```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
#define MULTITAG_COUNTMAX  20
typedef struct {
  char  szFormat[16];
  char  szTag[256];
  void *pvVal;
} MultiTag_type;

BOOL bOK;
int i, iCount;
static int iCnt = 0;

static BOOL   b001;
DWORD   by001;
DWORD   w001;
DWORD    dw001;
long   sby001;
long    i001;
DWORD    l001;
float f001;
char    sz001[256], sz002[256];
DWORD TI4711_wVal;
DWORD adwState[MULTITAG_COUNTMAX  ];
DWORD dwState, *pdwState;

MultiTag_type  ascTags[MULTITAG_COUNTMAX  ] ={
  {"%d", "CRT3894_b001",      &b001},
  {"%d", "CRT3894_by001",    &by001},
  {"%d", "CRT3894_w001",      &w001},
  {"%d", "CRT3894_dw001",   &dw001},
  {"%d", "CRT3894_sby001", &sby001},
  {"%d", "CRT3894_i001",       &i001},
  {"%d", "CRT3894_l001",       &l001},
  {"%f",   "CRT3894_f001",     &f001},
  {"%s", "CRT3894_sz001",  sz001},
  {"%s", "CRT3894_sz002",  sz002},
  {"%d", "TI4711_wVal",         &TI4711_wVal},
  {"END", "END", NULL}
};
MultiTag_type   *psc;
char szFormat[256];

//=====================================
// (2) prepare format string
//=====================================
for (i=0, psc=&ascTags[0], *szFormat='\0'; i<MULTITAG_COUNTMAX  ; i++, psc++){
  if (strcmp(psc->szFormat, "END") == 0) break;
  strcat (szFormat, psc->szFormat);
}//for
iCount = i;

//=====================================
// (3) increment values
//=====================================
iCnt++;
b001 = 0x01 & iCnt;
by001 = iCnt;
w001= iCnt;
dw001= 3*iCnt + 0x7FFFFFFA;
sby001= iCnt;
i001= iCnt;
l001= 0x7FFFFFFE + iCnt;
f001= iCnt;
sprintf(sz001, "Hallo %d!", iCnt);
sprintf(sz002, "World %d!", iCnt);
TI4711_wVal = iCnt;

printf ("#I101: iCnt=%d   szFormat=\"%s\"    by001=%d\r\n", iCnt, szFormat, by001 );
//=====================================
// (4) call SetTagMulti() function
//=====================================
bOK = SetTagMultiStateWait(adwState, szFormat,
//bOK = SetTagMultiWait(szFormat,
     ascTags[0].szTag,
  *((BYTE*)ascTags[0].pvVal),
     ascTags[1].szTag,
  *((DWORD*)ascTags[1].pvVal),
  ascTags[2].szTag,
  *((DWORD*)ascTags[2].pvVal),
  ascTags[3].szTag,
  *((DWORD*)ascTags[3].pvVal),
  ascTags[4].szTag,
  *((DWORD*)ascTags[4].pvVal),
  ascTags[5].szTag,
  *((DWORD*)ascTags[5].pvVal),
```

```
  ascTags[6].szTag,
  *((DWORD*)ascTags[6].pvVal),
  ascTags[7].szTag,
  *((float*)ascTags[7].pvVal),
  ascTags[8].szTag,
  (char *)ascTags[8].pvVal,
  ascTags[9].szTag,
  (char *)ascTags[9].pvVal,
  ascTags[10].szTag,
  *((DWORD*)ascTags[10].pvVal)
);
if (!bOK){
  printf ("#E405: - SetTagMultiWait()  failed!  \r\n");
  return ;
}
//====================================
// (5) display values of tags
//====================================
printf ("#I208: szFormat=\"%s\"\r\n", szFormat);
for (i=0, pdwState=adwState,  psc=ascTags; i < iCount; i++, pdwState++, psc++){
  dwState = *pdwState;  //data manager variable state
  dwState &= ((DM_VARSTATE_STARTUP_VALUE) ^ 0xFFFF);   //ignore  some status bits
  if (dwState != 0x0000){
    printf ("#E210:  Tag=\"%s\"  DM_VARSTATE=0x%04X    \r\n", psc->szTag, *pdwState);
  }
  if (strcmp(psc->szFormat, "%d") == 0){
    printf ("\ti=%3d  \"%s\"\t=%d  (0x%08X)  dwState=0x%08X\r\n",
      i, psc->szTag, *((int*)psc->pvVal), *((int*)psc->pvVal), *pdwState);
  } else if (strcmp(psc->szFormat, "%f") == 0){
    printf ("\ti=%3d  \"%s\"\t=%f    dwState=0x%08X \r\n",
      i, psc->szTag,  *((float*)psc->pvVal), *pdwState);
  }  else if (strcmp(psc->szFormat, "%s") == 0){
    printf ("\ti=%3d  \"%s\"\t=\"%s\"    dwState=0x%08X \r\n",
      i, psc->szTag,  ((char*)psc->pvVal), *pdwState);
  }
}//for i




return ;
}
```