

```
1
2 // DesignArk.cpp : Defines the class behaviors for the application.
3 //
4
5 #include "pch.h"
6 #include "framework.h"
7 #include "afxwinappex.h"
8 #include "afxdialogex.h"
9 #include "DesignArk.h"
10 #include "MainFrm.h"
11
12 #include "ChildFrm.h"
13 #include "CTextDocument.h"
14 #include "CTextDocView.h"
15
16 #include "CAboutDlg.h"
17 #include "CSmartColour.h"
18
19 #ifdef _DEBUG
20 #define new DEBUG_NEW
21 #endif
22
23
24 // CDesignArkApp
25
26 BEGIN_MESSAGE_MAP(CDesignArkApp, CWinAppEx)
27     ON_COMMAND(ID_VIEW_SMARTCOLOURWINDOW, &CDesignArkApp::OnSmartColour)
28     ON_COMMAND(ID_HELP_OPENWEBSITE, &CDesignArkApp::OnOpenSite)
29     ON_COMMAND(ID_HELP_VIEWUSERDOCSONLINE, &CDesignArkApp::OnUserdocs)
30     ON_COMMAND(ID_APP_ABOUT, &CDesignArkApp::OnAppAbout)
31
32     // Standard file based document commands
33     ON_COMMAND(ID_FILE_NEW, &CWinAppEx::OnFileNew)
34     ON_COMMAND(ID_FILE_OPEN, &CWinAppEx::OnFileOpen)
35     // Standard print setup command
36     ON_COMMAND(ID_FILE_PRINT_SETUP, &CWinAppEx::OnFilePrintSetup)
37 END_MESSAGE_MAP()
38
39
40 // CDesignArkApp construction
41 CDesignArkApp::CDesignArkApp() noexcept
42 {
43     m_bHiColorIcons = TRUE;
44     m_nAppLook = 0;
45
46     // support Restart Manager
47     m_dwRestartManagerSupportFlags =
48         AFX_RESTART_MANAGER_SUPPORT_ALL_ASPECTS;
49 #ifdef _MANAGED
50     // If the application is built using Common Language Runtime
51     support (/clr):
52     //      1) This additional setting is needed for Restart Manager
53     support to work properly.
54     //      2) In your project, you must add a reference to
```

```
    System.Windows.Forms in order to build.
52     System::Windows::Forms::Application::SetUnhandledExceptionMode
        (System::Windows::Forms::UnhandledExceptionMode::ThrowExceptio
            n);
53 #endif
54
55     SetAppID(_T("DesignArk.Verison.1.0"));
56
57     // TODO: add construction code here,
58     // Place all significant initialization in InitInstance
59     this->smartColour_String.resize(5);
60     this->smartColour_Colour.resize(5);
61
62     this->smartColour_String[0] = std::vector<CString>{
63         L"ELIF",
64         L"ELSE",
65         L"FOR",
66         L"IF",
67         L"IN",
68         L"RETURN",
69         L"WHILE"
70     };
71     this->smartColour_Colour[0] = COLORREF(128, 0, 128);
72
73     this->smartColour_String[1] = std::vector<CString>{
74         L"AS",
75         L"CREATE",
76         L"DECLARE",
77         L"DISPLAY",
78         L"DO",
79         L"FUNCTION",
80         L"INITIALISE",
81         L"INITIALLY",
82         L"PRINT",
83         L"PROCEDURE",
84         L"SET",
85         L"TO",
86     };
87     this->smartColour_Colour[1] = COLORREF(128, 0, 32);
88
89     this->smartColour_String[2] = std::vector<CString>{
90         L"ARRAY",
91         L"BOOLEAN",
92         L"CLASS",
93         L"CONST",
94         L"DICTIONARY",
95         L"FLOAT",
96         L"INT",
97         L"LIST",
98         L"OVERLOAD",
99         L"REAL",
100        L"STRING",
101        L"VIRTUAL",
102        L"VOID"
103    };
104     this->smartColour_Colour[2] = COLORREF(0, 0, 255);
```

```

105
106     this->smartColour_String[3] = std::vector<CString>{
107         L"vector"
108     };
109     this->smartColour_Colour[3] = COLORREF(RGB(0, 255, 0));
110
111     this->smartColour_String[4] = std::vector<CString>{
112         L"RANDOM"
113     };
114     this->smartColour_Colour[4] = COLORREF(RGB(255, 0, 0));
115
116     this->commentType = L"/";
117     this->commentColour = COLORREF(RGB(255, 0, 0));
118
119     this->numberColour = COLORREF(RGB(0, 255, 0));
120     this->highlightColour = COLORREF(RGB(110, 192, 250));
121
122     this->zoom = 1.f;
123     this->sFont = L"Courier New";
124     this->indentSize = 4;
125
126     // Initilise the GDI stuff
127     Gdiplus::GdiplusStartupInput gdiplusStartupInput;
128     ULONG_PTR gdiplusToken;
129
130     Gdiplus::GdiplusStartup(&gdiplusToken, &gdiplusStartupInput,
131         NULL);
132 }
133 CDesignArkApp::~CDesignArkApp()
134 {
135 }
136 // The one and only CDesignArkApp object
137 CDesignArkApp theApp;
138
139 // CDesignArkApp initialisation
140 BOOL CDesignArkApp::InitInstance()
141 {
142     /*/////////////////////////////////////////
143     AllocConsole();
144     FILE* fDummy;
145     freopen_s(&fDummy, "CONOUT$", "w", stdout);
146     freopen_s(&fDummy, "CONOUT$", "w", stderr);
147     freopen_s(&fDummy, "CONIN$", "r", stdin); // Console can be used
148         be debugging
149     //////////////////////////////////////////*/
150
151     // Initialise the common controls
152     INITCOMMONCONTROLSEX InitCtrls;
153     InitCtrls.dwSize = sizeof(InitCtrls);
154     InitCtrls.dwICC = ICC_WIN95_CLASSES;
155     InitCommonControlsEx(&InitCtrls);
156
157     CWinAppEx::InitInstance();
158
159     // Initialize OLE libraries

```

```
159     if (!AfxOleInit())
160     {
161         AfxMessageBox(IDP_OLE_INIT_FAILED);
162         return FALSE;
163     }
164
165     AfxEnableControlContainer();
166     EnableTaskbarInteraction();
167
168     AfxInitRichEdit2(); // is required to use RichEdit control
169
170     // Register app with OS and load INI file options (including MRU)
171     SetRegistryKey(_T("DesignArk"));
172     LoadStdProfileSettings(4); // Load standard INI file options (including MRU)
173
174     // Initialise managers
175     InitContextMenuManager();
176     InitKeyboardManager();
177     InitTooltipManager();
178     CMFCToolTipInfo ttParams;
179     ttParams.m_bVislManagerTheme = TRUE;
180     theApp.GetTooltipManager()->SetTooltipParams(AFX_TOOLTIP_TYPE_ALL,
181         RUNTIME_CLASS(CMFCToolTipCtrl), &ttParams);
182
183     // Initialise document template
184     CMultiDocTemplate* pDocTemplate;
185     pDocTemplate = new CMultiDocTemplate(IDR_DesignArkTYPE,
186         RUNTIME_CLASS(CTextDocument),
187         RUNTIME_CLASS(CChildFrame), // custom MDI child frame
188         RUNTIME_CLASS(CTextDocView));
189     if (!pDocTemplate)
190         return FALSE;
191     this->AddDocTemplate(pDocTemplate);
192
193     // create main MDI Frame window
194     CMainFrame* pMainFrame = new CMainFrame;
195     if (!pMainFrame || !pMainFrame->LoadFrame(IDR_MAINFRAME))
196     {
197         delete pMainFrame;
198         return FALSE;
199     }
200     m_pMainWnd = pMainFrame;
201
202     // Parse command line for standard shell commands, DDE, file open
203     CCommandLineInfo cmdInfo;
204     ParseCommandLine(cmdInfo);
205
206     // Dispatch commands specified on the command line. Will return FALSE if
207     // app was launched with /RegServer, /Register, /Unregserver or /Unregister.
208     if (!ProcessShellCommand(cmdInfo))
```

```

209         return FALSE;
210
211         // The main window has been initialized, so show and update it
212
213         //CRect client;
214         //pMainFrame->GetWindowRect(client);
215         //pMainFrame->MoveWindow(client.left, client.top, 750, 900);
216
217         pMainFrame->ShowWindow(m_nCmdShow);
218         pMainFrame->UpdateWindow();
219
220         return TRUE;
221     }
222     int CDesignArkApp::ExitInstance()
223     {
224         //TODO: handle additional resources you may have added
225         AfxOleTerm(FALSE);
226
227         return CWinAppEx::ExitInstance();
228     }
229
230     // Public Commands
231     void CDesignArkApp::OnSmartColour()
232     {
233         CString tempIndentSize;
234         tempIndentSize.Format(L"%d", this->indentSize);
235         CSmartColour smartColour(this->smartColour_String, this-      ↗
            >smartColour_Colour, this->commentType, this->commentColour,      ↗
            this->numberColour, this->highlightColour, this->sFont,      ↗
            tempIndentSize);
236
237         if (smartColour.DoModal() == IDOK) {
238
239             this->smartColour_String[0] = ((CSmartColourStatements*)      ↗
                smartColour.GetPage(0))->m_Strings_Statements_syntax_var;
240             this->smartColour_Colour[0] = ((CSmartColourStatements*)      ↗
                smartColour.GetPage(0))->m_Colour_Statements_syntax_var;
241
242             this->smartColour_String[1] = ((CSmartColourStatements*)      ↗
                smartColour.GetPage(0))-      ↗
                >m_Strings_Statements_statements_var;
243             this->smartColour_Colour[1] = ((CSmartColourStatements*)      ↗
                smartColour.GetPage(0))-      ↗
                >m_Colour_Statements_statements_var;
244
245             this->smartColour_String[2] = ((CSmartColourTypes*)      ↗
                smartColour.GetPage(1))->m_listbox_types_strings;
246             this->smartColour_Colour[2] = ((CSmartColourTypes*)      ↗
                smartColour.GetPage(1))->m_Combo_Types_colour_var;
247
248             this->smartColour_String[3] = ((CSmartColourBuiltin*)      ↗
                smartColour.GetPage(2))->m_Listbox_Builtin_Classes_var;
249             this->smartColour_Colour[3] = ((CSmartColourBuiltin*)      ↗
                smartColour.GetPage(2))-      ↗
                >m_Combo_Builtin_Classes_colour_var;
250

```

```

...mes\source\repos\DesignArk\DesignArk\DesignArk.cpp 6
251     this->smartColour_String[4] = ((CSmartColourBuiltin*)  ↗
        smartColour.GetPage(2))->m_Listbox_Builtin_Functions_var;
252     this->smartColour_Colour[4] = ((CSmartColourBuiltin*)  ↗
        smartColour.GetPage(2))-  ↗
        >m_Combo_Builtin_Functions_colour_var;
253
254     this->commentType = ((CSmartColourOther*) smartColour.GetPage  ↗
        (3))->m_pCombo_comments_type_var;
255     this->commentColour = ((CSmartColourOther*)  ↗
        smartColour.GetPage(3))->m_pCombo_comments_colour_var;
256
257     this->numberColour = ((CSmartColourOther*)  ↗
        smartColour.GetPage(3))->m_pCombo_numbers_colour_var;
258
259     this->highlightColour = ((CSmartColourOther*)  ↗
        smartColour.GetPage(3))->m_pCombo_highlight_colour_var;
260
261     this->sFont = ((CSmartColourOther*) smartColour.GetPage(3))-  ↗
        >m_pCombo_fonts_var;
262
263     this->indentSize = _wtoi(((CSmartColourOther*)  ↗
        smartColour.GetPage(3))->m_pCombo_indent_var);
264
265     // Will need to fixed if more views/docs are implemeted but  ↗
        will do for v1.0
266     if (this->m_ActiveView) {
267         ((CTextDocView*) this->m_ActiveView)->refresh();
268     }
269 }
270 }
271
272 void CDesignArkApp::OnAppAbout() // This bit is not working
273 {
274     CAboutDlg aboutDlg;
275     aboutDlg.DoModal(); // This line is triggering a debug assertion  ↗
        failure
276 }
277 void CDesignArkApp::OnOpenSite()
278 {
279     ShellExecute(0, 0, L"http://localhost/DesignArk/", 0, 0,  ↗
        SW_SHOW);
280 }
281 void CDesignArkApp::OnUserdocs()
282 {
283     ShellExecute(0, L"open", L"http://localhost/DesignArk/  ↗
        userdocs.php", 0, 0, SW_SHOW);
284 }
285
286 // Protected Implementation
287 void CDesignArkApp::PreLoadState()
288 {
289     BOOL bNameValid;
290     CString strName;
291     bNameValid = strName.LoadString(IDS_EDIT_MENU);
292     ASSERT(bNameValid);
293     GetContextMenuManager()->AddMenu(strName, IDR_POPUP_EDIT);

```

```
294     bNameValid = strName.LoadString(IDS_EXPLORER);
295     ASSERT(bNameValid);
296     GetContextMenuManager()->AddMenu(strName, IDR_POPUP_EXPLORER);
297 }
298 void CDesignArkApp::LoadCustomState()
299 {
300 }
301 void CDesignArkApp::SaveCustomState()
302 {
303 }
304
305
```