# HDACS Special Command Instructions

hacHWRWCommandCCH is a lower layer communication command. To run this has to follow the instructions from the Supplier. Or it might cause serious damage to the device

| | | |
|---|---|---|
| Model Supported | | |
| API | hacHWRWCommandCCH | |
| Syntax | int __stdcall hacHWRWCommandCCH(int iRWType,int iNodeID,int iCMD,unsigned char *cSendData,int iSendDataLen, unsigned char *cReceiveData,int *iReceiveLen,HANDLE hComm,unsigned int iTimeout) | |
| Purpose | Send / Receive Basic commands | |
| Parameters description | iRWType | 0,1,2 respectively stand for 3 different command blocks |
| | iNodeID | 1-255 Device ID |
| | iCMD | 0-255 stand for different functions |
| | cSendData | Data send to device |
| | iSendDataLen | Data length to send |
| | cReceiveData | Data received from device<br>**iRWType is 0:**<br>[head:9][data length:1or3][datacmd:1][data:n][crc:2][end:1]<br>Ex:<br>91 01 01 00 00 00 00 00 fd 02 1e 01 33 77 03<br>Head: 91 01 01 00 00 00 00 00 fd<br>data length(datacmd+data):02<br>datacmd:1e<br>data:01<br>crc:33 77<br>end:03<br>**iRWType is1:**<br>[head:9][data length:1or3][datacmd:1][crc:2][end:1]<br>**iRWType is 2:**<br>[head:9][data length:1or3][data:n][crc:2][end:1]<br><br>When returned data length is larger than 255, the data length is 3 bytes |

| | | |
|---|---|---|
| | iReceiveLen | Received data length from device |
| | hComm | handle value, Return value through OpenChannel |
| | iTimeout | Timeout waiting, unit: ms |
| Return Value | If the function succeeds, it will return RET_SUCESS(00h) ; Or a corresponding error message happens | |
| Application | | |
| Sample Code instruction | Function instruction ：<br>Please refer to :C# Sample code | |

## CMD:04(iRWType=2) Get device version and other information

Send:

Data= NULL

Response:

status= 0 :OK

Data= 16 Byte

| Length | Parameter | Address |
|---|---|---|
| 1 | Product Code | 0 |
| 1 | Firmware major version | 1 |
| 1 | Firmware minor version | 2 |
| 1 | Reserved | 3 |
| 1 | Firmware edited year | 4 |
| 1 | Firmware edited month | 5 |
| 1 | Firmware edited date | 6 |
| 4 | Number of cardholders | 7 |
| 4 | Number of transactions | 11 |
| 1 | Reserved | 15 |

status= 1: Error

Data= Error Code (2 byte)

*Description：Model code 0xA1 = RAC-960 PE/PM*

*0xA2 = RAC-960 PEF/PMF/F*

*0xA3 = RAC-960 PMD*

*0xA4 = RAC-960 PMDF*

*0xA6 = RAC-960 PCRF*

*0xB1 = HTA-860 PE/PM*

*0xB2 = HTA-860 PEF/PMF/F*

*0xB3 = HTA-860 PMD*
*0xB4 = HTA-860 PMDF*
*0xC3 = HDE-970 PE/PM*
*0xC5 = HDE-970 PE/PM-R*
*0xD1 = HTA-856 PE/PM*
*0xE1 = RAC-970 PE/PM*
*0xE2 = RAC-970 PEF/PMF*
*0xE3 = RAC-970 PMD*
*0xE4 = RAC-970 PMDF*
*0xF1 = HTA-870 PE/PM*
*0xF2 = HTA-870 PEF/PMF*
*0xF3 = HTA-870 PMD*
*0xF4 = HTA-870 PMDF*
*0x00 = HTA-850 PE/PM*
*0x03 = HTA-852 PEF/PMF*
*0x04 = RAC-852 PxFV*
*0x11 = PXR-96EFSK*
*0x12 = PXR-96MFSK*
*0x13 = PXR-96FSK*
*0x14 = PXR-96CRFSK*
*0x21 = PXR-96EFSKL*
*0x22 = PXR-96MFSKL*
*0x23 = PXR-96FSKL*
*0x31 = PXR-97EFSK*
*0x32 = PXR-97MFSK*
*0x33 = PXR-97FSK*
*0x41 = PXR-97EFSKL*
*0x42 = PXR-97MFSKL*
*0x43 = PXR-97FSKL*

*Firmware major version : Rom File major version*
*Firmware minor version : Rom File minor version*
*Firmware edited year month date：Rom File release date (BCD format)*

## CMD:30(iRWType=0) Get extended setting data

Models supported:RAC-2000WS/RAC-2000WSN

Send:

Data= NULL

Response:

Data= 1 Byte

0x01: HID Reader; Other values: Standard Readers

## CMD:30(iRWType=1) Set extended setting data

Models supported:RAC-2000WS/RAC-2000WSN

Send:

Data= 1 Byte

0x01: HID Reader; Other values: Standard Readers

Note: this command will reboot RAC-2000WS/N