**Purpose of CGI Command (Model)**

 **Update : 20180423**

* Provide the command in CGI format of HTTP in order to make the setting and the searching easy and quick.
* CGI is 2 types. One is the setting and another is the searching.

Setting format: http://ipaddress/basic.cgi?<param>=<value>

 Searching format: http:/ipaddress/basic.cgi?root.action.list=<value>

**1. Video**

- CGI Command : http://ipaddress/basic.cgi?root.video.<field>=<value>

- LIST Command : http://ipaddress/basic.cgi?root.action.list=video

- cf: http://192.168.1.201?basic.cgi?root.video.h264res1=1920x1080

Video can be simultaneously playback in main\_h264, sub\_h264

|  |  |  |
| --- | --- | --- |
| **Filed** | **Value** | **Description** |
| **Main Video H264** |
| h264profile1 | baseline, main |  |
| h264res1 | 1920x1080, 1280x720, 800x600, 704x480, 704x400, 640x480 | Setting the resolution of h264.  |
| h264fps1 | 30, 15, 10, 6, 5, 3, 2, 1 | Setting the framerate  |
| h264gop1 | 1~150 | Setting GOP |
| h264quality1 | 0~51 | Setting Quality |
| h264bps1 | 128k, 256k, 512k, 1m ~ 12m | Setting bps |
| h264bitratemode1 | off, vbr, cbr | Setting bps mode |
| **Sub Video H264** |
| h264res2 | 640x480, 640x360, 320x240 | Setting the resolution of h264.  |
| h264fps2 | 30, 15, 10, 6, 5, 3, 2, 1 | Setting the framerate  |
| h264gop2 | 1 ~ 150 | Setting GOP |
| h264quality2 | 0 ~ 50 | Setting Quality |
| h264bps2 | 128k,256k, 512k, 1~3m | Setting bps |
| h264bitratemode2 | off, vbr, cbr | Setting bps mode |
| **Sub Video JPEG** |
| jpegres2 | 1920x1080 | Setting the resolution of Jpeg |
| Jpegfps2 | 30, 15, 10, 6, 5, 3, 2, 1 | Setting the framerate |
| jpegquality2 | 1 ~ 159 | Setting Quality |
| jpegbps2 | 1m~20m, 25m, 30m | Setting bps |
| jpegbitratemode2 | off, vbr, cbr | Setting bps mode |

2. ISP

 - CGI Command : http://ipaddress/basic.cgi?root.isp.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=isp

 - cf) http://192.168.1.201/basic.cgi?root.isp.sys\_osddata=this\_is\_osddata

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **dscription** |
| **OSD** |
| sys\_osddata | string ( max length: 20) | “Alphabet a ~ z” is only allowed.Not available for the special symbol such as the space. |
| sys\_osdmode | off, rightup, leftdown | Setting OSD text location |
| sys\_osddton | on, off | Setting OSD date/time ON/OFF |
| **Ouptput** |
| sys\_outputfreq | 50, 60 | Setting the frequency  |
| **WDNR** |
| sys\_wdnrmode | off, wdr, dnr | Setting WDR, DNR |
| Sys\_wdnrlevel | Low, mid,high | Setting Level |
| **Defog** |
| sys\_defogon | on, off | Defog on/off |
| sys\_defogmode | manual, auto | Setting Defog mode |
| sys\_defoglevel | low, mid, high | Setting Defog Level |
| **Image** |
| image\_sharpness | 0 ~ 10 | Setting Sharpness |
| image\_saturation | 0 ~ 20 | Setting Saturation |
| image\_gamma | 0.45, 0.55, 0.66, 0.75 | Setting Gamma |
| image\_mirroron | on, off | Mirror on/off |
| image\_flipon | on, off | Flip on/off |
| image\_ace | off, low, mid, high | Setting ACE |
| **Shading** |
| image\_shadingon | on, off | Shading on/off |
| image\_shadingweight | 0 ~ 200 | Setting Shading weight |
| **Color** |
| color\_awb | auto, autoext, manual | Setting AWD mode |
| color\_ctemp | 3000, 5000, 8000 | Setting CTemp value on AWE manual mode.  |
| color\_rgain | 0 ~ 20 | Setting Rgain value on AWE manual mode  |
| color\_bgain | 0 ~ 20 | Setting Bgain value on AWE manual mode  |
| **BackLight** |
| backlight\_mode | off, hlc, blc | Setting BackLight mode |
| backlight\_hlclevel | 0 ~ 20 | Setting Level on BackLight HLC mode |
| backlight\_hlccolor | white, yellow, cyon, green, magenta, red, blue, black, darkgreen | Setting Color on BackLight HLC mode |
| Backlight\_blcblock | 0(xpos, ypos, xsize, ysize)xpos + xsize : 0 ~ 20ypos + ysize : 0 ~ 20 | Setting BLC Block ( Only one cab be set)  |
| **DayNight** |
| daynight\_mode | auto, color, blkwht, extern | Setting DayNight mode |
| daynight\_ispthres | 0 ~ 20 | Setting AGC Threshold on DayNight Auto mode |
| daynight\_ispmargin | 0 ~ 20 | Setting AGC Margin on DayNight Auto mode  |
| daynight\_extthres | 0 ~ 20 | Setting D/N Threadhold on DayNight Extern mode.  |
| daynight\_ extmargin | 0 ~ 20 | Setting N/D Threadhold on DayNight Extern mode |
| daynight\_externsw | high, low | Setting extern sw on DayNight Extern mode |
| daynight\_delay | Low, mid, high | Setting “Delay” on DayNight Auto mode |
| **IR** |
| daynight\_irledon | on, off | IR ON/OFF on DayNight Auto mode |
| daynight\_antisat | 0 ~ 20 | Setting antisat on DayNight Auto mode |
| **Exposure** |
| expo\_brightness | 0 ~ 20 | Setting Brightness  |
| expo\_shutter | auto, manual, flicher | Setting Shutter mode |
| expo\_stmode | normal, deblur | Setting Mode |
| expo\_stspeed | 50, 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600 | Setting the shutter speed on Shutter Manual mode |
| expo\_agc | 0 ~ 10 | Setting AGC on Shutter Manual mode |

3. Network

 - CGI Command : http://ipaddress/basic.cgi?root.network.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=network

 - cf) http://192.168.1.201/basic.cgi?root.network.ipaddress=192.168.1.202

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **description** |
| ipaddress |  xxx.xxx.xxx.xxx |  Setting IP address |
| gateway |  xxx.xxx.xxx.xxx |  Setting Gateway |
| netmask | x xx.xxx.xxx.xxx |  Setting Netmask |
| dns1 | x xx.xxx.xxx.xxx |  Setting First dns |
| dns1 | x xx.xxx.xxx.xxx |  Setting Second dns |
| rtspport | 1 1024 ~ 65353 |  Rtspport |
| httpport | 1 1024 ~ 65353 |  Httpport  |
| upnpon |  on, off |  Upnp on/off |
| upnpnip |  String Max character : 15 |  Upnp device id |
| upnpname | s string Max character : 32 |  Setting Upnp name |
| dhcpon |  on/off |  Setting the network address on DHCP mode |

4. Date/Time

 - CGI Command : http://ipaddress/basic.cgi?root.datetime.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=datetime

 - cf) http://192.168.1.201/basic.cgi?root.datetime.statictime=2017.4.24.3.21.24

|  |  |  |
| --- | --- | --- |
| **filed** | **value** | **Description** |
| sntpon | on, off | Bring the time from SNTP server |
| sntpserver | stringMax characters : 0 ~ 32 | Setting SNTP server |
| statictime | yyyy.mm.dd.hh.mm.ss | Setting the timecf)2017.4.24.3.21.24 |
| timezone | 0 ~ 27 | Setting the time zone |
| dst | on,off | Setting DST ON/FF |

**5. Account**

 - CGI Command : http://ipaddress/basic.cgi?root.account.<field>=<value>

- LIST Command : <http://ipaddress/basic.cgi?root.action.list=account>

When transferring the encrypted password, it will be “root.account.user=”.

If it is not, it will be “root.account.nuser=”.

 cf) The following shows the case that “1234” is encrypted to “TVRJek5BPT0=” and transferred

and another case that “1234” is transferred in “nuser” format as unencrypted. Both two shows the same result.

- cf) http://192.168.1.201/basic.cgi?root.account.**user**=a0(admin,TVRJek5BPT0=,0)

- cf) http://192.168.1.201/basic.cgi?root.account.**nuser**=a0(admin,1234,0)

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| user/nuser | d[index](id, 0,0)[index] : 0 ~ 4 | Delete USER.Cf) Delete id=”admin” on index #0. - d0(admin,0,0) |
| user/nuser | a[index](id, HashPw,level)[index] : 0 ~ 4Level: 0~2 0: administrator 1: operator 2: guest* Refer to appendix for HashPW
 | Change User Information. cf) id=”admin”, pw=”1234” on Index 0, Add level=”administrator”1. User format

a0(admin,TVRJek5BPT0=,0)1. Nuser format

a0(admin,1234,0) |
| user/nuser | e[index](id, id, pw, level)[index] : 0 ~ 4Level: 0~2 0: administrator 1: operator 2: guest | Add user. cf 1) Change the passwordChange the password “id=”admin”” on Index 0 to “3456”.1. User format

e0(admin,admin,TXpRMU5nPT0=,0)2) nuser formate0(admin,admin,3456,0)Change the ID and the password.Change id=admin”, pw=”3456” to id=”root”,pw=”7890”. 1. user format

e0(admin,root,TnpnNU1BPT0=,0)1. nuser format

e0(admin,root,7890,0) |

**6. Motion**

 - CGI Command : http://ipaddress/Basic.cgi?root.motion.<field>=<value>

 - LIST Command : http://ipdaddress/Basic.cgi?root.action.list=motion

 - cf) http://192.168.1.201/basic.cgi?root.motion.block=1(3,3,10, 5)

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| block | [num](xpos, ypos, xsize, ysize) num : 0 ~ 3xpos + xsize : 0 ~ 60ypos + ysize : 0 ~ 34 | Setting the motion detection area. cf) Setting #1 |
| block | [num](0,0,0,0)[num] : 0 ~ 3 |  Disable the motion detection area.  cf) Disable #1 window- 1(0,0,0,0) |
| alarmscreenon | on, off | Display the alarm when the motion is detected.  |
| sensitivity | 0 ~ 10 | Setting the motion sensitivity  |

**7. Privacy**

 - CGI Command : http://ipaddress/basic.cgi?root.privacy.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=privacy

 - cf) http://192.168.1.201/basic.cgi?root.privacy.block=1(3,3,10,5)

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| block | [num](xpos, ypos, xsize, ysize)[num] : 0 ~ 7xpos + xsize : 0 ~ 60ypos + ysize : 0 ~ 34 | Setting Privacy Zone cf) Setting #1 window. - 1(3,3, 10,5) |
| block | [num](0,0,0,0)[num] : 0 ~ 7 | cf) Setting #1 window- 1(0,0,0,0) |
| ylevel | 0 ~ 20 | Setting the privacy windowColor |
| cblevel | 0 ~ 20 |
| crLevel  | 0 ~ 20 |
| translevel | 0, 1, 2, 3 | Setting the privacy window transparency |

**8. OSD**

 - CGI Command : http://ipaddress/basic.cgi?root.osd.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=osd

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| titletext |  string ( max length: 20) | “Alphabet a ~ z“ is only allowed. The special mark such ‘space” is not allowed. |
| title | on, off | Title show  |
| titlepos | rightup, leftdown | Setting Title location |
| titlecolor | green, white, gray, yellow | Setting Title font color |
| datetime | on, off | Date time show |
| datetimeampm | on, off | Setting AM/PM  |
| datetimeline | one, two | Setting Line |
| datetimeweek | on, off | Display Week  |
| datetimeformat | ymd, mdy | Setting Time format |
| datetimecolor | green, white, gray, yellow | Setting Date font color |

**9. commoninfo**

 - CGI Command : http://ipaddress/basic.cgi?root.info.<field>=<value>

 - LIST Command : http://ipaddress/basic.cgi?root.action.list=info

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| productname | string ( max length: 24) | Allow the space between letter |
| devicename | string (max Length:24) | Allow the space between letter |
| manufacture | string ( max length: 24) | Allow the space between letter |
| contact | string ( max length: 24) | Allow the space between letter |
| fax | string ( max length: 24) | Allow the space between letter |
| email | string ( max length: 24) | Allow the space between letter |

EX) http://192.168.1.111/basic.cgi?root.info. productname =Front Camera

**10. Audio**

 - CGI Command : http://ipaddress/basic.cgi?root.audio.<field>=<value>

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| enable | On/off | Audio on/off |
| outgain | 0 ~ 127  | Audio volume level |

cf:) http://192.168.0.71/basic.cgi?root.audio.enable=off

11. **RTSP** - CGI Command : http://ipaddress/basic.cgi?root.rtsp.<field>=<value>

|  |  |  |
| --- | --- | --- |
| **field** | **value** | **Description** |
| auth | enable/disable | Check rtsp id/password  |
|  |  |  |

**12 . Custom Command**

- CGI Command : http://ipaddress/basic.cgi? root.rtsp.ledmode=<value>

|  |  |  |
| --- | --- | --- |
| **command**  | **value** | **Description** |
| root.rtsp.ledmode | 0, 1, 2 | 0: off1: on2: blink |

****cf:) http://192.168.1.201/basic.cgi? root.rtsp.ledmode=2 : Front led blink

 http://192.168.1.201/basic.cgi? root.rtsp.ledmode=1 : Front led on

**APPENDIX**

Encryption code for the password when adding or changing the user account.

//============ Base64.h =============================//

#include <string.h>

#include <stdlib.h>

#pragma once

/\*------ Base64 Encoding Table ------\*/

static const char MimeBase64[] = {

 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',

 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',

 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',

 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',

 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',

 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',

 'w', 'x', 'y', 'z', '0', '1', '2', '3',

 '4', '5', '6', '7', '8', '9', '+', '/',

};

/\*------ Base64 Decoding Table ------\*/

static int DecodeMimeBase64[256] = {

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* 00-0F \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* 10-1F \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,62,-1,-1,-1,63, /\* 20-2F \*/

 52,53,54,55,56,57,58,59,60,61,-1,-1,-1,-1,-1,-1, /\* 30-3F \*/

 -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14, /\* 40-4F \*/

 15,16,17,18,19,20,21,22,23,24,25,-1,-1,-1,-1,-1, /\* 50-5F \*/

 -1,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40, /\* 60-6F \*/

 41,42,43,44,45,46,47,48,49,50,51,-1,-1,-1,-1,-1, /\* 70-7F \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* 80-8F \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* 90-9F \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* A0-AF \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* B0-BF \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* C0-CF \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* D0-DF \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, /\* E0-EF \*/

 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1 /\* F0-FF \*/

};

const unsigned int MAX\_BUF\_SIZE = 1024;

class CBase64

{

public:

 CBase64(void);

 ~CBase64(void);

private:

 char\* m\_pszDecordText;

 char\* m\_pszEncordText;

 int base64\_decode(char \*text, unsigned char \*dst, int numBytes );

 int base64\_encode(char \*text, int numBytes, char \*\*encodedText);

public:

 BOOL Decord\_Base64( CString strText, CString& strResult );

 BOOL Encord\_BASE64( CString strText, CString& strResult );

};

//==================== Base64.cpp ==============================//

#include "stdafx.h"

#include "Base64.h"

CBase64::CBase64(void)

{

 // Init Original Data

 m\_pszDecordText = NULL;

 m\_pszEncordText = NULL;

}

CBase64::~CBase64(void)

{

 // Delete Dynamic Data

 if(m\_pszDecordText){

 delete [] m\_pszDecordText;

 m\_pszDecordText = NULL;

 }

 if(m\_pszEncordText){

 delete [] m\_pszEncordText;

 m\_pszEncordText = NULL;

 }

}

int CBase64::base64\_decode(char \*text, unsigned char \*dst, int numBytes )

 {

 const char\* cp;

 int space\_idx = 0, phase;

 int d, prev\_d = 0;

 unsigned char c;

 space\_idx = 0;

 phase = 0;

 for ( cp = text; \*cp != '\0'; ++cp ) {

 d = DecodeMimeBase64[(int) \*cp];

 if ( d != -1 ) {

 switch ( phase ) {

 case 0:

 ++phase;

 break;

 case 1:

 c = ( ( prev\_d << 2 ) | ( ( d & 0x30 ) >> 4 ) );

 if ( space\_idx < numBytes )

 dst[space\_idx++] = c;

 ++phase;

 break;

 case 2:

 c = ( ( ( prev\_d & 0xf ) << 4 ) | ( ( d & 0x3c ) >> 2 ) );

 if ( space\_idx < numBytes )

 dst[space\_idx++] = c;

 ++phase;

 break;

 case 3:

 c = ( ( ( prev\_d & 0x03 ) << 6 ) | d );

 if ( space\_idx < numBytes )

 dst[space\_idx++] = c;

 phase = 0;

 break;

 }

 prev\_d = d;

 }

 }

 return space\_idx;

}

int CBase64::base64\_encode(char \*text, int numBytes, char \*\*encodedText)

 {

 unsigned char input[3] = {0,0,0};

 unsigned char output[4] = {0,0,0,0};

 int index, i, j, size;

 char \*p, \*plen;

 plen = text + numBytes - 1;

 size = (4 \* (numBytes / 3)) + (numBytes % 3? 4 : 0) + 1;

 (\*encodedText) = (char\*)malloc(size);

 j = 0;

 for (i = 0, p = text;p <= plen; i++, p++) {

 index = i % 3;

 input[index] = \*p;

 if (index == 2 || p == plen) {

 output[0] = ((input[0] & 0xFC) >> 2);

 output[1] = ((input[0] & 0x3) << 4) | ((input[1] & 0xF0) >> 4);

 output[2] = ((input[1] & 0xF) << 2) | ((input[2] & 0xC0) >> 6);

 output[3] = (input[2] & 0x3F);

 (\*encodedText)[j++] = MimeBase64[output[0]];

 (\*encodedText)[j++] = MimeBase64[output[1]];

 (\*encodedText)[j++] = index == 0? '=' : MimeBase64[output[2]];

 (\*encodedText)[j++] = index < 2? '=' : MimeBase64[output[3]];

 input[0] = input[1] = input[2] = 0;

 }

 }

 (\*encodedText)[j] = '\0';

 return size;

 }

BOOL CBase64::Decord\_Base64( CString strText, CString& strResult )

{

 // Init String Buff

 if(m\_pszDecordText){

 delete [] m\_pszDecordText;

 m\_pszDecordText = NULL;

 }

 m\_pszDecordText = new char[MAX\_BUF\_SIZE];

 memset( m\_pszDecordText, 0, sizeof(char)\*MAX\_BUF\_SIZE );

 // Change MBCS

 CStringA strA = (CStringA)strText;

 sprintf\_s( m\_pszDecordText, sizeof(char)\*MAX\_BUF\_SIZE, strA.GetBuffer() );

 char\* pchBase64Decorded = NULL;

 pchBase64Decorded = new char[MAX\_BUF\_SIZE];

 memset( pchBase64Decorded, 0, sizeof(char)\*MAX\_BUF\_SIZE );

 int iBase64DecodeLen = base64\_decode( m\_pszDecordText, (unsigned char\*)pchBase64Decorded, MAX\_BUF\_SIZE );

 CStringA strResultA;

 strResultA.Format("%s", pchBase64Decorded );

 // Set Result

 strResult = (CString)strResultA;

 // Delete Dynamic Data

 if(pchBase64Decorded){

 delete [] pchBase64Decorded;

 pchBase64Decorded = NULL;

 }

 if(m\_pszDecordText){

 delete [] m\_pszDecordText;

 m\_pszDecordText = NULL;

 }

 return TRUE;

}

BOOL CBase64::Encord\_BASE64( CString strText, CString& strResult )

{

 // Init String Buffer

 if(m\_pszEncordText){

 delete [] m\_pszEncordText;

 m\_pszEncordText = NULL;

 }

 m\_pszEncordText = new char[MAX\_BUF\_SIZE];

 memset( m\_pszEncordText, 0, sizeof(char)\*MAX\_BUF\_SIZE );

 // Change MBCS

 CStringA strA = (CStringA)strText;

 sprintf\_s( m\_pszEncordText, sizeof(char)\*MAX\_BUF\_SIZE, strA.GetBuffer() );

 // Encord BASE64

 char\* pchBase64Encorded = NULL;

 int iBase64EncodeLen = base64\_encode( m\_pszEncordText, strlen(m\_pszEncordText), &pchBase64Encorded );

 CStringA strResultA;

 strResultA.Format( "%s", pchBase64Encorded );

 // Set Result

 strResult = (CString)strResultA;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 Caution !!

 First Param is Plain Text Pointer

 Second Param is Plain Text Length

 Third Param is Result Double Pointer ( Just do Free !! )

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 if(pchBase64Encorded ){

 free(pchBase64Encorded);

 pchBase64Encorded = NULL;

 }

 if(m\_pszEncordText){

 delete [] m\_pszEncordText;

 m\_pszEncordText = NULL;

 }

 return TRUE;

}

//====test.cpp ========//

MakeHashPw(CString \_strSrc, CString &\_strDst)

{

// Base 64 Encord

 CString strResult = \_T("");

 CBase64 Base64;

 // Make PW Hash

 Base64.Encord\_BASE64( \_strSrc , strResult );

 Base64.Encord\_BASE64( strResult, \_strDst );

 return true;

}

Void main ()

{

CString strHashPW;

CString strCgiCommand;

//1234를 hashcode화 한다.

MakeHashPw(L“1234”, strHashPW);

//hashPw화된 값을 user형식에 대입한다.

strCgiCommand.Format (L”192.168.0.201/basic.cgi?root.account.user=a0(admin,%s,0)”, strHashPW);

sendCommand (strCgiCommand);

}