMUTEX	108
ACROSETTIMEOUTFORWRITEMUTEX.....	109
ACROSETTRANSACTIONSTATE	110
ACROSETWAITBEFORETIMEOUT	111

ACROSETWAITBETWEENSTATUSFETCH	112
ACROSETWRITE TIMEOUT	113
ACROSETWRITE TIMER DELTA	114
ACROSHOWERRORDIALOGS	115
ACROTEXTBUFFEREMPTY	116
ACROWAIT	117
CHAPTER 4 ERRORS	119
WINDOWS NT	120
ACRO_SUCCESS (0)	121
ACRO_WRITE_TIMEOUT_ERROR (-1)	122
ACRO_READ_TIMEOUT_ERROR (-2)	123
ACRO_READ_MUTEX_TIMEOUT_ERROR (-3)	124
ACRO_WRITE_MUTEX_TIMEOUT_ERROR (-4)	125
ACRO_CARD_NOT_FOUND (-10)	126
ACRO_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-11)	127
ACRO_BIN_READ_MUTEX_TIMEOUT_ERROR (-13)	128
ACRO_BIN_WRITE_MUTEX_TIMEOUT_ERROR (-14)	129
ACRO_BIN_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-15)	130
ACRO_BIN_WRITE_TIMEOUT_ERROR (-16)	131
ACRO_BIN_READ_TIMEOUT_ERROR (-17)	132
ACRO_BIN_TRANS_TIMEOUT_ERROR (-18)	133
ACRO_BIN_TRANS_MUTEX_TIMEOUT_ERROR (-19)	134
ACRO_BIN_TRANS_WAIT_TIMEOUT_ERROR (-20)	135
ACRO_BIN_TRANS_RANGE_VIOLATION_ERROR (-22)	136
APPENDIX A PARAMETER REFERENCE	137
PARAMETER OVERVIEW	137
APPENDIX B FLAG REFERENCE	139
FLAG OVERVIEW	139
APPENDIX C BINARY HOST INTERFACE	141
BINARY DATA TRANSFER	141
APPENDIX D ARCHITECTURE	143
APPLICATION LEVEL	144
DLL LEVEL	145
DRIVER LEVEL	146
INDEX	147

This page intentionally left blank.

Chapter 1 Overview

This chapter provides an overview of the AcroLIB API.

The ACROLIB Interface

The goal of ACROLIB API is to provide programmers with high level functions that can be used to construct applications for the Acroloop cards for the Microsoft Windows operating system.

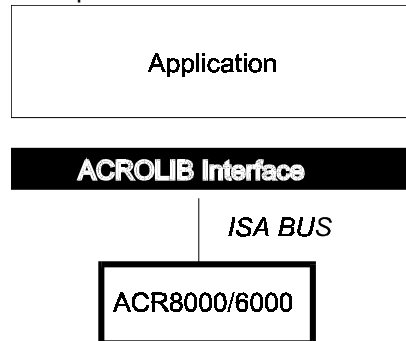


Figure 1 The ACROLIB API Interface

Figure 1 shows how this library allows an application to communicate with an Acroloop card over the PC ISA Bus.

Requirements for Using ACROLIB Interface

The ACROLIB interface is supported in the following programming languages and platforms.

platform	Language	Product
Microsoft Windows NT 3.51+	C/C++	AcroLib For NT
Microsoft Windows NT 3.51+	Microsoft Visual Basic	AcroLib For NT

This product assumes that you are familiar with Microsoft Visual C++ or Microsoft Visual Basic. It also assumes that you are familiar with programming for the Windows operating system.

Installing the API Interface

The instructions of installing the API interface are included in the disks provided for each of the products.

Distributing Your Application

When you write your application using the ACROLIB API, you must distribute the Acroloop DLL (ACROWNT.DLL), the appropriate driver (ACR8K.SYS for NT) and Control Panel Applet (ACR8K.CPL) and make the appropriate Registry settings for the driver (call Acroloop Technical Support) with your application.

Chapter 2 Using AcroLib API

This chapter explains how you can use the AcroLib API in your application.

After establishing a successful communication link with the card, a typical application carries out a subset of the following sequence of operations.

1. Sends down text files(s) containing onboard programs and PLCs to the card.
2. Periodically reads parameters such as the current position, input/output status or flags from the card.
3. Sets and clear bits/flags on the card.
4. Sets various parameters such as user defined arrays on the card.
5. Sends moves to the card.
6. Synchronizes with the moves being currently executed.
7. Dynamically changes the move profiles.
8. Sends ACROBASIC commands to the card.
9. Sets up a terminal to talk to the card.

The following table lists the source code examples provided for each of the functions above.

Function	C/C++ Source Code	Visual Basic Source Code
Sending text files	sendfile.mdp	File Tab in sample.vbp.
Reading parameters	showstatus.mdp, showgrp.mdp, showparm.mdp	Status Tab and Fast Status Tab in sample.vbp.
Setting and clearing flags	sendfile.mdp	Fast Status Tab in sample.vbp
Setting parameters	pekpok.mdp	Status Tab in sample.vbp
Sending moves	binmove.mdp	Moves Tab in sample.vbp
Synchronizing moves	binmove.mdp	Moves Tab in sample.vbp
Sending AcroBASIC commands	sendfile.mdp	Terminal Tab in sample.vbp
Terminal	terminal.mdp	Terminal Tab in sample.vbp

Table 1. List of C/C++ and Visual Basic source code examples.

Establishing Communications with the Card

An application establishes communications with the card by calling AcroInitialize function. An application should only continue if this function does not generate any error. See table above for examples in C/C++ and Visual Basic.

Sending a Text File to the Card

An application can send down programs and PLC's written in ACROBASIC using AcroSendString function in conjunction with AcroTextBufferNumberOfElements, AcroSendTextString and AcroReceiveTextString.

1. Read a line from FileToSend until the EOF is reached.
2. Use AcroTextBufferNumberOfElements to determine if the input TXT buffer has any elements.
3. Read elements (if any) in the input TXT buffer using AcroReceiveTextString. Display the elements rightaway or store them in a buffer for displaying later.
4. Write the line read from FileToSend using the AcroSendTextString function.
5. Goto 1.

See sendfile.mdp (for C/C++) or File Tab in sample.vbp (for Visual Basic) for example source code.

Reading Parameters from the Card

The Acroloop Motion Controller cards have two types of parameters.

1. Global System Parameters
2. Local User Parameters

The Global System Parameters can further be divided into User-defined Parameters (P0-P4095) and System-defined Parameters (P4096 and above). The Local User Parameters include LV, LA, SV, SA, DV, DA, \$V and \$A variables locally defined in each user program.

The following table lists the functions that can be used to read the various types of parameters from the card.

Parameter Type	Access Functions
User-defined Parameters	A8_BIN_GETIEEE, A8_BIN_GETLONG, A8_BIN_PEEK_LONG, A8_BIN_PEEK_IEEE
System-defined Parameters	A8_BIN_GETIEEE, A8_BIN_GETLONG, A8_BIN_GROUP_GETIEEE, A8_BIN_GROUP_GETLONG, A8_BIN_PEEK_LONG, A8_BIN_PEEK_IEEE, A8_GET_STATUS
Local User Parameters	A8_BIN_PEEK_LONG, A8_BIN_PEEK_IEEE

Reading User-Defined Parameters

See Binary Global Parameter Access, located in Acroloop Motion Controller User's Guide (P/N PM08120-2), Chapter 6, Binary Host Interface.

See showparm.mdp (for C/C++) or Status Tab in sample.vbp (for Visual Basic) for example sources code.

Reading System-Defined Parameters

System-defined parameters (see Appendix A) can be read one at a time using A8_BIN_GETLONG or A8_BIN_GETIEEE. Alternatively, groups of these variables (for example, current position for all 8 axes) can be read more efficiently by using the A8_BIN_GROUP_GETIEEE or A8_BIN_GROUP_LONG functions.

See showparm.mdp, showgrp.mdp and showstat.mdp (for C/C++) or Status and Fast Status Tab in sample.vbp (for Visual Basic) for example source code.

Reading Local User Parameters

User-defined local parameter can only be read using the A8_BIN_PEEK_LONG or A8_BIN_PEEK_IEEE commands.

Reading such local variables usually consist of three distinct steps.

1. Determine how many (if any) variables of a specific type (e.g., LV) are defined in a program space (e.g., PROG5) of interest by using A8_BIN_ADDRESS function. If the function returns 0, it means that no variables of that specific type have been dimensioned. If the requested variable type has actually been dimensioned, the A8_BIN_ADDRESS function will a non-zero number.
2. Peeking at the returned address using A8_BIN_PEEK will return the number of variables dimensioned for the requested type. In the case of numeric variables (DV,SV,LV) the count will be followed by the actual numeric data. For arrays (DA, SA, LA) the count will be followed by the addresses of the individual arrays. These addresses point to storage areas as if they were normal numeric variables of the same type (count followed by data.)
3. In case of numeric variables (DV,SV,LV), use A8_BIN_PEEK to retrieve the actual values of the variables. For Arrays (DA, DA, LA) use A8_BIN_PEEK to retrieve the addresses of storage areas and subsequently use A8_BIN_PEEK on each of these areas to retrieve the actual data.

See pekpok.mdp (for C/C++) or Status Tab in sample.vbp (for Visual Basic) for example source code.

Setting Parameters on the Card

The Acroloop cards have two types of parameters.

1. Global System Parameters
2. Local User Parameters

The Global System Parameters can further be divided into User-defined Parameters (P0-P4095) and System-defined Parameters (P4096 and above). The Local User Parameters include LV, LA, SV, SA, DV, DA, \$V and \$A variables locally defined in each user program.

The following table lists the functions that can be used to set or update various types of parameters on the card.

Parameter Type	Update Function
User-defined Parameters	A8_BIN_SETIEEEE, A8_BIN_SETLONG, A8_BIN_POKE_LONG, A8_BIN_POKE_IEEE
System-defined Parameters	A8_BIN_SETIEEEE, A8_BIN_SETLONG, A8_BIN_POKE_LONG, A8_BIN_POKE_IEEE
Local User Parameters	A8_BIN_POKE_LONG, A8_BIN_POKE_IEEE

Setting User-Defined Parameters

See Binary Global Parameter Access, located in Acroloop Motion Controller User's Guide (P/N PM08120-2), Chapter 6, Binary Host Interface.

See pekpok.mdp (for C/C++) or Status Tab in sample.vbp (for Visual Basic) for example source code.

Setting System-Defined Parameters

System-defined parameters can be changed by using A8_BIN_SETIEEEE or A8_BIN_SETLONG. Alternatively, A8_BIN_POKE or A8_BIN_POKE_IEEE can be used in conjunction with A8_BIN_SYS_ADDRESS.

See pekpok.mdp (for C/C++) or Status Tab in sample.vbp (for Visual Basic) for example source code.

Setting Local-User Parameters

User-defined local parameter can only be set using the `A8_BIN_POKE_LONG` or `A8_BIN_POKE_IEEE` commands.

Setting such local variables usually consist of three distinct steps.

1. Determine how many (if any) variables of a specific type (e.g., LV) are defined in a program space (e.g., PROG5) of interest by using `A8_BIN_ADDRESS` function. If the function returns 0, it means that no variables of that specific type have been dimensioned. If the requested variable type has actually been dimensioned, the `A8_BIN_ADDRESS` function will a non-zero number.
2. Peeking at the returned address using `A8_BIN_PEEK` will return the number of variables dimensioned for the requested type. In the case of numeric variables (DV,SV,LV) the count will be followed by the actual numeric data. For arrays (DA, SA, LA) the count will be followed by the addresses of the individual arrays. These addresses point to storage areas as if they were normal numeric variables of the same type (count followed by data.)
3. In case of numeric variables (DV,SV,LV), use `A8_BIN_POKE` to set the actual values of the variables located in addresses following the count retrieved in previous step. For Arrays (DA, DA, LA) use `A8_BIN_PEEK` to retrieve the addresses of storage areas and subsequently use `A8_BIN_POKE` on each of these areas to set the values for actual data.

See `pekpok.mdp` (for C/C++) or Status Tab in `sample.vbp` (for Visual Basic) for example source code.

Setting/Clearing Bits on the Card

Bits can be set and cleared on the card by using the `A8_BIN_SET` and `A8_BIN_CLR` functions respectively. Complete registers of flags or bits can be changed by using `A8_BIN_MASK` and `A8_BIN_SYS_MASK` functions.

See `sendfile.mdp` (for C/C++) or Fast Status Tab in `sample.vbp` (for Visual Basic) for example source code.

Sending Moves to the Card

Moves can be sent to the card using `A8_BIN_MOVE_LONG` or `A8_BIN_MOVE_IEEE` depending the type of target (long or IEEE floating point).

See `binmove.mdp` (for C/C++) or Moves Tab in `sample.vbp` (for Visual Basic) for example source code.

Synchronizing with Moves on the Card

Often it is necessary to synchronize the front-end application with the moves executing on the card. AcroLib provides an interrupt-based mechanism to carry out this synchronization.

1. AcroSetMoveCounter is used to set movecounter to a known number (say 0).
2. AcroGetMoveCounter is used to read the move counter which is incremented every time a move starts on the card (Binary Move interface only).

See binmove.mdp (for C/C++) or Moves Tab in sample.vbp (for Visual Basic) for example source code.

Dynamically Changing Move Profiles

The feedrate override for currently executing moves can be changed by using the A8_BIN_FOV function. The rapid override can be changed by using the A8_BIN_ROV function.

See binmove.mdp (for C/C++) or Moves Tab in sample.vbp (for Visual Basic) for example source code.

Sending ACROBASIC Commands to the Card

Any valid ACROBASIC command can be sent to the card by using the AcroSendString function.

See sendfile.mdp (for C/C++) or File Tab in sample.vbp (for Visual Basic) for example source code.

Setting Up a Terminal to the Card

An ASCII terminal can be setup between the Application and the card by using AcroSendTextString and AcroReceiveString functions.

See terminal.mdp (for C/C++) or Terminal Tab in sample.vbp (for Visual Basic) for example source code.

Chapter 3 Function Reference

This chapter is an alphabetic reference for functions in AcroLib API. The Libraries field for each function indicates the particular library that supports the function; AcroLibNT means AcroLib for NT.

A8_BIN_ADDRESS

Description	Read the address of a program specific user-defined variable from the card.(not for global user variables)
Visual Basic	<pre>Declare Function A8_BIN_ADDRESS Lib "acrownt.dll" (ByVal programnumber As Long, ByVal parametercode As Long, pReturnLong As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>int A8_BIN_ADDRESS(long programnumber, long parametercode, unsigned long* pReturnLong,int card) ;</pre>
Returns	Error returned by AcroGetError.
Arguments	<p>programnumber—number of the program space containing the user-defined variable.</p> <p>parametercode—code defining the type of user-defined variable being addressed.</p> <p>pReturnLong—pointer to a buffer that will receive the address</p> <p>card—number of the card being used.</p>
Error	Error returned by AcroGetError
Globals	none.
Remarks	This command is used in conjunction with A8_BIN_PEEK_LONG, A8_BIN_PEEK_IEEE to read user-defined variables from the card. Note that the address returned contains the number of variables of the type being sought followed by the actual variables. For more details see the Binary Hostinterface Section in the Appendix C.
Visual Basic Example	<pre>Dim MyError As Long Dim pAddress As Long 'from program space 0, read the base address for the SV variables MyError=A8_BIN_ADDRESS (0, 2,pAddress,0)</pre>
Visual C++ Example	<pre>int MyError; long pAddress[1]; // Read the base address for String Arrays (\$A) variables in program space 0 A8_BIN_ADDRESS(0,6, pAddress,0)</pre>
See Also	A8_BIN_PEEK_LONG, A8_BIN_PEEK_IEEE
Libraries	AcroLibNT

A8_BIN_CLR

Description	Clear a bit (immediately) on the card
Visual Basic	<i>Declare Sub A8_BIN_CLR Lib "acrownt.dll" (ByVal Flg As Integer, ByVal card As Long)</i>
Visual C++	<i>void A8_BIN_CLR(short Flag,int card);</i>
Returns	nothing.
Arguments	Flag—number of the flag being cleared. card—number of the card being used.
Error	Error returned by AcroGetError
Globals	none.
Remarks	A8_BIN_CLR clears the bit immediately. Use AcroSendString to queue a bit clearing behind a move. For example, AcroSendString("CLR32",0).
Visual Basic Example	<pre>'clear output 32 A8_BIN_CLR 32,0</pre>
Visual C++ Example	<pre>//clear output 32 A8_BIN_CLR(32,0)</pre>
See Also	A8_BIN_SET
Libraries	AcroLibNT

A8_BIN_FOV

Description	Send an immediate FOV to master(s) on the card.
Visual Basic	<pre>Declare Sub A8_BIN_FOV Lib "acrownt.dll" (ByVal mastermask As Integer, ByVal fovvalue As Single, ByVal card As Long)</pre>
Visual C++	<pre>void A8_BIN_FOV(short mastermask, float fovvalue, int card);</pre>
Returns	nothing.
Arguments	mastermask—mask specifying masters for which FOV will be activated. fovvalue—value of the feedrate overried. card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	
Visual Basic Example	<pre>'set the fov to 10 times for master 0 A8_BIN_FOV 1,10,0</pre>
Visual C++ Example	<pre>//set the FOV to 10 times for master0 A8_BIN_FOV(1,10,0);</pre>
See Also	
Libraries	AcroLibNT

A16_BIN_FOV

Description	Send an immediate FOV to 16 master(s) on the card.
Visual Basic	<i>Declare Sub A16_BIN_FOV Lib "acrownt.dll" (ByVal mastermask As long, ByVal fovvalue As Single, ByVal card As Long)</i>
Visual C++	<i>void A16_BIN_FOV(long mastermask, float fovvalue, int card);</i>
Returns	nothing.
Arguments	mastermask—mask specifying masters for which FOV will be activated. fovvalue—value of the feedrate overried. card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	ACR8020 only
Visual Basic Example	<pre>'set the fov to 10 times for master 0 A16_BIN_FOV 1,10,0</pre>
Visual C++ Example	<pre>//set the FOV to 10 times for master0 A16_BIN_FOV(1,10,0);</pre>
See Also	
Libraries	AcroLibNT

A8_BIN_GET_STATUS

Description	Reads the fast status buffer.
Visual Basic	<i>Declare Function A8_BIN_GET_STATUS Lib "acrownt.dll" (pStatusBuffer As ACR8K_FAST_STATUS, ByVal card As Long) As Long</i>
Visual C++	<i>unsigned long A8_BIN_GET_STATUS(PACR8K_FAST_STATUS pStatusBuffer,int card);</i>
Returns	Error returned by AcroGetError.
Arguments	pStatusBuffer—pointer to a fast status buffer set by A8_BIN_INIT_STATUS. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Use A8_BIN_INIT_STATUS to setup the fast status buffer before calling this routine.
Visual Basic Example	<pre>Dim AcroError as long AcroError = A8_BIN_GET_STATUS(StatusBuffer(0), 0)</pre>
Visual C++ Example	<pre>A8_BIN_GET_STATUS(pStatusBuffer, 0);</pre>
See Also	A8_BIN_INIT_STATUS
Libraries	AcroLibNT

A8_BIN_GETIEEE

Description	Read a floating point parameter from the card.
Visual Basic	<i>Declare Function A8_BIN_GETIEEE Lib "acrownt.dll" (ByVal parameterindex As Long, pReturnIEEE As Single, ByVal card As Long) As Long</i>
Visual C++	<i>int A8_BIN_GETIEEE(long parameterindex, float* pReturnIEEE,int card);</i>
Returns	Error returned by AcroGetError.
Arguments	parameterindex—parameter number for the value of parameter being fetched from the card. pReturnIEEE—pointer to a buffer receiving the value. card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	This function should only be used to fetch single parameters on the card. It is more efficient to use A8_BIN_GROUP_GETIEEE or A8_BIN_GET_STATUS
Visual Basic Example	<pre>Dim pReturn As Single Dim MyError As Long ' read proportional gain for axis0 MyError= A8_BIN_GETIEEE(12304,pReturn,0)</pre>
Visual C++ Example	<pre>float pReturn[1]; int MyError; //read P10 MyError = A8_BIN_GETIEEE(12304,pReturn,0);</pre>
See Also	A8_BIN_GROUP_GETIEEE, A8_BIN_GET_STATUS
Libraries	AcroLibNT

A8_BIN_GETLONG

Description	Read a single long variable from the card.
Visual Basic	<i>Declare Function A8_BIN_GETLONG Lib "acrownt.dll" (ByVal parameterindex As Long, pReturnLong As Long, ByVal card As long) As Long</i>
Visual C++	<i>int A8_BIN_GETLONG(long parameterindex, long* pReturnLong,int card);</i>
Returns	Error returned by AcroGetError.
Arguments	parameterindex—parameter number for the value of parameter being fetched from the card. pReturnLong—pointer to a buffer receiving the value. card—number of the card being used.
Error	Error returned by AcroGetError
Globals	none.
Remarks	This function should only be used to fetch single parameters on the card. It is more efficient to use A8_BIN_GROUP_GETIEEE or A8_BIN_GET_STATUS
Visual Basic Example	<pre>Dim pReturn As Long Dim MyError As Long 'read actual position for Axis0. MyError= A8_BIN_GETLONG(12290,pReturn,0)</pre>
Visual C++ Example	<pre>long pReturn[1]; int MyError; //read actual position for Axis0. MyError = A8_BIN_GETLONG(12290,pReturn,0);</pre>
See Also	A8_BIN_GROUP_GETLONG, A8_BIN_GET_STATUS
Libraries	AcroLibNT

A8_BIN_GROUP_GETIEEE

Description	Read a set of floating point parameters from the card.
Visual Basic	<pre>Declare Sub A8_BIN_GROUP_GETIEEE Lib "acrownt.dll" (ByVal code As Long, ByVal Index As Long, ByVal mask As Long, pArray As Single, ByVal card As Long)</pre>
Visual C++	<pre>int A8_BIN_GROUP_GETIEEE(long code, long index, long mask, float* pArray,int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	<p>code—group code of the parameter group being fetched from Appendix A or Appendix B.</p> <p>index—index of the group code being fetched from Appendix A or Appendix B.</p> <p>mask—mask specifying which parameters in the group to fetch.</p> <p>pArray—pointer to a buffer returning the group values.</p> <p>card—number of the card being used.</p>
Error	Error returned by GetLastError.
Globals	none.
Remarks	See Appendix A and Appendix B for index and codes corresponding to a parameter group. Calculation of the mask is explained in Appendix C.
Visual Basic Example	<pre>Dim pArray(2) As Single A8_BIN_GROUP_GETIEEE &H19,&H00,&H03,pArray(0),0 'read DAC output for DAC 0 and 1.</pre>
Visual C++ Example	<pre>float pArray[2]; A8_BIN_GROUP_GETIEEE(0x19,0x00,0x03,pArray,0); //Read DAC output for DAC 0 and 1.</pre>
See Also	A8_BIN_GROUP_GETIEEE
Libraries	AcroLibNT

A8_BIN_GROUP_GETLONG

Description	Read a set of floating point variables from the card.
Visual Basic	<pre>Declare Sub A8_BIN_GROUP_GETLONG Lib "acrownt.dll" (ByVal code As Long, ByVal Index As Long, ByVal mask As Long, pArray As Long, ByVal card As Long)</pre>
Visual C++	<pre>int A8_BIN_GROUP_GETLONG(long code, long index, long mask, long* pArray,int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	<p>code—group code of the parameter group being fetched from Appendix A or Appendix B.</p> <p>index—index of the group code being fetched from Appendix A or Appendix B.</p> <p>mask—mask specifying which parameters in the group to fetch.</p> <p>pArray—pointer to a buffer returning the group values.</p> <p>card—number of the card being used.</p>
Error	Error returned by GetLastError.
Globals	none.
Remarks	See Appendix A and Appendix B for index and codes corresponding to a parameter group. Calculation of the mask is explained in Appendix C.
Visual Basic Example	<pre>Dim pArray(2) As Long A8_BIN_GROUP_GETLONG &H18,&H00,&H03,pArray(0),0 'Read encoder position for ENC 0 and 1.</pre>
Visual C++ Example	<pre>long pArray[2]; A8_BIN_GROUP_GETLONG(0x18,0x00,0x03,pArray,0); //Read encoder position for ENC 0 and 1.</pre>
See Also	A8_BIN_GROUP_GETLONG
Libraries	AcroLibNT

A8_BIN_INIT_STATUS

Description	Setup the fast status buffer.
Visual Basic	<pre>Declare Function A8_BIN_INIT_STATUS Lib "acrownt.dll" (pStatusBuffer As ACR8K_FAST_STATUS, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long A8_BIN_INIT_STATUS(PACR8K_FAST_STATUS pStatusBuffer,int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pStatusBuffer—pointer to a fast status buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim StatusBuffer(0) As ACR8K_FAST_STATUS Dim AcroError as long 'Setup Fast Buffer StatusBuffer(0).HowMany = 10 'FIFO Status StatusBuffer(0).Data(0).Group = &H1B StatusBuffer(0).Data(0).Index = &H8 StatusBuffer(0).Data(0).mask = &H1F StatusBuffer(0).Data(0).type = ACRO_LONG 'General Flags StatusBuffer(0).Data(1).Group = &H10 StatusBuffer(0).Data(1).Index = &H0 StatusBuffer(0).Data(1).mask = &HFF StatusBuffer(0).Data(1).type = ACRO_LONG 'Card Stats StatusBuffer(0).Data(2).Group = &H0 StatusBuffer(0).Data(2).Index = &H0 StatusBuffer(0).Data(2).mask = &HFF StatusBuffer(0).Data(2).type = ACRO_LONG 'Master Distanceinto Move StatusBuffer(0).Data(3).Group = &H20 StatusBuffer(0).Data(3).Index = &H0 StatusBuffer(0).Data(3).mask = &HFF StatusBuffer(0).Data(3).type = ACRO_FP32</pre>

A8_BIN_INIT_STATUS, continued

Visual Basic Example, Continued

```
'Axis Following Error
StatusBuffer(0).Data(4).Group = &H30
StatusBuffer(0).Data(4).Index = &H3
StatusBuffer(0).Data(4).mask = &HFF
StatusBuffer(0).Data(4).type = ACRO_LONG

'Master Vector Velocity
StatusBuffer(0).Data(5).Group = &H20
StatusBuffer(0).Data(5).Index = &H1
StatusBuffer(0).Data(5).mask = &HFF
StatusBuffer(0).Data(5).type = ACRO_FP32

'primary set point
StatusBuffer(0).Data(6).Group = &H30
StatusBuffer(0).Data(6).Index = &H6
StatusBuffer(0).Data(6).mask = &HFF
StatusBuffer(0).Data(6).type = ACRO_LONG

'FOV for Masters
StatusBuffer(0).Data(7).Group = &H20
StatusBuffer(0).Data(7).Index = &H9
StatusBuffer(0).Data(7).mask = &HFF
StatusBuffer(0).Data(7).type = ACRO_FP32

'Program flags
StatusBuffer(0).Data(8).Group = &H10
StatusBuffer(0).Data(8).Index = &H4
StatusBuffer(0).Data(8).mask = &HFF
StatusBuffer(0).Data(8).type = ACRO_LONG

'Program Flags 1
StatusBuffer(0).Data(9).Group = &H10
StatusBuffer(0).Data(9).Index = &H5
StatusBuffer(0).Data(9).mask = &HFF
StatusBuffer(0).Data(9).type = ACRO_LONG

AcroError = A8_BIN_INIT_STATUS(StatusBuffer(0), 0)
```

Visual C++ Example

```
// packets is how many to put in

    ACR8K_FAST_STATUS pStatusBuffer[1]; //allocate
space
    pStatusBuffer->HowMany = 10;

//Master Distanceinto Move
    pStatusBuffer->Data[0].group= 0x20;
    pStatusBuffer->Data[0].index= 0x00;
    pStatusBuffer->Data[0].mask= 0xff;
    pStatusBuffer->Data[0].type = 1; //floating point
// Master Vector Velocity
    pStatusBuffer->Data[1].group= 0x20;
    pStatusBuffer->Data[1].index= 0x01;
    pStatusBuffer->Data[1].mask= 0xff;
    pStatusBuffer->Data[1].type = 1; //fp
```

A8_BIN_INIT_STATUS, continued

```
Visual C++ Example, //Master Distanceinto Path
Continued          pStatusBuffer->Data[2].group= 0x20;
                    pStatusBuffer->Data[2].index= 0x0f;
                    pStatusBuffer->Data[2].mask= 0xff;
                    pStatusBuffer->Data[2].type = 1; //fp
//Axis Secondary Set Point
                    pStatusBuffer->Data[3].group= 0x30;
                    pStatusBuffer->Data[3].index= 0x07;
                    pStatusBuffer->Data[3].mask= 0xff;
                    pStatusBuffer->Data[3].type = 2; //long

//Axis Following Error
                    pStatusBuffer->Data[4].group= 0x30;
                    pStatusBuffer->Data[4].index= 0x03;
                    pStatusBuffer->Data[4].mask= 0xff;
                    pStatusBuffer->Data[4].type = 2; //long

//General Flags
                    pStatusBuffer->Data[5].group= 0x10;
                    pStatusBuffer->Data[5].index= 0x00;
                    pStatusBuffer->Data[5].mask= 0xff;
                    pStatusBuffer->Data[5].type = 2;

//Master Flags
                    pStatusBuffer->Data[6].group= 0x10;
                    pStatusBuffer->Data[6].index= 0x02;
                    pStatusBuffer->Data[6].mask= 0xff;
                    pStatusBuffer->Data[6].type = 2;

//Axis Flags
                    pStatusBuffer->Data[7].group= 0x10;
                    pStatusBuffer->Data[7].index= 0x03;
                    pStatusBuffer->Data[7].mask= 0xff;
                    pStatusBuffer->Data[7].type = 2;

//Program Flags1
                    pStatusBuffer->Data[8].group= 0x10;
                    pStatusBuffer->Data[8].index= 0x04;
                    pStatusBuffer->Data[8].mask= 0xff;
                    pStatusBuffer->Data[8].type = 2;
//Program Flags2
                    pStatusBuffer->Data[9].group= 0x10;
                    pStatusBuffer->Data[9].index= 0x05;
                    pStatusBuffer->Data[9].mask= 0xff;
                    pStatusBuffer->Data[9].type = 2;
//setup the buffer in the driver
                    A8_BIN_INIT_STATUS(pStatusBuffer, 0);
```

See Also A8_BIN_GET_STATUS

Libraries AcroLibNT

A8_BIN_MASK

Description	Set a register (immediately) on the card using a NAND and an OR mask.
Visual Basic	<i>Declare Sub A8_BIN_MASK(long dataaddress, long nandmask, long ormask, long card);</i>
Visual C++	<i>A8_BIN_MASK(long dataaddress, long nandmask, long ormask, int card);</i>
Returns	Error returned by AcroGetError.
Arguments	dataaddress—address of a long integer storage area on the card. nandmask—mask used to clear bits ormask—mask used to set bits card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	A8_BIN_MASK sets the register immediately. Use AcroSendString to queue a bit setting in the ASCII FIFO on the ACR8000. For example, AcroSendString("P4097",0) will queue the register-setting behind other commands in the ASCII FIFO.
Visual Basic Example	<code>A8_BIN_MASK, &H400C02, 32,1, 0</code>
Visual C++ Example	<code>A8_BIN_MASK(0x400C02,32,1,0)</code>
See Also	A8_BIN_SET, A8_BIN_CLR
Libraries	AcroLibNT

A8_BIN_MOVE_IEEE

Description	Send a floating point Binary move to the card.
Visual Basic	<pre>Declare Sub A8_BIN_MOVE_IEEE Lib "acrownt.dll" (pMove As ACR8K_IEEE_MOVE, ByVal HowMany As Long, ByVal card As Long)</pre>
Visual C++	<pre>void A8_BIN_MOVE_IEEE(ACR8K_IEEE_MOVE pMove, int HowMany,int card);</pre>
Returns	nothing.
Arguments	<p>pMove—pointer to floating point move buffer of type ACR8K_IEEE_MOVE. HowMany—number of moves to send from the move buffer. card—number of the card being used.</p>
Error	Error returned by AcroGetError
Globals	none.
Remarks	Although A8_BIN_MOVE_IEEE can be used to send single binary moves at a time, significant performance gains can be achieved by batching a number of moves together.
Visual Basic Example	<pre>Dim MoveBuffer(10) As ACR8K_IEEE_MOVE For i = 0 To 9 MoveBuffer(i).code0 = &H88 MoveBuffer(i).code1 = &H8 MoveBuffer(i).code2 = &HFF MoveBuffer(i).code3 = &H5 For j = 0 To 7 MoveBuffer(i).moveto(j) = 100 Next j Next i MyError = AcroSetMoveCounter(0, 1, 0) A8_BIN_MOVE_IEEE MoveBuffer(0), 10,0</pre>

A8_BIN_MOVE_IEEE, continued

Visual C++ Example

```
ACR8K_IEEE_MOVE MoveBuffer[10]; // sending 10
moves
// how many to move packets to send
for (i=0; i<10; i++)
{
    MoveBuffer[i].code0 = 0x88; //code0
    MoveBuffer[i].code1 = 0x08; //code1
    MoveBuffer[i].code2 = 0xff; //8 axis moves
    MoveBuffer[i].code3 = 0x05; //incremental
target
    for (j=0; j<8; j++)
    {
        MoveBuffer[i].moveto[j] = 100;
    }
}
A8_BIN_MOVE_IEEE(MoveBuffer, 10, 0);
```

See Also

A8_BIN_MOVE_LONG

Libraries

AcroLibNT

A16_BIN_MOVE_IEEE

Description	Send a 16-axis floating point Binary move to the card. (ACR8020 only)
Visual Basic	<pre>Declare Sub A16_BIN_MOVE_IEEE Lib "acrownt.dll" (pMove As A16_IEEE_MOVE, ByVal HowMany As Long, ByVal card As Long)</pre>
Visual C++	<pre>void A16_BIN_MOVE_IEEE(A16_IEEE_MOVE pMove, int HowMany,int card);</pre>
Returns	nothing.
Arguments	pMove—pointer to floating point move buffer of type A16_IEEE_MOVE. HowMany—number of moves to send from the move buffer. card—number of the card being used.
Error	Error returned by AcroGetError
Globals	none.
Remarks	Although A16_BIN_MOVE_IEEE can be used to send single binary moves at a time, significant performance gains can be achieved by batching a number of moves together.
Visual Basic Example	<pre>Dim MoveBuffer(10) As A16_IEEE_MOVE For i = 0 To 9 MoveBuffer(i).code0 = &H88 MoveBuffer(i).code1 = &H8 MoveBuffer(i).code2 = &HFF MoveBuffer(i).code3 = &H85 MoveBuffer(i).code6 = &HFF For j = 0 To 15 MoveBuffer(i).moveto(j) = 100 Next j Next i MyError = AcroSetMoveCounter(0, 1, 0) A16_BIN_MOVE_IEEE MoveBuffer(0), 10,0</pre>

A16_BIN_MOVE_IEEE, continued

Visual C++ Example

```
A16_IEEE_MOVE MoveBuffer[10]; // sending 10 moves
// how many to move packets to send
for (i=0; i<10; i++)
{
    MoveBuffer[i].code0 = 0x88; //code0
    MoveBuffer[i].code1 = 0x08; //code1
    MoveBuffer[i].code2 = 0xff; //8 axis moves
    MoveBuffer[i].code3 = 0x85; //incremental
    target
    MoveBuffer[i].code6 = 0xFF; //slaves 8-15
    for (j=0; j<15; j++)
    {
        MoveBuffer[i].moveto[j] = 100;
    }
}
A16_BIN_MOVE_IEEE(MoveBuffer, 10, 0);
```

See Also

A16_BIN_MOVE_LONG

Libraries

AcroLibNT

A8_BIN_MOVE_LONG

Description	Send a long integer Binary move to the card.
Visual Basic	<pre>Declare Sub A8_BIN_MOVE_LONG Lib "acrownt.dll" (pMove As ACR8K_LONG_MOVE, ByVal HowMany As Long, ByVal card As Long)</pre>
Visual C++	<pre>void A8_BIN_MOVE_LONG(PACR8K_LONG_MOVE pMove, int HowMany, int card);</pre>
Returns	nothing.
Arguments	<p>pMove—pointer to long move buffer of type ACR8K_LONG_MOVE. HowMany—number of moves to send from the move buffer. card—number of the card being used.</p>
Error	Error returned by AcroGetError
Globals	none.
Remarks	Although A8_BIN_MOVE_LONG can be used to send single binary moves at a time, significant performance gains can be achieved by batching a number of moves together.
Visual Basic Example	<pre>Dim MoveBuffer(10) As ACR8K_LONG_MOVE For i = 0 To 9 MoveBuffer(i).code0 = &H88 MoveBuffer(i).code1 = &H8 MoveBuffer(i).code2 = &HFF MoveBuffer(i).code3 = &H1 For j = 0 To 7 MoveBuffer(i).moveto(j) = 100 Next j Next i MyError = AcroSetMoveCounter(0, 1, 0) A8_BIN_MOVE_LONG MoveBuffer(0), 10,0</pre>

A8_BIN_MOVE_LONG, continued

Visual C++ Example

```
ACR8K_LONG_MOVE MoveBuffer[10]; // sending 10 moves
// how many to move packets to send
for (i=0; i<10; i++)
{
    MoveBuffer[i].code0 = 0x80; //code0
    MoveBuffer[i].code1 = 0x88; //code1
    MoveBuffer[i].code2 = 0x02; //2 axis moves //code2
    MoveBuffer[i].code3 = 0x01; //incremental target
    for (j=0; j<8; j++)
    {
        MoveBuffer[i].moveto[j] = 100;
    }
}
A8_BIN_MOVE_LONG(MoveBuffer, 10, 0);
```

See Also [A8_BIN_MOVE_IEEE](#)

Libraries [AcroLibNT](#)

A16_BIN_MOVE_LONG

Description	Send a 16-axis floating point Binary move to the card. (ACR8020 only)
Visual Basic	<pre>Declare Sub A16_BIN_MOVE_LONG Lib "acrownt.dll" (pMove As A16_LONG_MOVE, ByVal HowMany As Long, ByVal card As Long)</pre>
Visual C++	<pre>void A16_BIN_MOVE_LONG(A16_LONG_MOVE pMove, int HowMany,int card);</pre>
Returns	nothing.
Arguments	pMove—pointer to floating point move buffer of type A16_LONG_MOVE. HowMany—number of moves to send from the move buffer. card—number of the card being used.
Error	Error returned by AcroGetError
Globals	none.
Remarks	Although A16_BIN_MOVE_LONG can be used to send single binary moves at a time, significant performance gains can be achieved by batching a number of moves together.
Visual Basic Example	<pre>Dim MoveBuffer(10) As A16_LONG_MOVE For i = 0 To 9 MoveBuffer(i).code0 = &H88 MoveBuffer(i).code1 = &H8 MoveBuffer(i).code2 = &HFF MoveBuffer(i).code3 = &H85 MoveBuffer(i).code6 = &HFF For j = 0 To 15 MoveBuffer(i).moveto(j) = 100 Next j Next i MyError = AcroSetMoveCounter(0, 1, 0) A16_BIN_MOVE_LONG MoveBuffer(0), 10,0</pre>

A16_BIN_MOVE_LONG, continued

Visual C++ Example

```
A16_LONG_MOVE MoveBuffer[10]; // sending 10 moves
// how many to move packets to send
for (i=0; i<10; i++)
{
    MoveBuffer[i].code0 = 0x88; //code0
    MoveBuffer[i].code1 = 0x08; //code1
    MoveBuffer[i].code2 = 0xff; //8 axis moves
    MoveBuffer[i].code3 = 0x85; //incremental
    target
    MoveBuffer[i].code6 = 0xFF; //slaves 8-15
    for (j=0; j<15; j++)
    {
        MoveBuffer[i].moveto[j] = 100;
    }
}
A16_BIN_MOVE_LONG(MoveBuffer, 10, 0);
```

See Also

A16_BIN_MOVE_LONG

Libraries

AcroLibNT

A8_BIN_PEEK_IEEE

Description	Read a user-defined floating point variable from the card.
Visual Basic	<pre>Declare Function A8_BIN_PEEK_IEEE Lib "acrownt.dll" (ByVal conversioncode As Long, ByVal peekwordcount As Long, ByVal peekaddress As Long, pReturnIEEE As Single, ByVal card As Long) As Long</pre>
Visual C++	<pre>int A8_BIN_PEEK_IEEE(long conversioncode, long peekwordcount, unsigned long peekaddress, float* pReturnIEEE, int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	conversioncode—LONG =0x00, FP64=0x01, FP32=0x02 peekwordcount—number of words to peek. peekaddress—address for peeking retrieved with A8_BIN_ADDRESS (except global user variables) pReturnIEEE—pointer to an array to contain the results of the peek. card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	This command is used in conjunction with A8_BIN_ADDRESS, to read user-defined variables from the card. When peeking values from the card, no more than 50 elements should be retrieved in a single call. For more details see the Binary Hostinterface Section in the Appendix C.
Visual Basic Example	<pre>Dim MyError As Long Dim pReturn(2) As Single 'read 2 floating point variables whose base address is 10000 MyError=A8_BIN_PEEK_IEEE (2, 2, 10000,pReturn(0),0)</pre>
Visual C++ Example	<pre>int MyError; float pReturn[2]; // read 2 floating point variables whose base address is 10000 MyError=A8_BIN_PEEK_IEEE (2, 2, 10000,pReturn,0);</pre>
See Also	A8_BIN_ADDRESS, A8_BIN_PEEK_LONG
Libraries	AcroLibNT

A8_BIN_PEEK_LONG

Description	Read a user-defined long variable from the card.
Visual Basic	<pre>Declare Function A8_BIN_PEEK_LONG Lib "acrownt.dll" (ByVal conversioncode As Long, ByVal peekwordcount As Long, ByVal peekaddress As Long, pReturnLong As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>int A8_BIN_PEEK_LONG(long conversioncode, long peekwordcount, unsigned long peekaddress, long* pReturnLong,int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	conversioncode—LONG =0x00, FP64=0x01,FP32=0x02 peekwordcount—number of words to peek. peekaddress—address for peeking retrieved with A8_BIN_ADDRESS (except global user variables) pReturnLong—pointer to an array to contain the results of the peek. card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	This command is used in conjunction with A8_BIN_ADDRESS, to read user-defined variables from the card. When peeking values from the card, no more than 50 elements should be retrieved in a single call. For more details see the Binary Hostinterface Section in the Appendix C.
Visual Basic Example	<pre>Dim MyError As Long Dim pReturn(2) As Long 'read 2 long variables whose base address is 10000 MyError=A8_BIN_PEEK_LONG (0, 2, 10000,pReturn(0),0)</pre>
Visual C++ Example	<pre>int myerror; long pReturn[2]; // read 2 long variables whose base address is 10000 MyError=A8_BIN_PEEK_LONG (0, 2, 10000,pReturn,0);</pre>
See Also	A8_BIN_ADDRESS, A8_BIN_PEEK_IEEE
Libraries	AcroLibNT

A8_BIN_POKE_IEEE

Description	Change a floating point user-defined variable on the card.
Visual Basic	<pre>Declare Function A8_BIN_POKE_IEEE Lib "acrownt.dll" (ByVal conversioncode As Long, ByVal pokewordcount As Long, ByVal pokeaddress As Long, pSendIEEE As Single, ByVal card As Long) As Long</pre>
Visual C++	<pre>int A8_BIN_POKE_IEEE(long conversioncode,long pokewordcount,unsigned long pokeaddress, float* pSendIEEE,int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	conversioncode—LONG =0x00, FP64=0x01,FP32=0x02 pokewordcount—number of words to poke. pokeaddress—address for poking retrieved with A8_BIN_ADDRESS. (except global user variables) pSendIEEE—pointer to an array to containing the data for poking. card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	This command is used in conjunction with A8_BIN_ADDRESS, to write to user-defined variables on the card. Note: When using A8_BIN_ADDRESS for a user-defined variable, increment the address returned by one before poking because the address thus returned contains the number of variables of that type dimensioned followed by the actual variables. When poking values to the card, no more than 50 elements should be sent in a single call. For more details see the Binary Host Interface Section in the Appendix C.
Visual Basic Example	<pre>Dim MyError As Long Dim pSendIEEE(2) As Single pSendIEEE(0)=10.99 pSendIEEE(1)=20.876776 'write 2 floating point variables to base address 10000 MyError=A8_BIN_POKE_IEEE (2, 2, 10000,pSendLong(0),0)</pre>
Visual C++ Example	<pre>int myerror; float pSendIEEE[2]; pSendIEEE[0] = -20.0345; pSendIEEE[1]=34.666; // write 2 floating point variables whose base address is 10000 MyError=A8_BIN_POKE_IEEE (2, 2, 10000,pSendLong,0);</pre>
See Also	A8_BIN_ADDRESS, A8_BIN_POKE_LONG
Libraries	AcroLibNT

A8_BIN_POKE_LONG

Description	Change a user-define long variable on the card.
Visual Basic	<pre>Declare Function A8_BIN_POKE_LONG Lib "acrownt.dll" (ByVal conversioncode As Long, ByVal pokewordcount As Long, ByVal pokeaddress As Long, pSendLong As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>int A8_BIN_POKE_LONG(long conversioncode, long pokewordcount, unsigned long pokeaddress, long* pSendLong, int card);</pre>
Returns	Error returned by AcroGetError.
Arguments	<p>conversioncode—LONG =0x00, FP64=0x01, FP32=0x02</p> <p>pokewordcount—number of words to poke.</p> <p>pokeaddress—address for poking retrieved with A8_BIN_ADDRESS. (except global user variables)</p> <p>pSendLong—pointer to an array containig the data for poking.</p> <p>card—number of the card being used</p>
Error	Error returned by AcroGetError
Globals	none.
Remarks	This command is used in conjunction with A8_BIN_ADDRESS, to write to user-defined variables on the card. Note: When using A8_BIN_ADDRESS for a user-defined variable, increment the address returned by one before poking because the address thus returned contains the number of variables of that type dimensioned followed by the actual variables. When poking values to the card, no more than 50 elements should be sent in a single call. For more details see the Binary Hostinterface Section in the Appendix C.
Visual Basic Example	<pre>Dim MyError As Long Dim pSendLong(2) As Long pSendLong(0)=10 pSendLong(1)=20 'write 2 long variables to base address 10000 MyError=A8_BIN_POKE_LONG (0, 2, 10000,pSendLong(0),0)</pre>
Visual C++ Example	<pre>int myerror; long pSendLong[2]; pSendLong[0] = -20; pSendLong[1]=34; // read 2 long variables whose base address is 10000 MyError=A8_BIN_POKE_LONG (0, 2, 10000,pSendLong,0);</pre>
See Also	A8_BIN_ADDRESS, A8_BIN_POKE_IEEE
Libraries	AcroLibNT

A8_BIN_ROV

Description	Send an immediate ROV to master(s) on the card.
Visual Basic	<pre>Declare Sub A8_BIN_ROV Lib "acrownt.dll" (ByVal mastermask As Integer, ByVal fovvalue As Single, ByVal card As Long)</pre>
Visual C++	<pre>void A8_BIN_ROV(short mastermask, float fovvalue,int card);</pre>
Returns	nothing.
Arguments	mastermask—mask specifying masters for which ROV will be activated. fovvalue—value of the rapid override. card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	
Visual Basic Example	<pre>'set the ROV to 10 times for master0. A8_BIN_ROV 1,10,0</pre>
Visual C++ Example	<pre>//set the ROV to 10 times for master0. A8_BIN_ROV(1,10,0);</pre>
See Also	
Libraries	AcroLibNT

A16_BIN_ROV

Description	Send an immediate ROV to 16 master(s) on the card.
Visual Basic	<i>Declare Sub A16_BIN_ROV Lib "acrownt.dll" (ByVal mastermask As long, ByVal fovvalue As Single, ByVal card As Long)</i>
Visual C++	<i>void A16_BIN_ROV(long mastermask, float fovvalue, int card);</i>
Returns	nothing.
Arguments	mastermask—mask specifying masters for which ROV will be activated. fovvalue—value of the feedrate overried. card—number of the card being used.
Error	Error returned by AcroGetError.
Globals	none.
Remarks	ACR8020 only
Visual Basic Example	<pre>'set the fov to 10 times for master 0 A16_BIN_ROV 1,10,0</pre>
Visual C++ Example	<pre>//set the ROV to 10 times for master0 A16_BIN_ROV(1,10,0);</pre>
See Also	
Libraries	AcroLibNT

A8_BIN_SET

Description	Set a bit (immediately) on the card.
Visual Basic	<i>Declare Sub A8_BIN_SET Lib "acrownt.dll" (ByVal Flg As Integer, ByVal card As Long</i>
Visual C++	<i>void A8_BIN_SET(short Flg,int card);</i>
Returns	nothing.
Arguments	Flag—number of flag to be set on the card. card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	A8_BIN_SET sets the bit immediately. Use AcroSendString to queue a bit setting in the ASCII FIFO on the ACR8000. For example, AcroSendString("SET32",0) will queue the bit-setting behind other commands in the ASCII FIFO.
Visual Basic Example	<pre>'set output 32 A8_BIN_SET 32,0</pre>
Visual C++ Example	<pre>//set output 32 A8_BIN_SET(32,0)</pre>
See Also	AcroSendString, A8_BIN_CLR
Libraries	AcroLibNT

A8_BIN_SETLONG

Description	Set a single long parameter on the card.
Visual Basic	<i>Declare Function A8_BIN_SETLONG Lib "acrownt.dll" (ByVal parameterindex As Long, ByVal value As Long, ByVal card As Long) As Long</i>
Visual C++	<i>int A8_BIN_SETLONG(long parameterindex, long value, int card);</i>
Returns	Error returned by AcroGetError.
Arguments	parameterindex—parameter number on the card from Appendix A or Appendix B. value—the value for the parameter. card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	A8_BIN_SETLONG sets the parameter immediately. Use AcroSendString to queue parameter setting in the ASCII FIFO on the ACR8000. For example, AcroSendString("P0=10",0) will queue the parameter setting behind other commands in the ASCII FIFO.
Visual Basic Example	<pre>Dim MyError As Long 'P4097=-10 MyError= A8_BIN_SETLONG(4097,-10,0)</pre>
Visual C++ Example	<pre>int MyError; //SET P4100=-100 MyError =A8_BIN_SETLONG(4100,-100,0);</pre>
See Also	A8_BIN_SETIEEEE,A8_BIN_POKE_LONG
Libraries	AcroLibNT

A8_BIN_SETIEEE

Description	Change the value of a floating point variable on the ard.
Visual Basic	<i>Declare Function A8_BIN_SETIEEE Lib "acrownt.dll" (ByVal parameterindex As Long, ByVal value As Single, ByVal card As Long) As Long</i>
Visual C++	<i>int A8_BIN_SETIEEE (long parameterindex, float value, int card);</i>
Returns	Error returned by AcroGetError.
Arguments	parameterindex—parameter number on the card from Appendix A or Appendix B. value—the value for the parameter. card—number of the card being used
Error	Error returned by AcroGetError.
Globals	none.
Remarks	A8_BIN_SETIEEE sets the parameter immediately. Use AcroSendString to queue parameter setting in the ASCII FIFO on the ACR8000. For example, AcroSendString("P12304=0.002",0) will queue the parameter setting behind other commands in the ASCII FIFO.
Visual Basic Example	<pre>Dim MyError As Long 'set proportional gain for axis0 to 0.002 MyError= A8_BIN_SETIEEE(12304,0.002,0)</pre>
Visual C++ Example	<pre>int MyError; //set integral Gain for axis0 to 0.0001 MyError = A8_BIN_SETIEEE(12305,0.0001,0);</pre>
See Also	A8_BIN_SETLONG, A8_BIN_POKE_IEEE
Libraries	AcroLibNT

A8_BIN_SYS_MASK

Description	Set a register (immediately) on the card using a NAND and an OR mask.
Visual Basic	<pre>Declare Sub A8_BIN_SYS_MASK Lib "acrownt.dll" (ByVal parameterindex As Long, ByVal nandmask As Long, ByVal ormask As Long, ByVal card As long)</pre>
Visual C++	<pre>void A8_BIN_SYS_MASK(long parameterindex, long nandmask, long ormask, int card);</pre>
Returns	nothing.
Arguments	parameterindex—parameter number on the card nandmask—mask used to clear bits ormask—mask used to set bits card—number of the card being used
Error	Error returned by AcroGetError
Globals	none.
Remarks	A8_BIN_SYS_MASK sets the register immediately. Use AcroSendString to queue a bit setting in the ASCII FIFO on the ACR8000. For example, AcroSendString("P4097",0) will queue the register-setting behind other commands in the ASCII FIFO.
Visual Basic Example	<pre>A8_BIN_SYS_MASK, 4156, 32, 0, 0</pre>
Visual C++ Example	<pre>A8_BIN_SYS_MASK(4156, 32, 0, 0)</pre>
See Also	AcroSendString, A8_BIN_CLR, A8_BIN_MASK, A8_BIN_SET
Libraries	AcroLibNT

ACR8K_FAST_STATUS

Description	This structure defines the buffer used for fast status access.
Visual Basic	<pre>Type ACR8K_FAST_STATUS HowMany As Integer Flags As Integer Data(0 To MAX_STATUS_REQUEST) As ACR8K_FAST_STATUS_REQ End Type</pre>
Visual C++	<pre>typedef struct _ACR8K_FAST_STATUS { short HowMany; short Flags; ACR8K_FAST_STATUS_REQ Data[MAX_STATUS_REQUEST]; } ACR8K_FAST_STATUS, *PACR8K_FAST_STATUS;</pre>
Returns	N/A
Arguments	HowMany—number of fast statuses to fetch. Flags—reserved. Data—an array of fast status requests (ACR8K_FAST_STATUS_REQ)
Error	N/A
Globals	N/A
Remarks	N/A
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	AcroLibNT

ACR8K_FAST_STATUS_REQ

Description	This structure defines a single status request for fast status buffer access. Not to be used except as a component of ACR8K_FAST_STATUS.
Visual Basic	<pre>Type ACR8K_FAST_STATUS_REQ Group As Byte Index As Byte mask As Byte type As Integer lng(0 To 7) As Long ulng(0 To 7) As Long fp(0 To 7) As Single End Type</pre>
Visual C++	<pre>typedef struct _ACR8K_FAST_STATUS_REQ{ unsigned char group; unsigned char index; unsigned char mask; short type; // 1 = fp, 2 = lng long lng[8]; unsigned long ulng[8]; float fp[8]; } ACR8K_FAST_STATUS_REQ, *PACR8K_FAST_STATUS_REQ;</pre>
Returns	N/A
Arguments	group—group code from Appendix A or Appendix B. index—index code from Appendix A or Appendix B. mask—mask indicating which parameters to fetch. type—FP32=1, LONG=2 lng—array returning a long result. ulng—reserved. fp—array returning a FP32 result.
Error	N/A
Globals	N/A
Remarks	Not to be used individually.
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	AcroLibNT

ACR8K_IEEE_MOVE

Description This structure defines a floating point move.

Visual Basic

```
Type ACR8K_IEEE_MOVE
code0 As Byte
code1 As Byte
code2 As Byte
code3 As Byte
mastervel As Single
masteraccdec As Single
moveto(0 To 7) As Single
primarycenter As Single
secondarycenter As Single
primaryscaling As Single
secondaryscaling As Single
End Type
```

Visual C++

```
typedef struct _ACR8K_IEEE_MOVE{
unsigned char code0;
unsigned char code1;
unsigned char code2;
unsigned char code3;
float mastervel;
float masteraccdec;
float moveto[8];
float primarycenter;
float secondarycenter;
float primaryscaling;
float secondaryscaling;
} ACR8K_IEEE_MOVE,*PACR8K_IEEE_MOVE;
```

Returns N/A

Arguments

code0—see the Binary Move Command in Appendix C for a description.

code1— see the Binary Move Command in Appendix C for a description

code2— see the Binary Move Command in Appendix C for a description

code3— see the Binary Move Command in Appendix C for a description

mastervel—master velocity.

masteraccdec—master acceleration and deceleration.

moveto—target points for the move.

primarycenter—see Binary Move Command in Appendix C for a description.

secondarycenter— see Binary Move Command in Appendix C for a description.

primaryscaling— see Binary Move Command in Appendix C for a description.

secondaryscaling— see Binary Move Command in Appendix C for a description.

ACR8K_IEEE_MOVE, continued

Error	N/A
Globals	N/A
Remarks	See the Binary Host Interface Section in the Appendix C for a detailed description of the binary move command.
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	N/A

A16_IEEE_MOVE

Description This structure defines a 16-axis floating point move.

Visual Basic

```
Type A16_LONG_MOVE
code0 As Byte
code1 As Byte
code2 As Byte
code3 As Byte
code4 As Byte
code5 As Byte
code6 As Byte
code7 As Byte
mastervel As Single
masteraccdec As Single
moveto(0 To 15) As Single
primarycenter As Long
secondarycenter As Long
primaryscaling As Long
secondaryscaling As Long
End Type
```

Visual C++

```
typedef struct _A16_IEEE_MOVE{
unsigned char code0;
unsigned char code1;
unsigned char code2;
unsigned char code3;
unsigned char code4;
unsigned char code5;
unsigned char code6;
unsigned char code7;
float mastervel;
float masteraccdec;
float moveto[16];
long primarycenter;
long secondarycenter;
float primaryscaling;
float secondaryscaling;
} ACR8K_IEEE_MOVE,*PACR8K_IEEE_MOVE;
```

Returns N/A

Arguments

code0—see the Binary Move Command in Appendix C for a description.

code1— see the Binary Move Command in Appendix C for a description

code2— see the Binary Move Command in Appendix C for a description.

code3— see the Binary Move Command in Appendix C for a description.

code4— see the Binary Move Command in Appendix C for a description.

code5— see the Binary Move Command in Appendix C for a description.

code6— see the Binary Move Command in Appendix C for a description.

A16_IEEE_MOVE, continued

code7- see the Binary Move Command in Appendix C for a description.
mastervel-master velocity.
masteraccdec-master acceleration and deceleration.
moveto-target points for the move.
primarycenter-see Binary Move Command in Appendix C for a description.
secondarycenter- see Binary Move Command in Appendix C for a description.
primaryscaling- see Binary Move Command in Appendix C for a description.
secondaryscaling- see Binary Move Command in Appendix C for a description.

Error	N/A
Globals	N/A
Remarks	(ACR8020 only) See the Binary Hostinterface Section in the Appendix C for a detailed description of the binary move command.
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	N/A

ACR8K_LONG_MOVE

Description	This structure defines a long move.
Visual Basic	<pre>Type ACR8K_LONG_MOVE code0 As Byte code1 As Byte code2 As Byte code3 As Byte mastervel As Single masteraccdec As Single moveto(0 To 7) As Long primarycenter As Long secondarycenter As Long primaryscaling As Long secondaryscaling As Long End Type</pre>
Visual C++	<pre>typedef struct _ACR8K_LONG_MOVE{ unsigned char code0; unsigned char code1; unsigned char code2; unsigned char code3; float mastervel; float masteraccdec; long moveto[8]; long primarycenter; long secondarycenter; float primaryscaling; float secondaryscaling; } ACR8K_LONG_MOVE,*PACR8K_LONG_MOVE;</pre>
Returns	N/A
Arguments	<p>code0—see the Binary Move Command in Appendix C for a description.</p> <p>code1— see the Binary Move Command in Appendix C for a description</p> <p>code2— see the Binary Move Command in Appendix C for a description.</p> <p>code3— see the Binary Move Command in Appendix C for a description.</p> <p>mastervel—master velocity.</p> <p>masteraccdec—master acceleration and deceleration.</p> <p>moveto—target points for the move.</p> <p>primarycenter—see Binary Move Command in Appendix C for a description.</p> <p>secondarycenter— see Binary Move Command in Appendix C for a description.</p> <p>primaryscaling— see Binary Move Command in Appendix C for a description.</p> <p>secondaryscaling— see Binary Move Command in Appendix C for a description.</p>

ACR8K_LONG_MOVE, continued

Error	N/A
Globals	N/A
Remarks	See the Binary Hostinterface Section in the Appendix C for a detailed description of the binary move command.
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	N/A

A16_LONG_MOVE

Description This structure defines a 16-axis long move.

Visual Basic

```
Type A16_LONG_MOVE
code0 As Byte
code1 As Byte
code2 As Byte
code3 As Byte
code4 As Byte
code5 As Byte
code6 As Byte
code7 As Byte
mastervel As Single
masteraccdec As Single
moveto(0 To 15) As Long
primarycenter As Long
secondarycenter As Long
primaryscaling As Long
secondaryscaling As Long
End Type
```

Visual C++

```
typedef struct _A16_LONG_MOVE{
unsigned char code0;
unsigned char code1;
unsigned char code2;
unsigned char code3;
unsigned char code4;
unsigned char code5;
unsigned char code6;
unsigned char code7;
float mastervel;
float masteraccdec;
long moveto[16];
long primarycenter;
long secondarycenter;
float primaryscaling;
float secondaryscaling;
} ACR8K_LONG_MOVE,*PACR8K_LONG_MOVE;
```

Returns N/A

Arguments

code0—see the Binary Move Command in Appendix C for a description.

code1— see the Binary Move Command in Appendix C for a description

code2— see the Binary Move Command in Appendix C for a description.

code3— see the Binary Move Command in Appendix C for a description.

code4— see the Binary Move Command in Appendix C for a description.

code5— see the Binary Move Command in Appendix C for a description.

code6— see the Binary Move Command in Appendix C for a description.

A16_LONG_MOVE, continued

code7- see the Binary Move Command in Appendix C for a description.
mastervel-master velocity.
masteraccdec-master acceleration and deceleration.
moveto-target points for the move.
primarycenter-see Binary Move Command in Appendix C for a description.
secondarycenter- see Binary Move Command in Appendix C for a description.
primaryscaling- see Binary Move Command in Appendix C for a description.
secondaryscaling- see Binary Move Command in Appendix C for a description.

Error	N/A
Globals	N/A
Remarks	(ACR8020 Only) See the Binary Hostinterface Section in the Appendix C for a detailed description of the binary move command.
Visual Basic Example	N/A
Visual C++ Example	N/A
See Also	N/A
Libraries	N/A

AcroBinaryBufferEmpty

Description	Returns a 1 if there are no elements in the Text Buffer. Otherwise, returns a 0.
Visual Basic	<pre>Declare Function AcroBinaryBufferEmpty Lib "acrownt.dll" (ByVal card As long) As Long</pre>
Visual C++	<pre>int AcroBinaryBufferEmpty(int card);</pre>
Returns	Returns a 1 if there are no elements in the Text Buffer. Otherwise, returns a 0.
Arguments	card—number of the card being used
Error	none.
Globals	none.
Remarks	This command is often useful when implementing a text terminal.
Visual Basic Example	<pre>Dim Flag as Long 'check if text buffer is empty for card 0. Flag = AcroBinaryBufferEmpty(0)</pre>
Visual C++ Example	<pre>int Flag ; //check if text buffer is empty for card 0. Flag = AcroBinaryBufferEmpty(0)</pre>
See Also	
Libraries	AcroLibNT

AcroBinaryBufferNumberOfElements

Description	Returns the number of elements (bytes) in the Binary Buffer.
Visual Basic	<i>Declare Function AcroBinaryBufferNumberOfElements Lib "acrownt.dll" (ByVal card As long) As Long</i>
Visual C++	<i>int AcroBinaryBufferNumberOfElements(int card);</i>
Returns	Returns the number of elements (bytes) in the Binary Buffer.
Arguments	card—number of the card being used
Error	none.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim Elements as long Elements= AcroBinaryBufferNumberOfElements(0)</pre>
Visual C++ Example	<pre>long elements; elements = AcroBinaryBufferNumberOfElements(0)</pre>
See Also	
Libraries	AcroLibNT

AcroCardPresent

Description	Tests to see if the card is present by sending out an empty binary request and receiving it back. Returns 1 if the card is present, otherwise returns a 0.
Visual Basic	<i>Declare Function AcroCardPresent Lib "acrownt.dll" (ByVal card As Long) As Long</i>
Visual C++	<i>int AcroCardPresent(int card);</i>
Returns	Returns 1 if the card is present, otherwise returns a 0.
Arguments	card—number of the card being used
Error	AcroGetError returns the error.
Globals	none.
Remarks	This functions should only be called AFTER AcroInitialize has been called successfully atleast once.
Visual Basic Example	<pre>Dim CardPresent As Long CardPresent = AcroCardPresent (0)</pre>
Visual C++ Example	<pre>int cardpresent; cardpresent=AcroCardPresent(0);</pre>
See Also	
Libraries	AcroLibNT

AcroGetBinaryTimeOut

Description	Returns the timeout used by RBIN service to wait on the hardware port before timing out. Default it 10 microseconds.
Visual Basic	<pre>Declare Function AcroGetBinaryTimeOut Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetBinaryTimeOut(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned timeout value in microseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyErrorAs Long MyError=AcroGetBinaryTimeOut (pDelta,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; long pDelta[1]; MyError=AcroGetBinaryTimeOut (pDelta,0);</pre>
See Also	AcroSetBinaryTimeOut
Libraries	AcroLibNT

AcroGetBinaryTimerDelta

Description	Gets the current period (in msec) of the BinaryTimer associated with RBIN service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetBinaryTimerDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetBinaryTimerDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned timeout value in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long ' Read the Binary TimerDelta into a long variable pDelta MyError = AcroGetBinaryTimerDelta (pDelta, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // Read the Binary TimerDelta into a long variable pDelta unsigned long* pDelta; MyError= AcroGetBinaryTimerDelta(pDelta,0);</pre>
See Also	AcroSetBinaryTimerDelta
Libraries	AcroLibNT

AcroGetBinTransThreshold

Description	Gets the threshold to control the fast status fetch. Default is 0.
Visual Basic	<pre>Declare Function AcroGetBinTransThreshold Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetBinTransThreshold(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	<p>pDelta—returned the binary threshold used to determine how fast status requests are slowed down in the presence of binary moves.</p> <p>card—number of the card being used.</p>
Error	Error returned by GetLastError.
Globals	none.
Remarks	none.
Visual Basic Example	<pre>'Read the threshold Dim AcroError as long Dim Delta as long AcroError = AcroGetBinTransThreshold(Delta, 0)</pre>
Visual C++ Example	<pre>//Read threshold unsigned long threshold; AcroGetBinTransThreshold(&threshold, 0);</pre>
See Also	
Libraries	AcroLibNT

AcroGetDrBinaryBufferState

Description	Returns the state of the Binary Buffer.
Visual Basic	<pre>Declare Function AcroGetDrBinaryBufferStateCh lib "acrownt.dll" (pState As String, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrBinaryBufferStateCh(char *pbinarystate, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pbinarystate—internal state of the binary buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	This function is intended for monitoring and debugging support only. The states are as follows: INACTIVE=1 GETGROUP=2 GETINDEX=6 GETMASK=7 WRITEHEADER=8 GETBODY=3 WHATNEXT=4 TIMEOUT=5
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pState Long 'Read the state of the Binary Buffer MyError=AcroGetDrBinaryBufferState (pState, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pState; // Read the state of the Binary Buffer MyError=AcroGetDrBinaryBufferState(pState, 0);</pre>
See Also	
Libraries	AcroLibNT

AcroGetDrBinaryBuffLength

Description	Returns the maximum size (in bytes) of BIN READ buffer. Default size is 4*4096 bytes on i386.
Visual Basic	<pre>Declare Function AcroGetDrBinaryBuffLength Lib "acrownt.dll" (pNumBytes As Long, ByVal card long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrBinaryBuffLength(unsigned As Long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—maximum number of bytes for the binary buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Maximum size of buffers is not user changeable and is in multiples of page size (4*4096 for i386) of the machine.
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the maximum number of characters in the BIN READ buffer MyError=AcroGetDrBinaryBuffLength (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the number of bytes in the BIN READ buffer MyError=AcroGetDrBinaryBuffLength(pNumBytes, 0);</pre>
See Also	AcroGetDrWriteBuffLength, AcroGetDrReadBuffLength.
Libraries	AcroLibNT

AcroGetDrBinaryBuffSize

Description	Returns the number of bytes in the BIN READ buffer.
Visual Basic	<pre>Declare Function AcroGetDrBinaryBuffSize Lib "acrownt.dll" (pNumBytes As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrBinaryBuffSize(unsigned long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—number of bytes in the binary buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the number of characters in the BIN READ buffer MyError=AcroGetDrBinaryBuffSize (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the number of bytes in the BIN READ buffer MyError=AcroGetDrBinaryBuffSize(pNumBytes, 0);</pre>
See Also	AcroGetDrWriteBuffSize, AcroGetDrReadBuffSize.
Libraries	AcroLibNT

AcroGetDrReadBuffLength

Description	Returns the maximum size (in bytes) of the READ TXT buffer. Default size is 4096 bytes on i386.
Visual Basic	<pre>Declare Function AcroGetDrReadBuffLength Lib "acrownt.dll" (pNumBytes As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrReadBuffLength(unsigned long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—maximum number of bytes for the read buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Maximum size of buffers is not user changeable and is in multiples of page size (4096 for i386) of the machine.
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the maximum number of characters in the READ TXT buffer MyError=AcroGetDrReadBuffLength (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the maximum number of bytes in the READ TXT buffer MyError=AcroGetDrReadBuffLength(pNumBytes, 0);</pre>
See Also	AcroGetDrWriteBuffLength, AcroGetDrBinaryBuffLength.
Libraries	AcroLibNt

AcroGetDrReadBuffSize

Description	Returns the number of bytes in the READ TXT buffer.
Visual Basic	<pre>Declare Function AcroGetDrReadBuffSize Lib "acrownt.dll" (pNumBytes As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrReadBuffSize(unsigned long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—number of bytes in the read buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Same as AcroTextBufferNumberOfElements.
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the number of characters in the READ TXT buffer MyError=AcroGetDrReadBuffSize (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the number of bytes in the READ TXT buffer MyError=AcroGetDrReadBuffSize(pNumBytes, 0);</pre>
See Also	AcroGetDrWriteBuffSize, AcroGetDrBinaryBuffSize.
Libraries	AcroLibNT

AcroGetDrWriteBuffLength

Description	Returns the maximum size (in bytes) of the WRITE buffer. Default size is 4*4096 bytes on i386.
Visual Basic	<pre>Declare Function AcroGetDrWriteBuffLength Lib "acrownt.dll" (pNumBytes As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrWriteBuffLength(unsigned long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—maximum number of bytes for the read buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Maximum size of buffers is not user changeable and is in multiples of page size (4096 for i386) of the machine.
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the maximum number of characters in the WRITE buffer MyError=AcroGetDrWriteBuffLength (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the maximum number of bytes in the WRITE buffer MyError=AcroGetDrWriteBuffLength(pNumBytes, 0);</pre>
See Also	AcroGetDrReadBuffLength, AcroGetDrBinaryBuffLength.
Libraries	AcroLibNT

AcroGetDrWriteBuffSize

Description	Returns the number of bytes in the WRITE buffer.
Visual Basic	<pre>Declare Function AcroGetDrWriteBuffSize Lib "acrownt.dll" (pNumBytes As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetDrWriteBuffSize(unsigned long *pNumBytes, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pNumBytes—number of bytes in the binary buffer. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pNumBytes As Long 'Read the number of characters in the WRITE buffer MyError=AcroGetDrWriteBuffSize (pNumBytes, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pNumBytes[1]; // Read the number of bytes in the WRITE buffer MyError=AcroGetDrWriteBuffSize(pNumBytes, 0);</pre>
See Also	AcroGetDrReadBuffSize, AcroGetDrBinaryBuffSize.
Libraries	AcroLibNT

AcroGetError

Description	Return an error code for the last procedure called. See Errors for a description of returned errors.
Visual Basic	<pre>Declare Function AcroGetError Lib "acrownt.dll" () As Long</pre>
Visual C++	<pre>long AcroGetError();</pre>
Returns	AcroLib error code.
Arguments	none.
Error	none
Globals	none
Remarks	This function should be called after all critical function calls to make sure that an error has not occurred. Error code must be explicitly set back to ACRO_SUCCESS using AcroSetError() after AcroGetError() has been called. The reason for this is the error code does not get reset automatically.
Visual Basic Example	<pre>Dim MyError as Long MyError = AcroGetError()</pre>
Visual C++ Example	<pre>long MyError; MyError = AcroGetError();</pre>
See Also	AcroSetError
Libraries	AcroLibNT

AcroGetFastStatusTimerDelta

Description	Gets the current period (in msec) of the FastStatusTimer in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetFastStatusTimerDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetFastStatusTimerDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'Read the FastStatus TimerDelta into a long variable pDelta MyError = AcroGetFastStatusTimerDelta (pDelta, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long Delta; // Read the FastStatus TimerDelta into a long variable pDelta. MyError= AcroGetFastStatusTimerDelta(&Delta,0);</pre>
See Also	AcroSetFastStatusTimerDelta
Libraries	AcroLibNT

AcroGetMaxBinTransPerFSIntr

Description	Gets the maximum binary transactions expected during a fast status periodinterval.
Visual Basic	<i>Declare Function AcroGetMaxBinTransPerFSIntr Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</i>
Visual C++	<i>unsigned long AcroGetMaxBinTransPerFSIntr(unsigned long *pDelta,int card);</i>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned number of binary transaction expected per period of the fast status access. card—number of the card being used
Error	Error returned by GetLastError.
Globals	none.
Remarks	This is used in conjunction with AcroSetMaxFastStatusDelta to control the fast status requests.
Visual Basic Example	<pre>Dim maxbintrans as long Dim AcroError as long AcroError = AcroGetMaxBinTransPerFSIntr(maxbintr, 0)</pre>
Visual C++ Example	<pre>unsigned long maxbintr; AcroGetMaxBinTransPerFSIntr(&maxbintr,0);</pre>
See Also	
Libraries	AcroLibNT

AcroGetMaxFastStatusDelta

Description	Gets the maximum period (in msec) for the fast status fetch.
Visual Basic	<pre>Declare Function AcroGetMaxFastStatusDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetMaxFastStatusDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned maximum period in milliseconds. card—number of the card being used
Error	Error returned by GetLastError.
Globals	none.
Remarks	none.
Visual Basic Example	<pre>'Read the max fast status delta Dim AcroError as long Dim Delta as long AcroError = AcroGetMaxFastStatusDelta(Delta, 0)</pre>
Visual C++ Example	<pre>//Read the max fast status delta unsigned long maxdelta; unsigned long Err; Err = AcroGetMaxFastStatusDelta(&maxdelta, 0);</pre>
See Also	
Libraries	AcroLibNT

AcroGetMoveCounter

Description	Gets the current value of the move counter.
Visual Basic	<pre>Declare Function AcroGetMoveCounter Lib "acrownt.dll" (pcounter As Long, pIncrement As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetMoveCounter(long* pcounter, long* pincrement, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pcounter—returned value of the move counter. pincrement—returned increment for the move counter card—number of the card being used
Error	Error returned by GetLastError.
Globals	none.
Remarks	This function can be used in conjunction with the interrupt generation capability of move packets to coordinate between the user application and the card. For example, it can be used to keep track of which move is currently executing on the card.
Visual Basic Example	<pre>Dim MyError As Long Dim MoveCounter as Long MyError = AcroGetMoveCounter(MoveCounter, 0)</pre>
Visual C++ Example	<pre>long pMoveCounter[1] unsigned long MyError; MyError= AcroGetMoveCounter(pMoveCounter, 0);</pre>
See Also	AcroSetMoveCounter
Libraries	AcroLibNT

AcroGetPerformance

Description	Retrieves the performance statistics from the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetPerformance Lib "acrownt.dll" (MA As Long, MS As Long, FSS As Long, SS As Long, SB As Long, WSC As Long, RSC As Long, BSC As Long, SSC As Long, BWS As Long, BRS As Long, BBS As Long, BSW As Long, BSR As Long, WC As Long, RC As Long, BW As Long, BR As Long, BWC As Long, BRC As Long, BBW As Long, BBR As Long, TWS As Long, TRS As Long, TBS As Long, TSS As Long, TR As Long, TW As Long, TBR As Long, TBW As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetPerformance(unsigned long* pNumMovesArrived, unsigned long* pNumMovesStarted, unsigned long* pNumStatusFastSent, unsigned long* pNumStatusSent, unsigned long* pNumStatusBack, unsigned long* pNumTimesWriteServiceCalled, unsigned long* pNumTimesReadServiceCalled, unsigned long* pNumTimesBinServiceCalled, unsigned long* pNumTimesStatusServiceCalled, unsigned long* pNumBytesTransfWriteService, unsigned long* pNumBytesTransfReadService, unsigned long* pNumBytesTransfBinService, unsigned long* pNumBytesWriteStatusService, unsigned long* pNumBytesReadStatusService, unsigned long* pNumTimesWriteCalled, unsigned long* pNumTimesReadCalled, unsigned long* pNumBytesTransfWrite, unsigned long* pNumBytesTransfRead, unsigned long* pNumTimesBinaryWriteCalled, unsigned long* pNumTimesBinaryReadCalled, unsigned long* pNumBytesTransfBinaryWrite, unsigned long* pNumBytesTransfBinaryRead, __int64* pTimeSpentWriteService, __int64* pTimeSpentReadService, __int64* pTimeSpentBinService, __int64* pTimeSpentStatusService, __int64* pTimeSpentRead, __int64* pTimeSpentWrite, __int64* pTimeSpentBinaryRead, __int64* pTimeSpentBinaryWrite, int card);</pre>
Returns	Error returned by GetLastError.

AcroGetPerformance, continued

Arguments

`pNumMovesArrived`—number of move blocks queued into the driver since last reset.

`pNumMovesStarted`—number of moves started by the card since last reset.

`pNumStatusFastSent`—number of fast statuses send to the card since the last reset.

`pNumStatusSent`—number of status requests send to the card.since last reset.

`pNumStatusBack`—number of status received back from the card since last reset. `pNumTimesWriteServiceCalled`—number of times write service was called since last reset.

`pNumTimesReadServiceCalled`—number of times read service was called since last reset.

`pNumTimesBinServiceCalled`—number of times binary service was called since last reset.

`pNumTimesStatusServiceCalled`—number of times status service was called since last reset.

`pNumBytesTransfWriteService`—number of bytes transferred by write service since last reset.

`pNumBytesTransfReadService`—number of bytes transferred by read service since last reset.

`pNumBytesTransfBinService`—number of bytes transferred by binary service since last reset.

`pNumBytesWriteStatusService`—number of bytes written by status service since last reset.

`pNumBytesReadStatusService`—number of bytes read by status service since last reset.

`pNumTimesWriteCalled`—number of times write was called by the application since last reset.

`pNumTimesReadCalled`—number of times read was called by the application since last reset.

`pNumBytesTransfWrite`—number of bytes transferred by write operation since last reset.

`pNumBytesTransfRead`—number of bytes transferred by read operation since last reset.

`pNumTimesBinaryWriteCalled`—number of times binary write was called since last reset.

`pNumTimesBinaryReadCalled`—number of time binary read was called since last reset.

`pNumBytesTransfBinaryWrite`—number of bytes transferred by binary write since last reset.

`pNumBytesTransfBinaryRead`—number of bytes transferred by binary read since last reset.

`pTimeSpentWriteService`—time spent (in nanoseconds) in write service since the last reset.

`pTimeSpentReadService`—time spent (in nanoseconds) in read service since the last reset.

`pTimeSpentBinService`—time spent in binary service since the last reset.

`pTimeSpentStatusService`—time spent in the status service since the last read.

`pTimeSpentRead`—time spent waiting to read since last reset.

`pTimeSpentWrite`—time spent (in nanoseconds) waiting to write since last reset. `pTimeSpentBinaryRead`—time spent in binary read since last reset.

`pTimeSpentBinaryWrite`—time spent (in nanoseconds) in binary write since last reset.

`card`—card number

AcroGetPerformance, continued

Error	Error returned by GetLastError.
Globals	none.
Remarks	This function is used to monitor critical parameters in the ACR8K.SYS driver.
Visual Basic Example	<pre>Dim MyError As Long MyError = AcroGetPerformance (MA, MS, FSS, SS, SB, WSC, RSC, BSC, SSC, BWS, BRS, BBS, BSS, WC, RC, BW, BR, BWC, BRC, BBW, BBR, TWS, TRS, TBS, TSS, TR, TW, TBR, TBW, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; MyError = AcroGetPerformance(pNumMovesArrived, pNumMovesStarted, pNumStatusFastSent, pNumStatusSent, pNumStatusBack, pNumTimesWriteServiceCalled, pNumTimesReadServiceCalled, pNumTimesBinServiceCalled, pNumTimesStatusServiceCalled, pNumBytesTransfWriteService, pNumBytesTransfReadService, pNumBytesTransfBinService, pNumBytesWriteStatusService, pNumBytesReadStatusService, pNumTimesWriteCalled, pNumTimesReadCalled, pNumBytesTransfWrite, pNumBytesTransfRead, pNumTimesBinaryWriteCalled, pNumTimesBinaryReadCalled, pNumBytesTransfBinaryWrite, pNumBytesTransfBinaryRead, pTimeSpentWriteService, pTimeSpentReadService, pTimeSpentBinService, pTimeSpentStatusService, pTimeSpentRead, pTimeSpentWrite, pTimeSpentBinaryRead, pTimeSpentBinaryWrite, 0); //Example assumes all variables are declared.</pre>
See Also	AcroInitPerformance
Libraries	AcroLibNT

AcroGetReadTimeOut

Description	Returns the timeout used by RTXT service to wait on the hardware port before timing out. Default is 10 microseconds.
Visual Basic	<pre>Declare Function AcroGetReadTimeOut Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetReadTimeOut(unsigned long *pDelta, int card);</pre>
Returns	
Arguments	pDelta—returned timeout in microseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyErrorAs Long pDelta As Long MyError=AcroGetReadTimeOut (pDelta,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; long pDelta[1]; MyError=AcroGetReadTimeOut(pDelta,0);</pre>
See Also	AcroSetReadTimeOut
Libraries	AcroLibNT

AcroGetReadTimerDelta

Description	Gets the current period (in msec) of the Read Timer associated with the RTXT service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetReadTimerDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetReadTimerDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long Dim pDelta As Long 'Read the Read TimerDelta into a long variable pDelta MyError = AcroGetReadTimerDelta (pDelta, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pDelta[1]; //Read the Read TimerDelta into a long variable pDelta MyError= AcroGetReadTimerDelta(pDelta,0);</pre>
See Also	AcroSetReadTimerDelta
Libraries	AcroLibNT

AcroGetStatusTimerDelta

Description	Gets the current period (in msec) of the BinaryTimer associated with the RSTAT service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetStatusTimerDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetStatusTimerDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long Dim pDelta as Long ' Read the Status TimerDelta into a long variable pDelta MyError = AcroGetStatusTimerDelta (pDelta, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; unsigned long pDelta[1]; //Read the Status TimerDelta into a long variable pDelta MyError= AcroGetStatusTimerDelta(pDelta,0);</pre>
See Also	AcroSetStatusTimerDelta
Libraries	AcroLibNT

AcroGetTimeMilli

Description	Returns the current time in milliseconds.
Visual Basic	<i>Function AcroGetTimeMilli Lib "acrownt.dll" () As Long</i>
Visual C++	<i>long AcroGetTimeMilli();</i>
Returns	none.
Arguments	none.
Error	none.
Globals	none.
Remarks	This function is intended to work for times that are no more than an hour apart. It should not be used for any time critical applications.
Visual Basic Example	<pre>Dim CurrentTime as Long CurrentTime = AcroGetTimeMilli</pre>
Visual C++ Example	<pre>long CurrentTime; CurrentTime = AcroGetTimeMilli();</pre>
See Also	
Libraries	AcroLibNT

AcroGetTransactionState

Description	Gets the number of transactions entering and leaving the system since the state was last initialized.
Visual Basic	<pre>Declare Function AcroGetTransactionState Lib "acrownt.dll" (pnumberadded As Long, pnumberdeleted As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetTransactionState(long* numberadded, long* numberdeleted, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	numberadded—number of transactions added to the driver since last reset. numberdeleted—number of transactions deleted from the driver since last reset. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	This function is used in conjunction with AcroSetTransaction State to monitor the rate at which transactions are entering and leaving the system.
Visual Basic Example	<pre>'Get the Transaction state dim numadd as long dim numdel as long dim AcroError as long AcroError= AcroGetTransactionState(numadd, numdel,0)</pre>
Visual C++ Example	<pre>//read the transaction state long numadd; long numdel; AcroGetTransactionState(&numadd, &numdel, 0);</pre>
See Also	AcroSetTransactionState
Libraries	AcroLibNT

AcroGetVersion

Description	Returns version of the current library encoded in a long integer.
Visual Basic	<i>Function AcroGetVersion Lib "acrownt.dll" () As Long</i>
Visual C++	<i>long AcroGetVersion();</i>
Returns	none.
Arguments	none.
Error	none
Globals	none
Remarks	0 d r m m r r r where d/r =debug/release(0=debug or 1=release), m=main version, r=revision.
Visual Basic Example	<pre>Dim LongVar As Long LongVar = AcroGetVersion</pre>
Visual C++ Example	<pre>long LongVar; LongVar = AcroGetVersion();</pre>
See Also	
Libraries	AcroLibNT

AcroGetWriteTimeOut

Description	Returns the timeout used by WRITE service to wait on the hardware port before timing out. Default is 10 microseconds.
Visual Basic	<pre>Declare Function AcroGetWriteTimeOut Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetWriteTimeOut(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—timeout in microseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyErrorAs Long Dim pDelta As Long MyError=AcroGetWriteTimeOut (pDelta,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; long pDelta[1]; MyError=AcroGetWriteTimeOut(pDelta,0);</pre>
See Also	AcroSetWriteTimeOut
Libraries	AcroLibNT

AcroGetWriteTimerDelta

Description	Gets the current period (in msec) of the WriteTimer associated with the WRITE service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroGetWriteTimerDelta Lib "acrownt.dll" (pDelta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetWriteTimerDelta(unsigned long *pDelta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pDelta—returned period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long Dim pDelta As Long ' Read the Write TimerDelta into a long variable pDelta MyError = AcroGetWriteTimerDelta (pDelta, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; //Read the Write TimerDelta into a long variable. long pDelta[1]; MyError= AcroGetWriteTimerDelta(pDelta,0);</pre>
See Also	AcroSetWriteTimerDelta
Libraries	AcroLibNT

AcroHideErrorDialogs

Description	Disables the showing of TimeOut Error Dialogs
Visual Basic	<i>Declare Sub AcroHideErrorDialogs Lib "acrownt.dll" ()</i>
Visual C++	<i>void AcroHideErrorDialogs()</i>
Returns	nothing.
Arguments	none.
Error	none
Globals	none
Remarks	This function should be called to suppress error dialogs for release version of the application
Visual Basic Example	<code>AcroHideErrorDialogs</code>
Visual C++ Example	<code>AcroHideErrorDialogs();</code>
See Also	<code>AcroShowErrorDialogs</code>
Libraries	AcroLibNT

AcroInitialize

Description	Initialize the buffers and the channels for the driver.
Visual Basic	<i>Declare Sub AcroInitialize Lib "acrownt.dll" (ByVal card As Long)</i>
Visual C++	<i>void AcroInitialize(int card);</i>
Returns	nothing.
Arguments	card—number of the card being used.
Error	AcroGetError.
Globals	none.
Remarks	This function must be called once in the user application before any other function from the ACROWNT API is called.
Visual Basic Example	<code>AcroInitialize (0)</code>
Visual C++ Example	<code>AcroInitialize(0);</code>
See Also	AcroCardPresent
Libraries	AcroLibNT

AcroInitPerformance

Description	Set all the performance statistics for ACR8K.SYS to 0.
Visual Basic	<i>Declare Sub AcroInitPerformance Lib "acrownt.dll" (ByVal card As Long)</i>
Visual C++	<i>void AcroInitPerformance(int card);</i>
Returns	nothing.
Arguments	card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	Used in conjunction with AcroGetPerformance.
Visual Basic Example	<code>AcroInitPerformance 0</code>
Visual C++ Example	<code>AcroInitPerformance(0);</code>
See Also	<code>AcroGetPerformance</code>
Libraries	AcroLibNT

AcroReceiveTextString

Description	Reads a string from the TXT READ channel.
Visual Basic	<pre>Declare Function AcroReceiveTextString Lib "acrownt.dll" (ByVal pString As String, ByVal NumberOfBytesToGet As Long, pNumberOfBytesReceived As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>int AcroReceiveTextString(unsigned char* pString, long NumberOfBytesToGet, long* pNumberOfBytesReceived, int card);</pre>
Returns	returns an error from AcroGetError.
Arguments	<p>pString—pointer to a user allocated buffer for reading the text string</p> <p>NumberOfBytesToGet—number of bytes to read from the text buffer.</p> <p>pNumberOfBytesReceived—number of bytes actually read from the text buffer.</p> <p>card—number of the card being used.</p>
Error	AcroGetError returns the error.
Globals	none.
Remarks	This function is typically used to implement a text terminal. The binaryinterface should be used for all other transfers back to the card.
Visual Basic Example	<pre>Dim pString as String Dim NumberOfBytesToGet as Long Dim pNumberOfBytesReceived as Long Dim MyError as Long ' Read 10 bytes from the TXT READ channel NumberOfBytesToGet=10 MyError=AcroReceiveTextString (pString, NumberOfBytesToGet, pNumberOfBytesReceived, 0)</pre>
Visual C++ Example	<pre>char pString[20]; long NumberOfBytesToGet; long pNumberOfBytesReceived[1]; int Error; // read 10 bytes from TXT READ channel NumberOfBytesToGet=10; Error=AcroReceiveTextString(pString, NumberOfBytesToGet, pNumberOfBytesReceived, 0);</pre>
See Also	AcroSendTextString
Libraries	AcroLibNT

AcroSendEscape

Description	Sends an ESCAPE character to the card.
Visual Basic	<i>Declare Function AcroSendEscape Lib "acrownt.dll" (ByVal card As Long) As Long</i>
Visual C++	<i>int AcroSendEscape(int card);</i>
Returns	returns an Acrolib error code(error is derived from AcroGetError which is called internally by AcroSendEscape).
Arguments	card—number of the card being used.
Error	returns an Acrolib error code.
Globals	none.
Remarks	Sending an escape to the card stops any immediate mode moves and clears the cards move buffer. Has no effect on driver buffers.
Visual Basic Example	<pre>Dim MyError As Long MyError=AcroSendEscape(0)</pre>
Visual C++ Example	<pre>int Error; Error=AcroSendEscape(0);</pre>
See Also	
Libraries	AcroLibNT

AcroSendString

Description	Sends text string to the WRITE channel.
Visual Basic	<i>Declare Function AcroSendString Lib "acrownt.dll" (ByVal card As Long) As Long</i>
Visual C++	<i>int AcroSendString(unsigned char* pSstring int card);</i>
Returns	returns an error from AcroGetError.
Arguments	pString-pointer to a string containing the string to write. card-number of the card being used.
Error	AcroGetError returns an error.
Globals	none.
Remarks	This command appends a carriage-return to the string being sent to the card.
Visual Basic Example	<pre>Dim MyError as Long MyError=AcroSendString("PROG0",0)</pre>
Visual C++ Example	<pre>AcroSendString("PROG0",0);</pre>
Libraries	AcroLibNT

AcroSendTextString

Description	Send a text string to the WRITE channel.
Visual Basic	<pre>Declare Function AcroSendTextString Lib "acrownt.dll" (ByVal pString As String, ByVal NumberOfBytesToSend As Long, pNumberOfBytesSent As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>int AcroSendTextString(unsigned char* pString, long NumberOfBytesToSend, long* pNumberOfBytesSent,int card);</pre>
Returns	returns an error from AcroGetError.
Arguments	<p>pString—pointer to a string containing the string to write to the write channel</p> <p>NumberOfBytesToSend—number of bytes to send to the write channel.</p> <p>pNumberOfBytesSent—number of bytes actually sent to the write channel.</p> <p>card—number of the card being used</p>
Error	AcroGetError returns the error.
Globals	none.
Remarks	This function is not thread safe. In other words, it will send the string character by character to the card. This function is useful for implementing a terminal program or sending a long file to the card.
Visual Basic Example	<pre>Dim pString as String Dim NumberOfBytesToSend as Long Dim pNumberOfBytesSent as Long Dim MyError as Long 'Write 6 bytes to the WRITE channel NumberOfBytesToSend=6 pString="PROG0"+ Chr\$(13) MyError=AcroSendTextString (pString, NumberOfBytesToSend, pNumberOfBytesSent,0)</pre>
Visual C++ Example	<pre>char* pString="PROG0\r"; long NumberOfBytesToSend; long pNumberOfBytesSent[1]; NumberOfBytesToSend=6; AcroSendTextString(pString, long NumberOfBytesToSend, pNumberOfBytesSent, 0);</pre>
See Also	AcroSendString
Libraries	AcroLibNT

AcroSetBinaryTimeout

Description	Sets the value of timeout used by RBIN service to wait at the hardware port in the ACR8K.SYS driver. The default is 10 microseconds.
Visual Basic	<pre>Declare Function AcroSetBinaryTimeout Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetBinaryTimeout(unsigned long Delta,int card);</pre>
Returns	returns error from GetLastError.
Arguments	Delta—period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'set the timeout for RBIN service to 20 microseconds AcroSetBinaryTimeout (20,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the timeout for RBIN service to 20 microseconds. MyError=AcroSetBinaryTimeout(20,0);</pre>
See Also	AcroGetBinaryTimeout
Libraries	AcroLibNT

AcroSetBinaryTimerDelta

Description	Sets the current period (in msec) of the BinaryTimer associated with the RBIN service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroSetBinaryTimerDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetBinaryTimerDelta(unsigned long Delta, int card);</pre>
Returns	returns error from GetLastError.
Arguments	Delta—period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'Set the Binary TimerDelta (period) to 20 msec. MyError = AcroSetBinaryTimerDelta (20, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the BinaryTimerDelta (period) to 20 msec. MyError= AcroSetBinaryTimerDelta(20,0);</pre>
See Also	AcroGetBinaryTimerDelta
Libraries	AcroLibNT

AcroSetBinTransThreshold

Description	Sets the threshold to control the fast status fetch. Default is 0.
Visual Basic	<i>Declare Function AcroSetBinTransThreshold Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</i>
Visual C++	<i>AcroSetBinTransThreshold(unsigned long Delta,int card);</i>
Returns	returns error from GetLastError.
Arguments	Delta—binary threshold. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	This is used in conjunction with AcroSetBinTransThreshold to control the fast status requests.
Visual Basic Example	<pre>'set threshold to 0 Dim AcroError as long AcroError = AcroSetBinTransThreshold(0, 0)</pre>
Visual C++ Example	<pre>//set the threshold to 1. AcroSetBinTransThreshold(1,0);</pre>
See Also	
Libraries	AcroLibNT

AcroSetError

Description	Resets an error for the ACROWNT.
Visual Basic	<pre>Declare Sub AcroSetError Lib "acrownt.dll" (ByVal ErrorCode As Long)</pre>
Visual C++	<pre>void AcroSetError(long errorcode);</pre>
Returns	nothing.
Arguments	errorcode—error code. See Errors for specific error codes.
Error	none
Globals	none
Remarks	This function is used to set the internal error variable to a specified value. The function is most often used to reset the error code to 0 (ACRO_SUCCESS) after calling AcroGetError.
Visual Basic Example	<pre>Dim MyError As Long MyError=AcroGetError if MyError<>0 then AcroSetError 0</pre>
Visual C++ Example	<pre>long MyError; MyError=AcroGetError(); if (MyError!=0) AcroSetError(0);</pre>
See Also	AcroGetError
Libraries	AcroLibNT

AcroSetFastStatusTimerDelta

Description	Sets the current period (in msec) of the FastStatusTimer in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroSetFastStatusTimerDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetFastStatusTimerDelta(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'Set the FastStatus TimerDelta (period) to 20 msec. MyError = AcroSetFastStatusTimerDelta (20, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the FastStatusTimerDelta (period) to 20 msec. MyError= AcroSetFastStatusTimerDelta(20,0);</pre>
See Also	AcroGetFastStatusTimerDelta
Libraries	AcroLibNT

AcroSetInitialWait

Description	Sets the wait time (in milliseconds) used by AcroInitialize routine to wait between flushing the queues.
Visual Basic	<i>AcroSetInitialWait(ByVal WaitInMilliseconds As Long)</i>
Visual C++	<i>void AcroSetInitialWait(long WaitInMilliSecond);</i>
Returns	nothing.
Arguments	WaitInMilliseconds—wait in milliseconds.
Error	none
Globals	none
Remarks	Reducing this wait will make AcroInitialize execute faster.
Visual Basic Example	'Set the initial wait to 10 milliseconds AcroSetInitialWait(10)
Visual C++ Example	'Set the initial wait to 20 milliseconds AcroSetInitialWait(20);
See Also	
Libraries	AcroLibNT

AcroSetMaxBinTransPerFSIntr

Description	Sets the maximum binary transactions expected during a fast status periodinterval.
Visual Basic	<pre>Declare Function AcroSetMaxBinTransPerFSIntr Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetMaxBinTransPerFSIntr(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum binary transactions expected per period of fast status fetch. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	This is used in conjunction with AcroSetMaxFastStatusDelta to control the fast status requests.
Visual Basic Example	<pre>'set the max bin. Trans to be 20 transactions/period of the fast status. Dim AcroError as Long AcroError = AcroSetMaxBinTransPerFSIntr(20, 0)</pre>
Visual C++ Example	<pre>//set the max bin. Trans/period of the fast status timer to be 30. AcroSetMaxBinTransPerFSIntr(30,0);</pre>
See Also	
Libraries	AcroLibNT

AcroSetMaxFastStatusDelta

Description	Sets the maximum period (in msec) for the fast status fetch. The default is 100 msec.
Visual Basic	<pre>Declare Function AcroSetMaxFastStatusDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroGetMaxFastStatusDelta(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	none.
Visual Basic Example	<pre>'Set the max fast status delta to 100 msec. Dim AcroError as long AcroError = AcroSetMaxFastStatusDelta(100, 0)</pre>
Visual C++ Example	<pre>//Set the max fast status delta to 100 msec. AcroSetMaxFastStatusDelta(100, 0);</pre>
See Also	
Libraries	AcroLibNT

AcroSetMaxSizeofInputBinaryBuffer

Description	Sets the maximum size of the input Binary buffer (in bytes). The default size is 100 bytes.
Visual Basic	<pre>Declare Sub AcroSetMaxSizeofInputBinaryBuffer Lib "acrownt.dll" (ByVal SizeInCharacters As Long)</pre>
Visual C++	<pre>void AcroSetMaxSizeofInputBinaryBuffer(long SizeInCharacters);</pre>
Returns	nothing.
Arguments	SizeInCharacters—maximum number of bytes allowed in the binary input buffer.
Error	none
Globals	none
Remarks	The buffer is dynamically allocated until it reaches the MaxSize limit.
Visual Basic Example	<pre>'set the input Binary buffer size to be 4096 bytes AcroSetMaxSizeofInputBinaryBuffer(4096)</pre>
Visual C++ Example	<pre>//set the input Binary buffer size to be 4096 bytes AcroSetMaxSizeofInputBinaryBuffer(4096);</pre>
See Also	
Libraries	AcroLibNT

AcroSetMoveCounter

Description	Set the move counter to a specific value and increment the movecounter by "increment" every time an interrupt is detected from the ACR8000 card.
Visual Basic	<pre>Declare Function AcroSetMoveCounter Lib "acrownt.dll" (ByVal counter As Long, ByVal increment As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetMoveCounter(long pcounter, long increment, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	pcounter—move counter increment—increment to the move counter card—number of the card being used.
Error	Error returned by GetLastError.
Globals	
Remarks	none.
Visual Basic Example	<pre>Dim MyError as Long MyError = AcroSetMoveCounter(0,1,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; MyError= AcroSetMoveCounter(0,1,0);</pre>
See Also	AcroGetMoveCounter
Libraries	AcroLibNT

AcroSetReadTimeout

Description	Sets the value of timeout used by RTXTservice to wait at the hardware port in the ACR8K.SYS driver. The default is 10 microseconds.
Visual Basic	<pre>Declare Function AcroSetReadTimeOut Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetReadTimeOut(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—timeout in microseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'set the timeout for RTXT service to 20 microseconds MyError = AcroSetReadTimeOut (20,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the timeout for RTXT service to 20 microseconds. MyError=AcroSetReadTimeOut(20,0);</pre>
See Also	AcroGetReadTimeOut
Libraries	AcroLibNT

AcroSetReadTimerDelta

Description	Sets the current period (in msec) of the ReadTimer associated with the RTXT service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroSetReadTimerDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetReadTimerDelta(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'Set the Read TimerDelta (period) to 20 msec. MyError = AcroSetReadTimerDelta (20, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the ReadTimerDelta (period) to 20 msec. MyError= AcroSetReadTimerDelta(20,0);</pre>
See Also	AcroGetReadTimerDelta
Libraries	AcroLibNT

AcroSetRetryBeforeBinReadTimeOut

Description	Set number of times to retry before timing out on BINR channel. Default is 10000.
Visual Basic	<i>AcroSetRetryBeforeBinReadTimeOut(ByVal HowManyTimes As Long)</i>
Visual C++	<i>void AcroSetRetryBeforeBinReadTimeOut(long HowManyTimes);</i>
Returns	Error returned by GetLastError.
Arguments	HowManyTimes—number of times binary read is tried before timing out.
Error	none
Globals	none
Remarks	This timeout is used by AcroReceiveBinRetry to timeout. Therefore, this parameter will affect all binary reads from the card.
Visual Basic Example	<pre>AcroSetRetryBeforeBinReadTimeOut(20000) 'increases the binary read retry to 20000 times.</pre>
Visual C++ Example	<pre>AcroSetRetryBeforeBinReadTimeOut(5000) //decreases the binary read timeout to 5000 times.</pre>
See Also	
Libraries	AcroLibNT

AcroSetRetryBeforeBinWriteTimeOut

Description	Set number of times to retry before timing out on BINW channel. Default is 10000.
Visual Basic	<pre>Declare Sub AcroSetRetryBeforeBinWriteTimeOut Lib "acrownt.dll" (ByVal HowManyTimes As Long)</pre>
Visual C++	<pre>void AcroSetRetryBeforeBinWriteTimeOut(long HowManyTimes)</pre>
Returns	nothing.
Arguments	HowManyTimes—number of times binary write is tried before timing out.
Error	none
Globals	none
Remarks	This timeout is used by AcroSendBinRetry to timeout. Therefore, this parameter will affect all binary writes to the card.
Visual Basic Example	<pre>AcroSetRetryBeforeBinWriteTimeOut(20000) 'increases the binary write retry to 20000 times.</pre>
Visual C++ Example	<pre>AcroSetRetryBeforeBinWriteTimeOut(5000); //decreases the binary write timeout to 5000 times.</pre>
See Also	
Libraries	AcroLibNT

AcroSetRetryBeforeReadTimeOut

Description	Set number of times to retry before timing out on TXTR channel. Default is 10000.
Visual Basic	<pre>Declare Sub AcroSetRetryBeforeReadTimeOut Lib "acrownt.dll" (ByVal HowManyTimes As Long)</pre>
Visual C++	<pre>void AcroSetRetryBeforeReadTimeOut(long HowManyTimes);</pre>
Returns	nothing.
Arguments	HowManyTimes—number of times read is tried before timing out.
Error	none
Globals	none
Remarks	This timeout is used by AcroReceiveTextRetry to timeout. Therefore, this parameter will affect all text reads from the card.
Visual Basic Example	<pre>AcroSetRetryBeforeReadTimeOut(20000) 'increases the readretry to 20000 times.</pre>
Visual C++ Example	<pre>AcroSetRetryBeforeReadTimeOut(5000); //decreases the readtimeout to 5000 times.</pre>
See Also	
Libraries	AcroLibNT

AcroSetRetryBeforeWriteTimeOut

Description	Set number of times to retry before timing out on TXTW channel. Default is 10000.
Visual Basic	<pre>Declare Sub AcroSetRetryBeforeWriteTimeOut Lib "acrownt.dll" (ByVal HowManyTimes As Long)</pre>
Visual C++	<pre>void AcroSetRetryBeforeWriteTimeOut(long HowManyTimes);</pre>
Returns	nothing.
Arguments	HowManyTimes—number of times write is tried before timing out.
Error	none
Globals	none
Remarks	This timeout is used by AcroSendTextRetry to timeout. Therefore, this parameter will affect all text writes to the card.
Visual Basic Example	<pre>AcroSetRetryBeforeWriteTimeOut(20000) 'increases the write retry to 20000 times.</pre>
Visual C++ Example	<pre>AcroSetRetryBeforeWriteTimeOut(5000); //decreases the write timeout to 5000 times.</pre>
See Also	
Libraries	AcroLibNT

AcroSetStatusTimerDelta

Description	Sets the current period (in msec) of the StatusTimer associated with the RSTAT service in the ACR8K.SYS driver.
Visual Basic	<i>Declare Function AcroSetStatusTimerDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</i>
Visual C++	<i>unsigned long AcroSetStatusTimerDelta(unsigned long Delta, int card);</i>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'Set the Status TimerDelta (period) to 20 msec. MyError = AcroSetStatusTimerDelta (20, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the StatusTimerDelta (period) to 20 msec. MyError= AcroSetStatusTimerDelta(20,0);</pre>
See Also	AcroGetStatusTimerDelta
Libraries	AcroLibNT

AcroSetTimeOutForBinReadMutex

Description	Timeout for resource mutex for the BINR channel. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForBinReadMutex Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetTimeOutForBinReadMutex(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none
Globals	none
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most 10 milliseconds before timing out the on BINR channel. AcroSetTimeOutForBinReadMutex 10</pre>
Visual C++ Example	<pre>// Make a thread wait forever before timing out the on BINR channel. AcroSetTimeOutForBinReadMutex(-1);</pre>
See Also	AcroSetTimeOutForBinWriteMutex
Libraries	AcroLibNT

AcroSetTimeOutForBinTransMutex

Description	Time out for resource mutex for BINTR channel. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForBinTransMutex Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>AcroSetTimeOutForBinTransMutex(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none.
Globals	none.
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most 10 milliseconds before timing out the on BINTR channel. AcroSetTimeOutForBinTransMutex 10</pre>
Visual C++ Example	<pre>// Make a thread wait forever before timing out the on BINTR channel. AcroSetTimeOutForBinTransMutex(-1);</pre>
See Also	
Libraries	AcroLibNT

AcroSetTimeOutForBinTransWait

Description	Timeout between when a binary transaction is issued and when an answer is expected back. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForBinTransWait Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetTimeOutForBinTransWait(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none.
Globals	none.
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most 10 milliseconds before timing out the on reading the BINTR channel. AcroSetTimeOutForBinTransWait 10</pre>
Visual C++ Example	<pre>// Make a thread wait forever before timing out on reading the BINTR channel. AcroSetTimeOutForBinTransWait(-1);</pre>
See Also	
Libraries	AcroLibNT

AcroSetTimeOutForBinWriteMutex

Description	Timeout for resource mutex for the BINW channel. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForBinWriteMutex Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetTimeOutForBinWriteMutex(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none
Globals	none
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most for 10 milliseconds before timing out on BINW channel. AcroSetTimeOutForBinWriteMutex 10</pre>
Visual C++ Example	<pre>//Make a thread wait for ever before timing out on reading the BINW channel. AcroSetTimeOutForBinWriteMutex(-1);.</pre>
See Also	AcroSetTimeOutForBinReadMutex
Libraries	AcroLibNT

AcroSetTimeOutForReadMutex

Description	Timeout for resource mutex for the TXTR channel. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForReadMutex Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetTimeOutForReadMutex(long WaitInMilliseconds)</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none
Globals	none
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most for 10 milliseconds before timing out on TXTR channel. AcroSetTimeOutForReadMutex 10.</pre>
Visual C++ Example	<pre>//Make a thread wait forever before timing out on the TXTR channel. AcroSetTimeOutForReadMutex(-1);</pre>
See Also	AcroSetTimeOutForWriteMutex
Libraries	AcroLibNT

AcroSetTimeOutForWriteMutex

Description	Timeout for resource mutex for the TXTW channel. The default is 10000 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetTimeOutForWriteMutex Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetTimeOutForWriteMutex(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to acquire the read mutex. A negative value indicates a wait of INFINITY.
Error	none
Globals	none
Remarks	A negative value indicates a wait of INFINITE.
Visual Basic Example	<pre>'Make a thread wait at most for 10 milliseconds before timing out on TXTW channel. AcroSetTimeOutForWriteMutex(10)</pre>
Visual C++ Example	<pre>//Make a thread wait forever before timing out on the TXTW channel. AcroSetTimeOutForWriteMutex(-1);</pre>
See Also	AcroSetTimeOutForReadMutex
Libraries	AcroLibNT

AcroSetTransactionState

Description	Sets the state of the transactions in the system for observation purposes.
Visual Basic	<i>Declare Function AcroSetTransactionState Lib "acrownt.dll" (ByVal numberadded As Long, ByVal numberdeleted As Long, ByVal card As Long) As Long</i>
Visual C++	<i>unsigned long AcroSetTransactionState(long numberadded, long numberdeleted,int card);</i>
Returns	Error returned by GetLastError.
Arguments	numberadded—number of transactions added to the driver since last reset. numberdeleted—number of transactions deleted from the driver since last reset. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	none.
Visual Basic Example	<pre>'Initialize the Transaction State Dim AcroError as long AcroError = AcroSetTransactionState(0,0,0)</pre>
Visual C++ Example	<pre>//Initialize the transaction state. AcroSetTransactionState(0,0,0);</pre>
See Also	AcroGetTransactionState
Libraries	AcroLibNT

AcroSetWaitBeforeTimeOut

Description	Sets the wait time (in milliseconds) used AcroWaitputCh and AcroWaitGetCh to timeout from the FIFO.
Visual Basic	<pre>Declare Sub AcroSetWaitBeforeTimeOut Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetWaitBeforeTimeOut(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time before timing out from a read or write to a port.
Error	none
Globals	none
Remarks	Reducing this wait will make AcroInitialize execute faster.
Visual Basic Example	<pre>'Set timeout to 10 msec AcroSetWaitBeforeTimeOut(10)</pre>
Visual C++ Example	<pre>'Set timeout to be 20 msec AcroSetWaitBeforeTimeOut(20);</pre>
See Also	
Libraries	AcroLibNT

AcroSetWaitBetweenStatusFetch

Description	Sets the wait time between sending and receiving a non-fast status request. Default is 2 milliseconds.
Visual Basic	<pre>Declare Sub AcroSetWaitBetweenStatusFetch Lib "acrownt.dll" (ByVal WaitInMilliseconds As Long)</pre>
Visual C++	<pre>void AcroSetWaitBetweenStatusFetch(long WaitInMilliseconds);</pre>
Returns	nothing.
Arguments	WaitInMilliseconds—maximum time to wait for a status coming back.
Error	none
Globals	none
Remarks	This wait only applies to slow status fetch commands.
Visual Basic Example	<pre>'sets the wait to about 1 millisecond AcroSetWaitBetweenStatusFetch(1)</pre>
Visual C++ Example	<pre>//sets the wait to be 0. AcroSetWaitBetweenStatusFetch(0);</pre>
See Also	
Libraries	AcroLibNT

AcroSetWriteTimeout

Description	Sets the value of timeout used by WRITE service to wait at the hardware port in the ACR8K.SYS driver. The default is 10 microseconds.
Visual Basic	<pre>Declare Function AcroSetWriteTimeOut Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetWriteTimeOut(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long 'set the timeout for WRITE service to 20 microseconds AcroSetWriteTimeOut (20,0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the timeout for WRITE service to 20 microseconds. MyError=AcroSetWriteTimeOut(20,0);</pre>
See Also	AcroGetWriteTimeOut
Libraries	AcroLibNT

AcroSetWriteTimerDelta

Description	Sets the current period (in msec) of the WriteTimer associated with the WRITE service in the ACR8K.SYS driver.
Visual Basic	<pre>Declare Function AcroSetWriteTimerDelta Lib "acrownt.dll" (ByVal Delta As Long, ByVal card As Long) As Long</pre>
Visual C++	<pre>unsigned long AcroSetWriteTimerDelta(unsigned long Delta, int card);</pre>
Returns	Error returned by GetLastError.
Arguments	Delta—maximum period in milliseconds. card—number of the card being used.
Error	Error returned by GetLastError.
Globals	none.
Remarks	
Visual Basic Example	<pre>Dim MyError as Long ' Set the Write TimerDelta (period) to 20 msec. MyError = AcroSetWriteTimerDelta (20, 0)</pre>
Visual C++ Example	<pre>unsigned long MyError; // set the WriteTimerDelta (period) to 20 msec. MyError= AcroSetWriteTimerDelta(20,0);</pre>
See Also	AcroGetWriteTimerDelta
Libraries	AcroLibNT

AcroShowErrorDialogs

Description	Enables showing of Error Dialogs for Debugging
Visual Basic	<i>Declare Sub AcroShowErrorDialog Lib "acrownt.dll"()</i>
Visual C++	<i>void AcroShowError Dialog();</i>
Returns	nothing.
Arguments	none.
Error	none
Globals	none
Remarks	This function should only be used for debugging the application
Visual Basic Example	<code>AcroShowErrorDialog</code>
Visual C++ Example	<code>AcroShowErrorDialog();</code>
See Also	<code>AcroHideErrorDialog</code>
Libraries	AcroLibNT

AcroTextBufferEmpty

Description	Returns a 1 if there are no elements in the Text Buffer. Otherwise, returns a 0.
Visual Basic	<pre>Declare Function AcroTextBufferEmpty Lib "acrownt.dll" (ByVal card As long) As Long</pre>
Visual C++	<pre>int AcroTextBufferEmpty(int card);</pre>
Returns	Returns a 1 if there are no elements in the Text Buffer. Otherwise, returns a 0.
Arguments	card—number of the card being used
Error	none.
Globals	none.
Remarks	This command is often useful when implementing a text terminal.
Visual Basic Example	<pre>Dim Flag as Long 'check if text buffer is empty for card 0. Flag = AcroTextBufferEmpty(0)</pre>
Visual C++ Example	<pre>int Flag ; //check if text buffer is empty for card 0. Flag = AcroTextBufferEmpty(0);</pre>
See Also	
Libraries	AcroLibNT

AcroWait

Description	Wait in the program.
Visual Basic	<pre>Declare Sub AcroWait Lib "acrownt.dll" (ByVal milliseconds As Long)</pre>
Visual C++	<pre>void AcroWait(unsigned long milliseconds);</pre>
Returns	nothing.
Arguments	milliseconds—wait in milliseconds
Error	AcroGetError returns the error.
Globals	none.
Remarks	This function should only be used to wait for times that are less than one minute.
Visual Basic Example	<pre>' Wait for 2 milliseconds AcroWait 2</pre>
Visual C++ Example	<pre>// wait for 2 milliseconds AcroWait(2);</pre>
See Also	
Libraries	AcroLibNT

This page intentionally left blank.

Chapter 4 Errors

This chapter explains the types of errors generated in the NT version of the AcroLib API.

Windows NT

Error codes and their mapping is shown in the table below.

Error Code	Error
0	ACRO_SUCCESS
-1	ACRO_WRITE_TIMEOUT_ERROR
-2	ACRO_READ_TIMEOUT_ERROR
-3	ACRO_READ_MUTEX_TIMEOUT_ERROR
-4	ACRO_WRITE_MUTEX_TIMEOUT_ERROR
-10	ACRO_CARD_NOT_FOUND
-11	ACRO_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR
-13	ACRO_BIN_READ_MUTEX_TIMEOUT_ERROR
-14	ACRO_BIN_WRITE_MUTEX_TIMEOUT_ERROR
-15	ACRO_BIN_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR
-16	ACRO_BIN_WRITE_TIMEOUT_ERROR
-17	ACRO_BIN_READ_TIMEOUT_ERROR
-18	ACRO_BIN_TRANS_TIMEOUT_ERROR
-19	ACRO_BIN_TRANS_MUTEX_TIMEOUT_ERROR
-20	ACRO_BIN_TRANS_WAIT_TIMEOUT_ERROR
-22	ACRO_BIN_TRANS_RANGE_VIOLATION_ERROR

ACRO_SUCCESS (0)

No error has occurred.

ACRO_WRITE_TIMEOUT_ERROR (-1)

Meaning

The application was not able to write to the text channel in the driver.

Cause

The DLL was not able to write to the text channel in the number of retries specified by the `AcroSetRetryBeforeWriteTimeOut` function (defaults to 10000).

Remedies

1. Reduce traffic to the write channel.
2. Increase the retries for write channel by using `AcroSetRetryBeforeWriteTimeOut`.

ACRO_READ_TIMEOUT_ERROR (-2)

Meaning

The application was not able to read from the text channel in the driver.

Cause

The DLL was not able to read from the read channel in the number of retries specified by the AcroSetRetryBeforeReadTimeOut function (defaults to 10000).

Remedies

1. Reduce traffic to the read channel.
2. Increase the retries for write channel by using AcroSetRetryBeforeReadTimeOut.

ACRO_READ_MUTEX_TIMEOUT_ERROR (-3)

Meaning

The application was not able to acquire the text channel for writing.

Cause

The DLL was not able to acquire the mutex associates with the write channel in the time specified by the AcroSetTimeOutForWriteMutex function.

Remedies

1. Reduce traffic to the write channel.
2. Increase the timeout for WriteMutex by using AcroSetTimeOutForWriteMutex.
3. Set the timeout for WriteMutex to INFINITY by using a negative timeout value.

ACRO_WRITE_MUTEX_TIMEOUT_ERROR (-4)

Meaning

The application was not able to acquire the text channel for reading.

Cause

The DLL was not able to acquire the mutex associates with the read channel in the time specified by the AcroSetTimeOutForReadMutex function.

Remedies

1. Reduce traffic to the read channel.
2. Increase the timeout for ReadMutex by using AcroSetTimeOutForReadMutex.
3. Set the timeout for ReadMutex to INFINITY by using a negative timeout value.

ACRO_CARD_NOT_FOUND (-10)

Meaning

The application was not able to connect to a card.

Cause

1. The card is not present in the computer.
2. The address for the card specified in the registry is not the same as used on hardware jumpers on the card.

Remedies

1. Check to see that a card is present in the computer.
2. Make sure the the card address specified in the registry is also selected on the hardware jumpers on the card.

ACRO_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-11)

Meaning

The application was not able to acquire the text channel for reading and writing.

Cause

The DLL was not able to acquire the mutexes associated with the read and write channels in the time specified by the `AcroSetTimeOutForReadMutex` and `AcroSetTimeOutforWriteMutex` function.

Remedies

1. Reduce outgoing and incoming traffic to the read and write channel.
2. Increase the timeout for read mutex by using `AcroSetTimeOutForReadMutex` or increase timeout for write write mutex by using `AcroSetTimeOutForWriteMutex`.
3. Set the timeout for read mutex to INFINITY by using a negative timeout value or set the timeout for write mutex to INFINITY by using a negative timeout value.

ACRO_BIN_READ_MUTEX_TIMEOUT_ERROR (-13)

Meaning

The application was not able to acquire the binary read channel for reading.

Cause

The DLL was not able to acquire the mutex associates with the binary read channel in the time specified by the AcroSetTimeOutForBinReadMutex function (default is 10000 msec).

Remedies

1. Reduce traffic to the binary read channel.
2. Increase the timeout for binary read mutex by using AcroSetTimeOutForBinReadMutex.
3. Set the timeout for binary read mutex to INFINITY by using a negative timeout value.

ACRO_BIN_WRITE_MUTEX_TIMEOUT_ERROR (-14)

Meaning

The application was not able to acquire the binary write channel for writing.

Cause

The DLL was not able to acquire the mutex associates with the binary write channel in the time specified by the AcroSetTimeOutForBinWriteMutex function (default is 10000 msec).

Remedies

1. Reduce traffic to the binary write channel.
2. Increase the timeout for binary write mutex by using AcroSetTimeOutForBinWriteMutex.
3. Set the timeout for binary write mutex to INFINITY by using a negative timeout value.

ACRO_BIN_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-15)

Meaning

The application was not able to acquire the binary read and write channels for reading and writing.

Cause

The DLL was not able to acquire the mutexes associates with the binary read and write channels in the time specified by the `AcroSetTimeOutForBinReadMutex` and `AcroSetTimeOutForBinWriteMutex` functions (default is 10000 msec).

Remedies

1. Reduce traffic to the binary read and write channels.
2. Increase the timeout for binary read mutex by using `AcroSetTimeOutForBinReadMutex` or increase the timeout for binary write mutex by using `AcroSetTimeOutForBinWriteMutex`.
3. Set the timeout for binary read mutex to INFINITY by using a negative timeout value or set the timeout for binary read mutex to INFINITY by using a negative timeout value.

ACRO_BIN_WRITE_TIMEOUT_ERROR (-16)

Meaning

The application was not able to write to the binary write channel in the driver.

Cause

The DLL was not able to write to the binary write channel in the number of retries specified by the `AcroSetRetryBeforeBinWriteTimeOut` function (defaults to 10000).

Remedies

1. Reduce traffic to the binary write channel.
2. Increase the retries for binary write channel by using `AcroSetRetryBeforeBinWriteTimeOut`.

ACRO_BIN_READ_TIMEOUT_ERROR (-17)

Meaning

The application was not able to read from the binary read channel in the driver.

Cause

The DLL was not able to read the binary read channel in the number of retries specified by the `AcroSetRetryBeforeBinReadTimeOut` function (defaults to 10000).

Remedies

1. Reduce traffic to the binary read channel.
2. Increase the retries for binary read channel by using `AcroSetRetryBeforeBinReadTimeOut`.

ACRO_BIN_TRANS_TIMEOUT_ERROR (-18)

Meaning

The application was not able to write the binary transaction to the binary write channel in the driver.

Cause

The DLL was not able to write the binary transaction to the binary write channel because the write buffer was full or it could not allocate enough memory for it.

Remedies

1. Reduce traffic to the binary write channel.
2. Retry the transactions on such errors.

ACRO_BIN_TRANS_MUTEX_TIMEOUT_ERROR (-19)

Meaning

The application was not able to acquire the binary write channel for writing a binary transaction.

Cause

The DLL was not able to acquire the mutex associates with the binary write channel in the time specified by the AcroSetTimeOutForBinTransMutex function (default is 10000 msec).

Remedies

1. Reduce traffic to the binary write channel.
2. Increase the timeout for binary transaction mutex by using AcroSetTimeOutForBinTransMutex.
3. Set the timeout for binary transaction mutex to INFINITY by using a negative timeout value.

ACRO_BIN_TRANS_WAIT_TIMEOUT_ERROR (-20)

Meaning

The application was not able to receive an answer back to a binary transaction request in the time specified.

Cause

The DLL was not able to receive an answer back in the time specified by AcroSetTimeOutForBinTransWait function (default is 10000 msec).

Remedies

1. Reduce traffic to the binary read channel.
2. Increase the wait timeout for binary transaction by using AcroSetTimeOutForBinTransWait.
3. Set the timeout for binary transaction wait to INFINITY by using a negative timeout value.

ACRO_BIN_TRANS_RANGE_VIOLATION_ERROR (-22)

Meaning

The application was not able to submit a binary transaction because the size of the transaction was zero or negative

Cause

The binary request has illegal values.

Remedies

Check the function call to make sure all the parameter are valid.

Appendix A Parameter Reference

Parameter Overview

See Acroloop Motion Control User's Guide, Part Number PM08120, Appendix A.

This page intentionally left blank.

Appendix B Flag Reference

Flag Overview

See Acroloop Motion Control User's Guide, Part Number PM08120, Appendix B.

This page intentionally left blank.

Appendix C Binary Host Interface

Binary Data Transfer

See Acroloop Motion Control User's Guide, Part Number PM08120, Chapter 6.

This page intentionally left blank.

Appendix D Architecture

This Appendix explains the basic architecture underlying the NT version of the AcroLib API.

Application Level

As figure below shows, when an application calls a function using the binary interface in the AcroLib API, the acrownt.dll converts it into a call to AcroBinaryWriteRetry, AcroBinaryReadRetry, AcroBinaryTransactionSubmit. Similarly, acrownt.dll converts calls to functions in API that deal with text (e.g., AcroSendTextString) to AcroReadRetry and AcroWriteRetry.

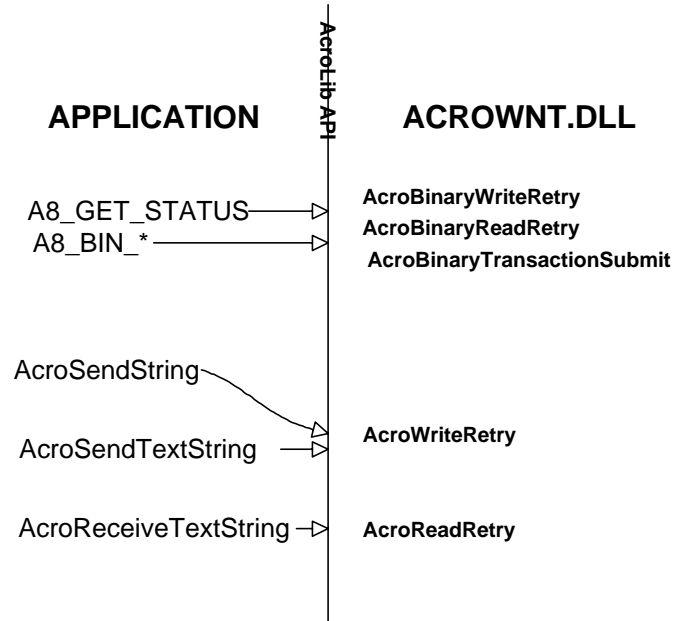


Figure 3 Application and ACROWNT.DLL

DLL Level

As figure below shows, acrownt.dll uses lower level function such as AcroBinaryWriteRetry to write to various channels in the ACR8K.SYS driver. For example, AcroBinaryWriteRetry can be used to write to the BINW channel (Binary Write).

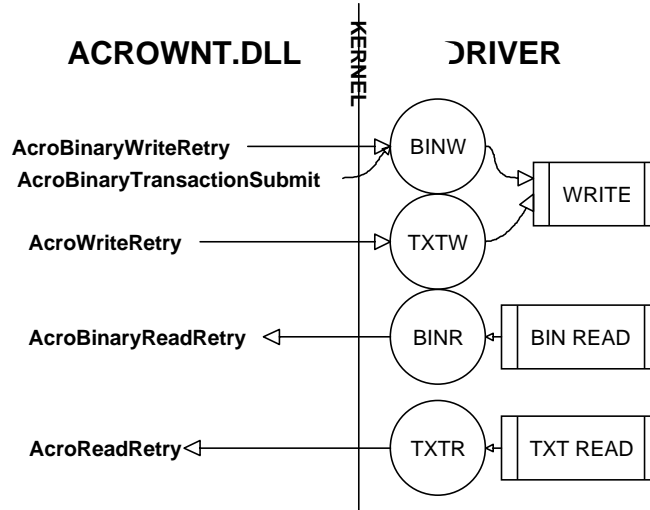


Figure 4. ACROWNT.DLL and ACR8K.SYS DRIVER

Since the ACROWNT.DLL supports multithreaded applications, it uses mutexes to coordinate access to the four channels.

Driver Level

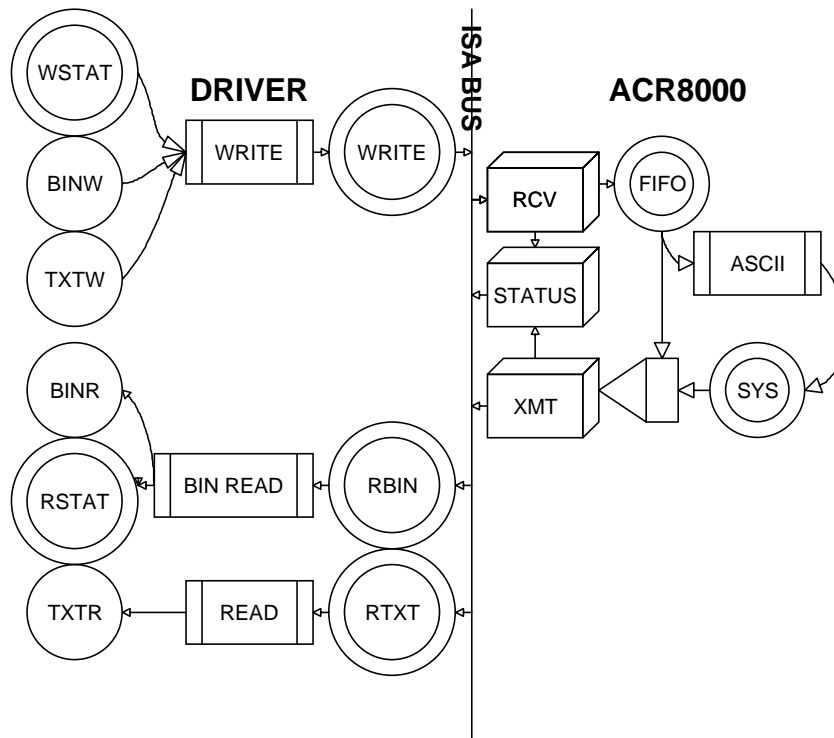


Figure 5 ACR8K.SYS DRIVER and ACR8000 Card

As figure above shows, the ACR8K.SYS driver supports two output channels BINW (Binary Write), TXTW (Text Write) and two input channels BINR (Binary Read) and TXTR (Text Read). BINW and TXTW use an intermediate software WRITE buffer to transmit a packet to the RCV hardware FIFO on the ACR8000 card. The RCV FIFO service either queues a packet to the ASCII queue (e.g., a MOVE packet) or immediately return the result back (e.g., a STATUS request packet) through the XMT hardware FIFO. BINR (Binary Read) and TXTR (Text Read) use BINREAD and TXTR buffers to read data coming back from the XMT hardware FIFO on the ACR8000 side.

The driver uses five tasks. The WRITE task writes the WRITE buffer to the card. RBIN and RTXT tasks are used to read data from the card. RSTAT and WSTAT tasks are used to update fast status access from and to the card.

Index

A

A16_BIN_FOV, 13
A16_BIN_MOVE_IEEE, 25
A16_BIN_MOVE_LONG, 29
A16_BIN_ROV, 36
A16_IEEE_MOVE, 45
A16_LONG_MOVE, 49
A8_BIN_ADDRESS, 10
A8_BIN_CLR, 11
A8_BIN_FOV, 12
A8_BIN_GET_STATUS, 14
A8_BIN_GETIEEE, 15
A8_BIN_GETLONG, 16
A8_BIN_GROUP_GETIEEE, 17
A8_BIN_GROUP_GETLONG, 18
A8_BIN_INIT_STATUS, 19
A8_BIN_MASK, 22
A8_BIN_MOVE_IEEE, 23
A8_BIN_MOVE_LONG, 27
A8_BIN_PEEK_IEEE, 31
A8_BIN_PEEK_LONG, 32
A8_BIN_POKE_IEEE, 33
A8_BIN_POKE_LONG, 34
A8_BIN_ROV, 35
A8_BIN_SET, 37
A8_BIN_SETIEEE, 39
A8_BIN_SETLONG, 38
A8_BIN_SYS_MASK, 40
ACR8K_FAST_STATUS, 41
ACR8K_FAST_STATUS_REQ, 42
ACR8K_IEEE_MOVE, 43
ACR8K_LONG_MOVE, 47
ACRO_BIN_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-15), 130
ACRO_BIN_READ_MUTEX_TIMEOUT_ERROR (-13), 128
ACRO_BIN_READ_TIMEOUT_ERROR (-17), 132
ACRO_BIN_TRANS_MUTEX_TIMEOUT_ERROR (-19), 134
ACRO_BIN_TRANS_RANGE_VIOLATION_ERROR (-22), 136
ACRO_BIN_TRANS_TIMEOUT_ERROR (-18), 133
ACRO_BIN_TRANS_WAIT_TIMEOUT_ERROR (-20), 135
ACRO_BIN_WRITE_MUTEX_TIMEOUT_ERROR (-14), 129
ACRO_BIN_WRITE_TIMEOUT_ERROR (-16), 131
ACRO_CARD_NOT_FOUND (-10), 126
ACRO_READ_AND_WRITE_MUTEX_TIMEOUT_ERROR (-11), 127
ACRO_READ_MUTEX_TIMEOUT_ERROR (-3), 124
ACRO_READ_TIMEOUT_ERROR (-2), 123
ACRO_SUCCESS (0), 121
ACRO_WRITE_MUTEX_TIMEOUT_ERROR (-4), 125
ACRO_WRITE_TIMEOUT_ERROR (-1), 122
AcroBinaryBufferEmpty, 51
AcroBinaryBufferNumberOfElements, 52
AcroCardPresent, 53
AcroGetBinaryTimeOut, 54
AcroGetBinaryTimerDelta, 55
AcroGetBinTransThreshold, 56
AcroGetDrBinaryBufferState, 57
AcroGetDrBinaryBuffLength, 58
AcroGetDrBinaryBuffSize, 59
AcroGetDrReadBuffLength, 60
AcroGetDrReadBuffSize, 61

AcroGetDrWriteBuffLength, 62
 AcroGetDrWriteBuffSize, 63
 AcroGetError, 64
 AcroGetFastStatusTimerDelta, 65
 AcroGetMaxBinTransPerFSIntr, 66
 AcroGetMaxFastStatusDelta, 67
 AcroGetMoveCounter, 68
 AcroGetPerformance, 69, 71
 AcroGetReadTimeOut, 72
 AcroGetReadTimerDelta, 73
 AcroGetStatusTimerDelta, 74
 AcroGetTimeMilli, 75
 AcroGetTransactionState, 76
 AcroGetVersion, 77
 AcroGetWriteTimeOut, 78
 AcroGetWriteTimerDelta, 79
 AcroHideErrorDialogs, 80
 AcroInitialize, 81
 AcroInitPerformance, 82
 AcroReceiveTextString, 83
 AcroSendEscape, 84, 85
 AcroSendTextString, 86
 AcroSetBinaryTimeout, 87
 AcroSetBinaryTimerDelta, 88
 AcroSetBinTransThreshold, 89
 AcroSetError, 90
 AcroSetFastStatusTimerDelta, 91
 AcroSetInitialWait, 92
 AcroSetMaxBinTransPerFSIntr, 93
 AcroSetMaxFastStatusDelta, 94
 AcroSetMaxSizeofInputBinaryBuffer, 95
 AcroSetMoveCounter, 96
 AcroSetReadTimeout, 97
 AcroSetReadTimerDelta, 98
 AcroSetRetryBeforeBinReadTimeOut, 99
 AcroSetRetryBeforeBinWriteTimeOut, 100
 AcroSetRetryBeforeReadTimeOut, 101
 AcroSetRetryBeforeWriteTimeOut, 102
 AcroSetStatusTimerDelta, 103
 AcroSetTimeOutForBinReadMutex, 104
 AcroSetTimeOutForBinTransMutex, 105
 AcroSetTimeOutForBinTransWait, 106
 AcroSetTimeOutForBinWriteMutex, 107
 AcroSetTimeOutForReadMutex, 108
 AcroSetTimeOutForWriteMutex, 109
 AcroSetTransactionState, 110
 AcroSetWaitBeforeTimeOut, 111
 AcroSetWaitBetweenStatusFetch, 112
 AcroSetWriteTimeout, 113
 AcroSetWriteTimerDelta, 114
 AcroShowErrorDialogs, 115
 AcroTextBufferEmpty, 116
 AcroWait, 117
 Appendix A Parameter Reference, 137
 APPENDIX B FLAG REFERENCE, 139
 APPENDIX C BINARY HOST INTERFACE, 141
 Appendix D Architecture, 143
 Application Level, 144

B

Binary Data Transfer, 141

C

Chapter 1 Overview, 1
Chapter 2 Using AcroLib API, 3
Chapter 3 Function Reference, 9
Chapter 4 Errors, 119

D

Distributing Your Application, 2
DLL Level, 145
Driver Level, 146
Dynamically Changing Move Profiles, 8

E

Establishing Communications with the Card, 4

F

Flag Overview, 139

I

Installing the API Interface, 2

P

Parameter Overview, 137

R

Reading Local User Parameters, 5
Reading Parameters from the Card, 4
Reading System-Defined Parameters, 5
Reading User-Defined Parameters, 5
Requirements for Using ACROLIB Interface, 2

S

Sending a Text File to the Card, 4
Sending ACROBASIC Commands to the Card, 8
Sending Moves to the Card, 7
Setting Local-User Parameters, 7
Setting Parameters on the Card, 6
Setting System-Defined Parameters, 6
Setting Up a Terminal to the Card, 8
Setting User-Defined Parameters, 6
Setting/Clearing Bits on the Card, 7
Synchronizing with Moves on the Card, 8

T

The ACROLIB Interface, 2

W

Windows NT, 120