CLP - Guide to Automated Subkects*

leandro alencar

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

Código ladder para um programa simples

Aula 8 – "AppWizard"

1. Introdução

Nas aulas anteriores, todos os programas exemplo foram construídos a partir do zero, cada linha de código teve que ser inserida pelo programador. A partir desta aula, os programas exemplo serão gerados a partir de um programa esqueleto gerado pelo "AppWizard".

1. Criando um Programa com o "Appwizard"

A sequência de figuras a seguir ilustra a criação de um programa utilizando o "AppWizard" do Visual C++ 6.0.

Selecione New no menu principal. Depois selecione MFCAppWizard(exe), de um nome para o projeto e pressione OK.

*Version 1.1: Jun 20, 2008 10:48 pm -0500

†http://creativecommons.org/licenses/by/2.0/

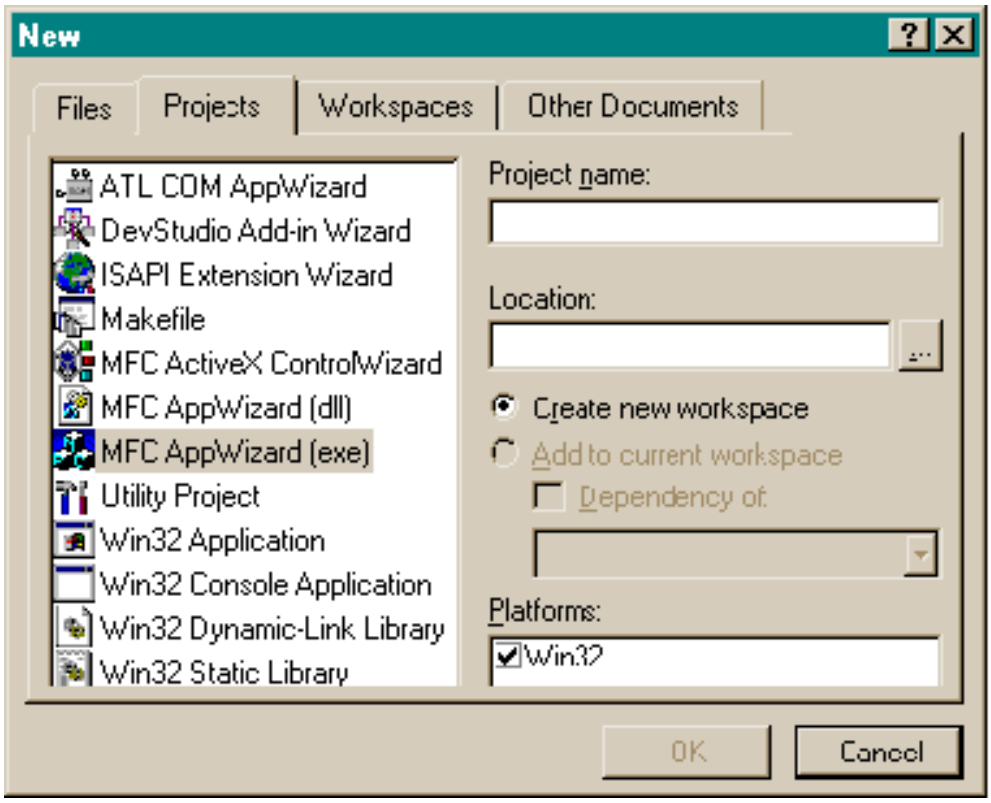

Figure 1

Na janela "MFC AppWizard – Step 1", selecione "Single Document" e desmarque a opção "Document/View architecture".

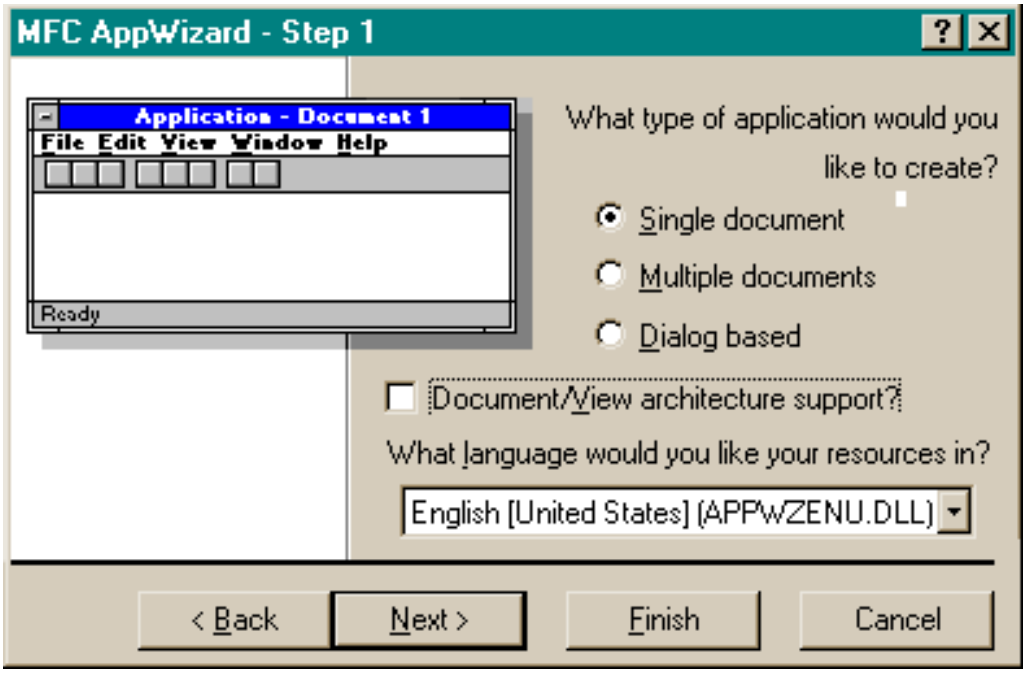

Figure 2

Em “MFC AppWizard – Step 2”, selecione “None”.

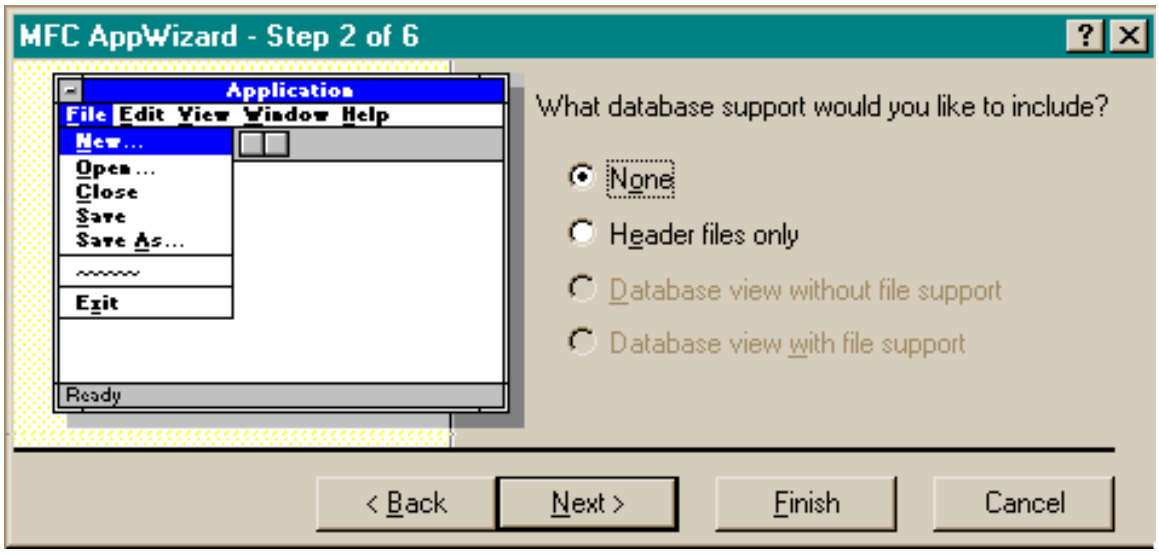

Figure 3

Da janela "Step 3" até a "Step 6" mantenha os "defaults".

Image not finished

Figure 4

Image not finished

Figure 5

Image not finished

Figure 6

Image not finished

Figure 7

Na janela “Step 6”, pressione “Finish” e o “AppWizard” fornecerá uma lista de informações sobre o novo projeto que será gerado. Os arquivos serão gerados quando você pressionar “OK”.

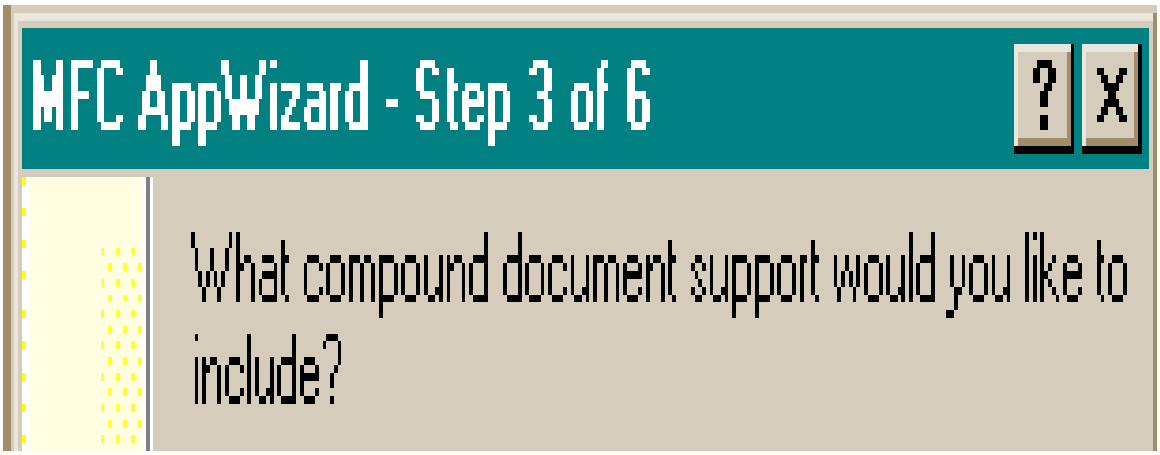

Figure 8

O Programa Gerado
Os arquivos gerados são mostrados na janela “Workspace”.

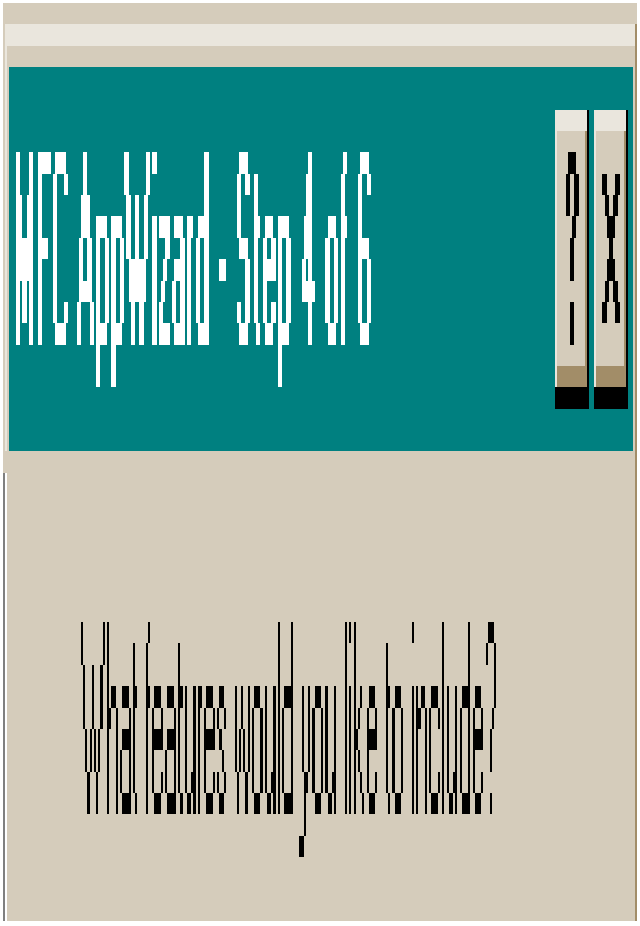

Figure 9

Compile e execute o programa. O resultado deve ser parecido com o mostrado abaixo.

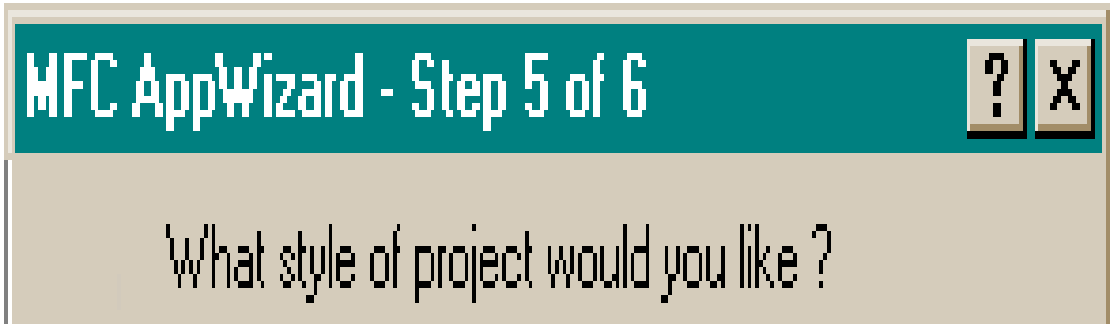

Figure 10

IMPORTANTE: O "AppWizard" utiliza comentários especiais para identificar partes do código gerado. Por exemplo:

```
//{{AFX_MSG_MAP(CChildView)
ON_WM_PAINT()
//}}AFX_MSG_MAP
```

Não modifique nenhuma linha entre esses comentários.

1. Entendendo o Código Gerado pelo "AppWizard"

Esta seção visa explicar as principais funções presentes no código gerado pelo "AppWizard". Para melhor entendimento, sugere-se que o leitor acompanhe as explicações com o código do programa aberto no Microsoft Visual C++.

- Estrutura da janela

A janela do programa gerado é, na verdade, constituída de duas janelas: uma janela principal associada à classe "CMainFrame" e uma janela ("filha" da principal) associada à classe "CChildView". A figura seguinte ilustra essa estrutura.

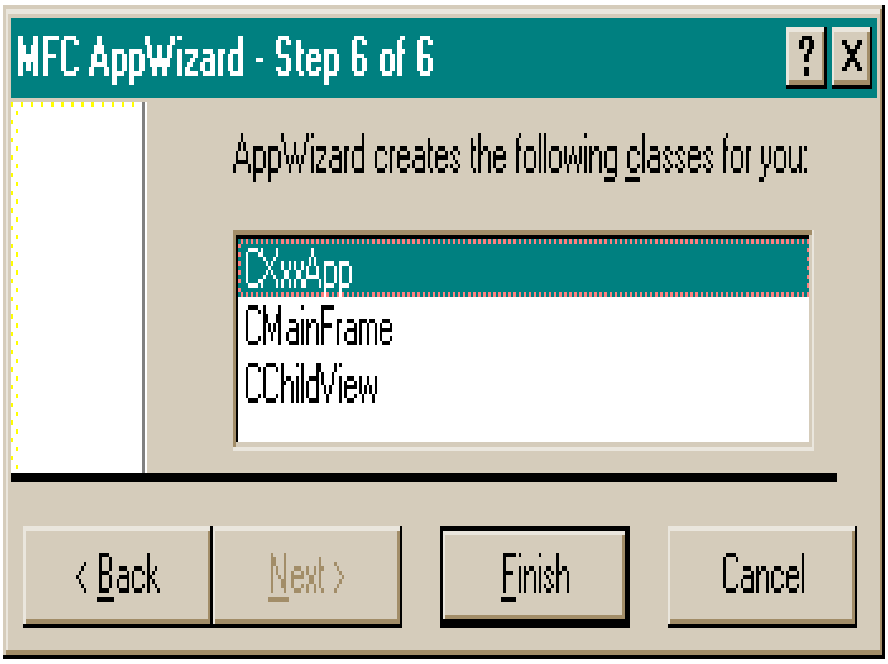

Figure 11

O menu, a barra de ferramentas e a barra de status não são janelas, eles são itens que pertencem à janela principal e serão estudados nas próximas aulas.

Como indicado pela figura anterior, a janela "CChildView" ocupa toda a região vazia da janela "CMainFrame".

- Estrutura do Programa

As principais tarefas realizadas pelo código de um programa gerado pelo "AppWizard" podem ser resumidas nas seguintes etapas:

- Criação da janela "CMainFrame";
- Criação do menu e das barras de status e de ferramentas, quando for o caso;
- Criação da janela "CChildView";
- Estabelecimento da comunicação entre o usuário e a janela "filha".
- Criação da janela "CMainFrame"

O processo de criação da janela principal começa, como nos programas das aulas anteriores, na função "InitInstance()" da classe da aplicação ("CXxxApp" para o caso do programa gerado na seção 2 desta aula):

```
CMainFrame* pFrame = new CMainFrame;
m_pMainWnd = pFrame;
```

Antes da janela ser realmente criada, a função "CMainFrame::PreCreateWindow()" é chamada. No programa gerado anteriormente, o código dessa função é:

```
BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
```

```
if( !CFrameWnd::PreCreateWindow(cs) )return FALSE;
cs.dwExStyle &= ~WS_EX_CLIENTEDGE;
cs.lpszClass = AfxRegisterWndClass(0);
return TRUE;
}
```

O parâmetro que essa função recebe é uma estrutura que nos permite especificar várias características da janela. Os campos dessa estrutura são:

```
typedef struct tagCREATESTRUCT {
LPVOIDlpCreateParams;
HANDLEhInstance;
HMENUhMenu;
HWNDhWndParent;
intcy;//altura da janela
intcx;//largura da janela
inty;//posição y da janela
intx;//posição x da janela
LONGstyle;
LPCSTRlpszName;//nome na barra de título
LPCSTRlpszClass;
DWORDdwExStyle;//20 estilos WS_EX_style
}
```

O campo "lpszClass" é um ponteiro para o nome de uma classe registrada de janelas. Quando fazemos "lpszClass = AfxRegisterWndClass(0);" a classe com as opções "by default" é utilizada. Se desejarmos alterar características como: cor de fundo, cursor do mouse ou ícone da janela, devemos utilizar a função "AfxRegisterWndClass()" com parâmetros específicos (consulte o Help).

Observação: A cor de fundo e o cursor que o usuário verá serão os determinados na criação da janela "CChildView".

Após definidas as características da janela, o construtor da classe base cria a janela principal.

- Criação do Menu, Barra de Status e Barra de Ferramentas

Antes da janela principal tornar-se visível, a estrutura MFC envia para a classe "CMainFrame" a mensagem WM_CREATE que, por sua vez, é tratada pela função "OnCreate()".

O menu, a barra de status e a barra de ferramentas são criados nessa função.

- Criação da janela "CChildView"

No arquivo de cabeçalho da classe "CMainFrame" encontra-se a seguinte declaração:

```
protected:
CChildView m_wndView;
```

Portanto, a janela "CChildView" é representada dentro da classe "CMainFrame" pela variável "m_wndView".

O processo de criação da janela "filha" tem início na função "CMainFrame::OnCreate()", como no código seguinte:

```
m_wndView.Create(NULL, NULL, AFX_WS_DEFAULT_VIEW,
CRect(0, 0, 0, 0),
this, // janela principal
AFX_IDW_PANE_FIRST,
NULL);
```

Note que o tamanho da janela não é importante, porque o tamanho da janela "filha" ajusta-se ao da janela principal.

O "parent" da janela "CChildView" é especificado por "this", que é um ponteiro para o objeto "CMainFrame".

Antes da janela "filha" ser realmente criada, a função "CChildView::PreCreateWindow()" é chamada. Como na criação da janela principal, essa função é utilizada para determinar algumas características da janela. As únicas características importantes são a cor de fundo da janela e o cursor padrão do mouse. As outras características - como título, tamanho e ícone - são determinadas pela janela principal.

- Estabelecimento da Comunicação entre o Usuário e "CChildView"

Observe o código da função abaixo:

```
void CMainFrame::OnSetFocus(CWnd* pOldWnd)
{
// forward focus to the view window
m_wndView.SetFocus();
}
```

Essa função é chamada quando a janela principal recebe o "foco" do sistema operacional. Para que a janela "CChildView" responda à mensagens referentes ao acionamento das teclas do teclado, é necessário passar o "foco" da janela principal para a janela "filha".

As mensagens do sistema operacional são enviadas para a janela principal. Para que essas mensagens cheguem à janela "CChildView", a janela principal deve repassá-las à janela "filha". Isso é feito pela seguinte função:

```
BOOL CMainFrame::OnCmdMsg(UINT nID, int nCode, void* pExtra,
AFX_CMDHANDLERINFO* pHandlerInfo)
{
// let the view have first crack at the command
if (m_wndView.OnCmdMsg(nID, nCode, pExtra, pHandlerInfo))
return TRUE;
// otherwise, do default handling
return CFrameWnd::OnCmdMsg(nID, nCode, pExtra, pHandlerInfo);
}
```

Note que uma mensagem só será tratada pela janela principal se não for tratada pela janela "filha".

- "Message Map" e Tratamento de Mensagens

O tratamento das mensagens recebidas pela janela "CChildView" se dá da mesma forma que em qualquer outra janela: através do "Message Map".

No código do programa gerado na seção 2, a única mensagem tratada explicitamente é a WM_PAINT e, por isso, o mapa de mensagens da classe "CChildView" é:

```
BEGIN_MESSAGE_MAP(CChildView,CWnd )
//{{AFX_MSG_MAP(CChildView)
ON_WM_PAINT()
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

Para tratar outras mensagens basta inserir as linhas correspondentes no "MESSAGE_MAP" e escrever o código das funções relacionadas. Isso pode ser feito de duas maneiras: inserindo linhas abaixo do comentário "//}}AFX_MSG_MAP" ou utilizando o "ClassWizard".

- AssertValid() e Dump()

Essas funções existem por motivos de diagnóstico e checagem de erros.

1. Exercício

Modifique o programa gerado na seção 2 da seguinte forma:

- Faça com que o menu, a barra de status e a barra de ferramentas não apareçam na janela do programa;
- Modifique a cor de fundo da janela e o cursor padrão do mouse;
- Trate as mensagens WM_LBUTTONDOWN e WM_LBUTTONUP para desenhar uma reta entre os pontos em que o usuário pressionar e soltar o botão esquerdo do mouse;

Dicas:

- O menu é criado na função "InitInstance()" e as barras de status e de ferramentas na função "OnCreate()";
- Utilize as funções "::LoadCursor()" e "::CreateSolidBrush()" para alterar os parâmetros passados à função "AfxRegisterWndClass()";
- Lembre-se de chamar a função "Invalidate()" quando quiser que a janela seja atualizada.