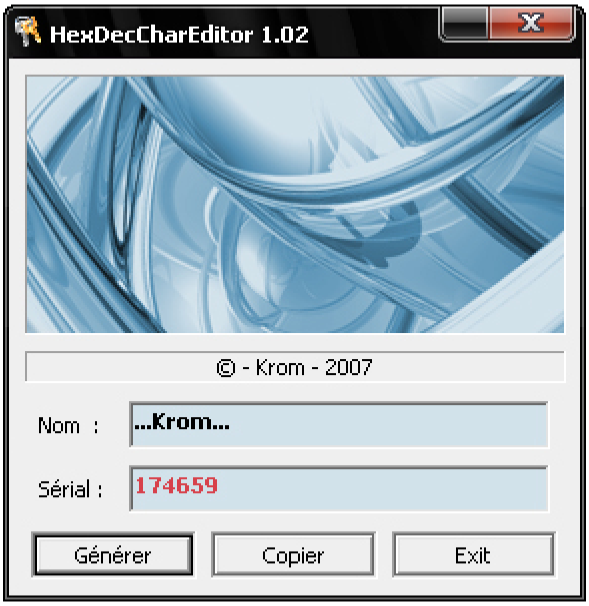
Cours N° 6

Ce cours vous montrera comment créer un Keygen avec cette fois une interface graphique.
Voici le résultat du Keygen :

Télécharger le Keygen :

- http://www.KromCrack.com/prog/Keygen-Krom.exe



HexDecCharEditor 1.02

© - Krom - 2007

Nom : ...Krom...

Sérial : 174659

L'algorithme reste le même mais la façon de le coder elle, est différente sur un Keygen en fenêtre. Je vais faire comme dans le cours précédent, vous donner le code source du Projet et vous expliquer ensuite en détail le code.

Vous pouvez télécharger l'espace de travail complet du Keygen (Code source, .exe, images et icône ...) :

Espace de travail du Keygen :

- http://www.KromCrack.com/prog/Keygen-GUI.exe

Keygen.c

Le fichier Keygen.c contient toutes les fonctions importantes pour l'affichage de la fenêtre.

```
/********************************************************************
*        Keygen.c                                                   *
*                                                                   *
*        - Keygen pour Hexa.exe                                     *
*                                                                   *
*        Auteur                  : - Krom -                         *
*        Date de Création        : - 07 - 12 - 2007 -               *
*        Dernière modification   : - 11 - 02 - 2008 -               *
*                                                                   *
********************************************************************/

#define  WIN32_LEAN_AND_MEAN
#include <windows.h>
#include "identifiers.h"

#define CR_FIELD    RGB(210, 226, 235)
#define CR_INFIELD  RGB(0  , 0  , 0  )
#define CR_OUTFIELD RGB(215, 65 , 75 )

BOOL Init(HWND);
BOOL Gen (HWND);
BOOL Copy(HWND);
BOOL Exit(HWND);

BOOL CALLBACK DlgProc    (HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK InfoDlgProc(HWND, UINT, WPARAM, LPARAM);
BOOL EnableDlgItem(HWND, DWORD, BOOL);
BOOL SetStatusText(HWND, LPCTSTR);

#include "core.c"

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPTSTR lpCmdLine, int nCmdShow) {
  return DialogBox(hInstance, MAKEINTRESOURCE(IDD_KEYGEN), NULL, DlgProc);
}

BOOL CALLBACK DlgProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {
  static HINSTANCE hInstance;
  static HBRUSH hFieldColor;
  static HFONT hBoldFont;
  HICON hIcon;
  LOGFONT BoldFont;
  int nFieldId;

  #if !AUTO_UPDATE
    TCHAR sGen[] = TEXT("Hit Generate, please");
  #endif

  switch (uMsg) {

    case WM_COMMAND:

      switch(LOWORD(wParam)) {


        case IDC_INFO:
          DialogBox(hInstance, MAKEINTRESOURCE(IDD_INFO), hWnd, InfoDlgProc);
          break;


        case IDC_GEN:

          if(Gen(hWnd)) {

            #if !AUTO_UPDATE && !MULTI_SERIALS
              EnableDlgItem(hWnd, IDC_GEN, FALSE);
            #endif

            EnableDlgItem(hWnd, IDC_COPY, TRUE);
          }
          else {
            EnableDlgItem(hWnd, IDC_GEN, FALSE);
            EnableDlgItem(hWnd, IDC_COPY, FALSE);
          }

          break;


        case IDC_COPY:

          if(Copy(hWnd))
            EnableDlgItem(hWnd, IDC_COPY, FALSE);

          break;


        case IDC_EXIT:
          SendMessage(hWnd, WM_CLOSE, 0, 0);
          break;


        default:

          if(LOWORD(wParam) >= INFIELD_BEGIN && LOWORD(wParam) <= INFIELD_END && HIWORD(wParam) ==
EN_UPDATE) {

            #if AUTO_UPDATE

              if(Gen(hWnd)) {

                #if MULTI_SERIALS
                  EnableDlgItem(hWnd, IDC_GEN, TRUE);
                #endif

                EnableDlgItem(hWnd, IDC_COPY, TRUE);
              }
              else {

                #if MULTI_SERIALS
                  EnableDlgItem(hWnd, IDC_GEN, FALSE);
                #endif

                EnableDlgItem(hWnd, IDC_COPY, FALSE);
              }

            #else
              EnableDlgItem(hWnd, IDC_GEN, TRUE);
              EnableDlgItem(hWnd, IDC_COPY, FALSE);
              SetStatusText(hWnd, sGen);
            #endif

          }
          else
            return FALSE;
      }

      break;

    case WM_INITDIALOG:
      hInstance = GetModuleHandle(NULL);
      hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_KEYGEN));

      SendMessage(hWnd, WM_SETICON, ICON_BIG, (LPARAM) hIcon);
      SendMessage(hWnd, WM_SETICON, ICON_SMALL,(LPARAM) hIcon);



      hFieldColor = CreateSolidBrush(CR_FIELD);
      GetObject((HFONT) SendMessage(hWnd, WM_GETFONT, 0, 0), sizeof(LOGFONT), &BoldFont);
      BoldFont.lfWeight = FW_BOLD;
      hBoldFont = CreateFontIndirect(&BoldFont);

      for(nFieldId = OUTFIELD_BEGIN; nFieldId <= OUTFIELD_END; nFieldId++)

        if(GetDlgItem(hWnd, nFieldId) != NULL)
          SendDlgItemMessage(hWnd, nFieldId, WM_SETFONT, (WPARAM) hBoldFont, TRUE);
        else
          break;
      for(nFieldId = INFIELD_BEGIN; nFieldId <= INFIELD_END; nFieldId++)

        if(GetDlgItem(hWnd, nFieldId) != NULL)
          SendDlgItemMessage(hWnd, nFieldId, WM_SETFONT, (WPARAM) hBoldFont, TRUE);
        else
          break;


      #ifdef TARGET_VERSION
        SetWindowText(hWnd, TEXT(TARGET " " TARGET_VERSION " " TYPE));
      #else
        SetWindowText(hWnd, TEXT(TARGET " " TYPE));
      #endif


      if(!Init(hWnd)) {
        SendMessage(hWnd, WM_CLOSE, 0, 0);
        break;
      }

      #if AUTO_UPDATE

        #if !MULTI_SERIALS
          EnableDlgItem(hWnd, IDC_GEN, FALSE);
        #endif

        if(!Gen(hWnd)) {

          #if MULTISERIALS
            EnableDlgItem(hWnd, IDC_GEN, FALSE);
          #endif

          EnableDlgItem(hWnd, IDC_COPY, FALSE);
        }

      #else
        EnableDlgItem(hWnd, IDC_COPY, FALSE);
        SetDlgItemText(hWnd, IDC_STATUS, sGen);
      #endif

      break;

    case WM_CLOSE:

      Exit(hWnd);

      DeleteObject(hBoldFont);
      DeleteObject(hFieldColor);

      EndDialog(hWnd, TRUE);
      break;


    case WM_CTLCOLORSTATIC:
      nFieldId = GetDlgCtrlID((HWND) lParam);

      if(nFieldId >= OUTFIELD_BEGIN && nFieldId <= OUTFIELD_END) {
        SetBkColor((HDC) wParam, CR_FIELD);
        SetTextColor((HDC) wParam, CR_OUTFIELD);
        return (LRESULT) hFieldColor;
      }
      else
        return FALSE;


    case WM_CTLCOLOREDIT:
      SetBkColor((HDC) wParam, CR_FIELD);
      SetTextColor((HDC) wParam, CR_INFIELD);
      return (LRESULT) hFieldColor;

    default:
      return FALSE;

  }

  return TRUE;
}
```

```
BOOL CALLBACK InfoDlgProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {

  switch(uMsg) {

    case WM_COMMAND:

      switch(LOWORD(wParam)) {

        case IDOK:
          EndDialog(hWnd, TRUE);

        default:
          return FALSE;

      }

      break;

    case WM_INITDIALOG:

      SetDlgItemText(hWnd, IDC_TITLE, TEXT("HexDecCharEditor 1.02 - Keygen"));


      break;

    default:
      return FALSE;

  }

  return TRUE;
}

BOOL EnableDlgItem(HWND hWnd, DWORD nDlgItem, BOOL bEnable) {
  return EnableWindow(GetDlgItem(hWnd, nDlgItem), bEnable);
}

BOOL SetStatusText(HWND hWnd, LPCTSTR sText) {
  return SetDlgItemText(hWnd, IDC_STATUS, sText);
}
```

Core.c

Le fichier Core.c contient toutes les fonctions importantes pour la création et l'affichage du Sérial.

```
/*********************************************************************
*        Core.c                                                      *
*                                                                    *
*        - Keygen pour Hexa.exe                                      *
*                                                                    *
*        Auteur                 : - Krom -                           *
*        Date de Création       : - 07 - 12 - 2007 -                 *
*        Dernière modification  : - 11 - 02 - 2008 -                 *
*                                                                    *
*********************************************************************/

#define ID             "1"
#define TARGET         "HexDecCharEditor"
#define TARGET_VERSION "1.02"
#define TYPE           ""

#define NOTE "HexDecCharEditor 1.02 Keygen - By Krom"

#define AUTO_UPDATE   TRUE
#define MULTI_SERIALS TRUE


CHAR sInfield1 [0x100];
CHAR sOutfield1[0x100];

BOOL Init(HWND hWnd) {
  DWORD nInfield1Size = sizeof sInfield1;


  SendDlgItemMessage(hWnd, IDC_INFIELD1, EM_SETLIMITTEXT, sizeof sInfield1 - 1, 0);


  GetUserName(TEXT(""), &nInfield1Size);

  SetDlgItemText(hWnd, IDC_INFIELD1, sInfield1);

  SetStatusText(hWnd, TEXT("Keygen ready ..."));
  return TRUE;
}

BOOL Gen(HWND hWnd) {


  if(GetDlgItemText(hWnd, IDC_INFIELD1, sInfield1, sizeof sInfield1)) {

        int result = 0;
        int serial = 0;
        int count = 0;
        int nbrChar = 0;
        char Chaine[32];
        char ChaineSerial[32];
        long algo1[32] = {0xD5, 0x00, 0xF7, 0x24, 0x5D, 0x5C, 0x0A, 0x8C,
        0x10, 0xD6, 0x9E, 0xB5, 0x9C, 0x66, 0x00, 0x24, 0x6F, 0x20, 0x44, 0x4D,
        0xD4, 0x63};
        for(int i = 0; i < sizeof(Chaine);i++){
                Chaine[i] = 0;
        }
        for(int i = 0; i < sizeof(ChaineSerial);i++){
                ChaineSerial[i] = 0;
        }
        for(nbrChar = 0;sInfield1[nbrChar] != 0;nbrChar++){

        }
        for(int count2 = 0; count2 < nbrChar; count2++)
        {
             result = algo1[count2] ^ sInfield1[count2];
             serial += result << count2;
        }
        for(int i = 0;serial > 0; i++){
              Chaine[i] = ((serial % 10)+48);
              serial = serial / 10;
        }
        count = 0;
        for(int i = 0;Chaine[i] != 0;i++){
                count++;
        }
        for(int i = 0;i < count;i++){
                ChaineSerial[count-i-1] = Chaine[i];
        }
    if(nbrChar >= 10){
        SetDlgItemText(hWnd, IDC_OUTFIELD1, ChaineSerial);
    }
    else
    {
        SetDlgItemText(hWnd, IDC_OUTFIELD1, "Entrez plus de 10 caractères");
    }


    SetStatusText(hWnd, TEXT("© - Krom - 2007"));
    return TRUE;
  }
  else {
    SetStatusText(hWnd, TEXT("© - Krom - 2007"));
    SetDlgItemText(hWnd, IDC_OUTFIELD1, 0);
    return FALSE;
  }

}
BOOL Copy(HWND hWnd) {
  HGLOBAL hOutfield1;

  GetDlgItemText(hWnd, IDC_OUTFIELD1, sOutfield1, sizeof sOutfield1);


  hOutfield1 = GlobalAlloc(GMEM_MOVEABLE | GMEM_DDESHARE, sizeof sOutfield1);
  lstrcpy(GlobalLock(hOutfield1), sOutfield1);
  GlobalUnlock(hOutfield1);


  OpenClipboard(hWnd);
  EmptyClipboard();
  SetClipboardData(CF_TEXT, hOutfield1);
  CloseClipboard();

  SetStatusText(hWnd, TEXT("© - Krom - 2007"));
  return TRUE;
}
BOOL Exit(HWND hWnd) {
  SetStatusText(hWnd, TEXT("See you later..."));
  return TRUE;
}
```

Ressources.rc

Voici le fichier ressource contenant les différents boutons, image et icône :

```
/*********************************************************************
*        ressources.rc                                               *
*                                                                    *
*        - Keygen pour Hexa.exe                                      *
*                                                                    *
*        Auteur                  : - Krom -                          *
*        Date de Création        : - 07 - 12 - 2007 -                *
*        Dernière modification   : - 11 - 02 - 2008 -                *
*                                                                    *
*********************************************************************/

#define  WIN32_LEAN_AND_MEAN
#include <windows.h>

#include "identifiers.h"

IDI_KEYGEN ICON   "icon.ico"

IDB_KEYGEN BITMAP "keygen.bmp"

IDD_KEYGEN DIALOGEX 0, 0, 190, 165
STYLE WS_POPUPWINDOW | WS_CAPTION | DS_CENTER
FONT 8, "Tahoma"
BEGIN

  CONTROL IDB_KEYGEN, IDC_STATIC, "STATIC", SS_BITMAP | SS_CENTERIMAGE, 5, 5, 180, 80,
WS_EX_STATICEDGE

  LTEXT "", IDC_STATUS, 5 , 90, 180, 10, SS_CENTER, WS_EX_STATICEDGE

  LTEXT "Nom  :", IDC_STATIC, 10, 105, 25, 15, SS_CENTERIMAGE
  EDITTEXT IDC_INFIELD1, 40, 105, 140, 15, NOT WS_BORDER | ES_AUTOHSCROLL, WS_EX_CLIENTEDGE

  LTEXT "Sérial :", IDC_STATIC, 10, 125, 25, 15, SS_CENTERIMAGE
  EDITTEXT IDC_OUTFIELD1, 40, 125, 140, 15, NOT WS_BORDER | ES_READONLY, WS_EX_CLIENTEDGE

  DEFPUSHBUTTON "&Générer", IDC_GEN , 7   , 145, 55, 15, BS_DEFPUSHBUTTON, WS_EX_CLIENTEDGE
  PUSHBUTTON    "&Copier" , IDC_COPY, 67  , 145, 55, 15, BS_PUSHBUTTON   , WS_EX_CLIENTEDGE
  PUSHBUTTON    "E&xit"   , IDC_EXIT, 127 , 145, 55, 15, BS_PUSHBUTTON   , WS_EX_CLIENTEDGE
END
```

Algo.c

Voici le détail de la création du code :

```
BOOL Gen(HWND hWnd)
{
  if(GetDlgItemText(hWnd, IDC_INFIELD1, sInfield1, sizeof sInfield1))
  {
        int result = 0;
        int serial = 0;
        int count = 0;
        int nbrChar = 0;
        char Chaine[32];
        char ChaineSerial[32];
        long algo1[32] = {0xD5, 0x00, 0xF7, 0x24, 0x5D, 0x5C, 0x0A, 0x8C,
        0x10, 0xD6, 0x9E, 0xB5, 0x9C, 0x66, 0x00, 0x24, 0x6F, 0x20, 0x44, 0x4D,
        0xD4, 0x63};
        for(int i = 0; i < sizeof(Chaine);i++){
                Chaine[i] = 0;
        }
        for(int i = 0; i < sizeof(ChaineSerial);i++){
                ChaineSerial[i] = 0;
        }
        for(nbrChar = 0;sInfield1[nbrChar] != 0;nbrChar++){

        }
        for(int count2 = 0; count2 < nbrChar; count2++)
        {
             result = algo1[count2] ^ sInfield1[count2];
             serial += result << count2;
        }
        for(int i = 0;serial > 0; i++){
              Chaine[i] = ((serial % 10)+48);
              serial = serial / 10;
        }
        count = 0;
        for(int i = 0;Chaine[i] != 0;i++){
                count++;
        }
        for(int i = 0;i < count;i++){
                ChaineSerial[count-i-1] = Chaine[i];
        }
    if(nbrChar >= 10){
        SetDlgItemText(hWnd, IDC_OUTFIELD1, ChaineSerial);
    }
    else
    {
        SetDlgItemText(hWnd, IDC_OUTFIELD1, "Entrez plus de 10 caractères");
    }
    SetStatusText(hWnd, TEXT("© - Krom - 2007"));
    return TRUE;
  }
  else {
    SetStatusText(hWnd, TEXT("© - Krom - 2007"));
    SetDlgItemText(hWnd, IDC_OUTFIELD1, 0);
    return FALSE;
  }

}
```

J'espère que ce cours a été clair ;)

Si vous avez rencontré une erreur ou que quelque chose ne marche pas, vous pouvez m'envoyer un mail à **Admin@KromCrack.com** ou en parler sur le forum :

- **http://www.KromCrack.com/forum/**