



编程爱好者

www.cppfans.com



思科最新协作AI产品免费体验两个

立即查看

主页 -> C++ Builder 资料 -> C++ Builder 编程技巧 -> 界面处理 -> 使用 Action 控件

- C++ Builder 串口控件
- C++ Builder 编程技巧
- 字符串及文字处理
- 多媒体处理
- 图片处理
- 文件处理
- 界面处理
 - 使用 Action 控件
 - 其他程序文字拖拽到Memo里
 - 把文件拖拽到窗口里面
 - 枚举控件的属性、事件、成员函数和变量
- C++ Builder 操作指南
- C++ Builder 参考手册
- 网友留言/技术支持

使用 Action 控件

- 使用标准 Action 控件实现标准功能
例：制作一个简单的记事本
- 使用自定义 Action 控件
例：“制作一个简单的记事本”基础上增加“保存”和“新建”功能

使用标准 Action 控件实现标准功能

- 在 Form 上放一个 TActionList 控件 ActionList1 和一个 TImageList 控件 ImageList1
- ActionList1 的 Images 属性关联到 ImageList1
- 双击 ActionList1 打开编辑对话框，在打开的对话框里面，点击鼠标右键，选择 New Standard Action...
- 在打开的对话框里面，是可用的标准 Action 控件的列表。
- 选择需要的标准功能，点击 OK 按钮添加到 ActionList1 里面，这些标准功能的图标也会随着加入关联的 ImageList1
- 菜单项、工具条按钮、或者其他的控件，都可以通过 Action 属性选择加入 ActionList1 里面的标准功能了。

请参考 CHM 帮助文件处理里面的“显示帮助内容和帮助索引”章节的内容

例：制作一个简单的记事本

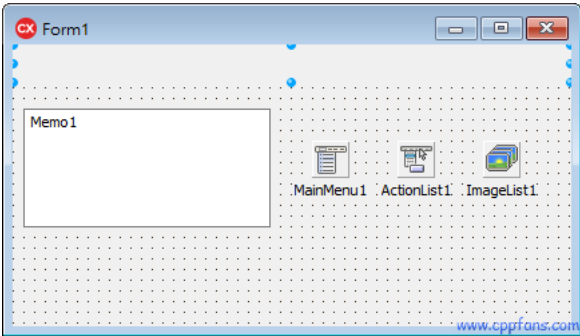
1. 新建一个 VCL 项目：选择菜单 File -> New -> VCL Forms Application - C++ Builder

新建的项目放在一个文件夹里面是一个好习惯：

- 可以点击工具条上的 Save All 按钮，
 - 或者选择菜单 File -> Save All，
 - 也可以通过快捷键 Shift + Ctrl + S，
- 保存所有的文件到一个新建的文件夹里面

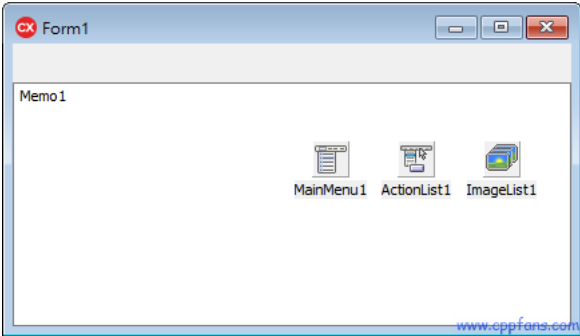


2. 在 Form1 上放一个 TMainMenu 控件、一个 TToolBar 控件、一个 TMemo 控件、一个 TActionList 控件和一个 TImageList 控件：



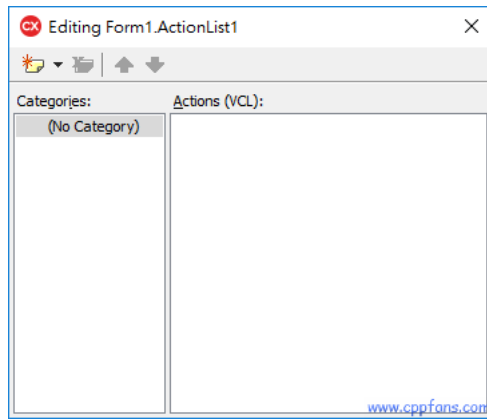
3. 控件关联和调整控件的位置

- 主菜单 MainMenu1 的 Images 属性关联到 ImageList1
- Form1 的 Menu 属性关联到 MainMenu1，一般会自动关联的，现在看不见主菜单是因为菜单是空的，没有菜单项
- 工具条 ToolBar1 的 Images 属性关联到 ImageList1
- 编辑框 Memo1 的 Align 属性选择 alClient
- ActionList1 的 Images 属性关联到 ImageList1

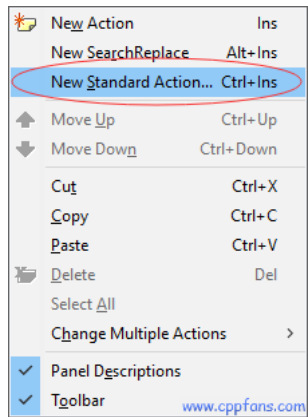


4. 加入标准 Action 控件

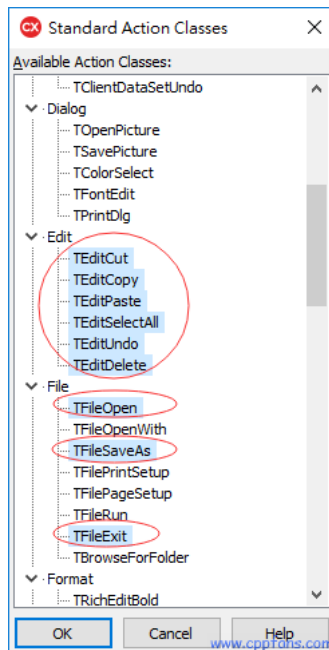
双击 ActionList1 控件，打开编辑对话框，左边是分类，右边是 Action 列表：



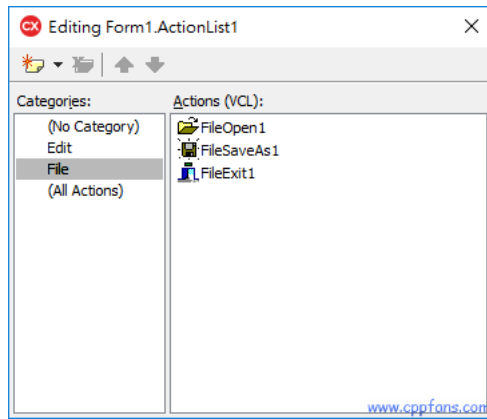
用鼠标右键点击 Action 列表, 在弹出的菜单里面选择 “New Standard Action...”



打开 Standard Action Classes 对话框:



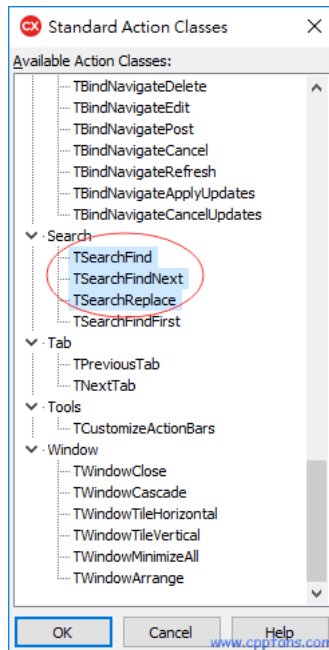
选择 Edit 里面的 TEditCut、TEditCopy、TEditPaste、TEditSelectAll、TEditUndo、TEditDelete;
选择 File 里面的 TFileOpen、TFileSaveAs、TFileExit
点击 OK 按钮, 把这些标准 Action 加入 Actions 列表, 如下图所示:



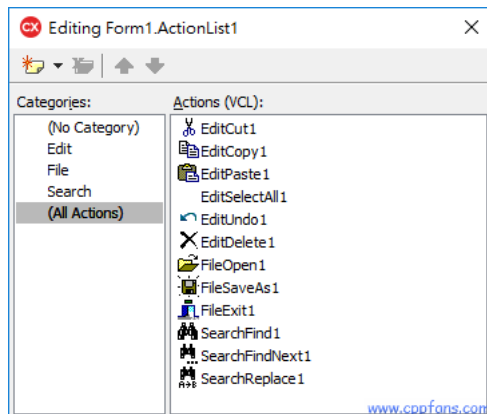
对话框里面：左边是分类，右边是每个分类里面的 Action 列表。

“(No Category)” 是没有分类的 Action, “(All Actions)” 是所有分类的 Action 都显示出来。

如果需要查找替换功能，可以继续添加，用鼠标右键点击 Action 列表，选择 “New Standard Action...”：



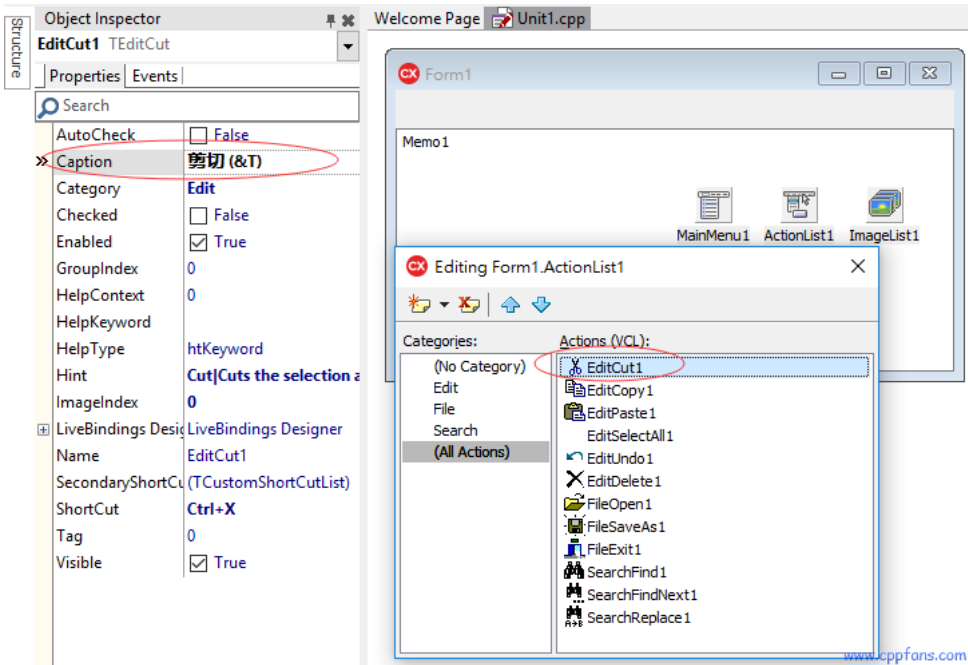
选择 Search 里面的 TSearchFind、TSearchFindNext、TSearchReplace，点击 OK 按钮，添加到 Action 列表：



上面截图是所有分类 (All Actions) 的 Action 列表，所有添加进来的 Action 都在这里。

5. 修改每个 Action 的标题

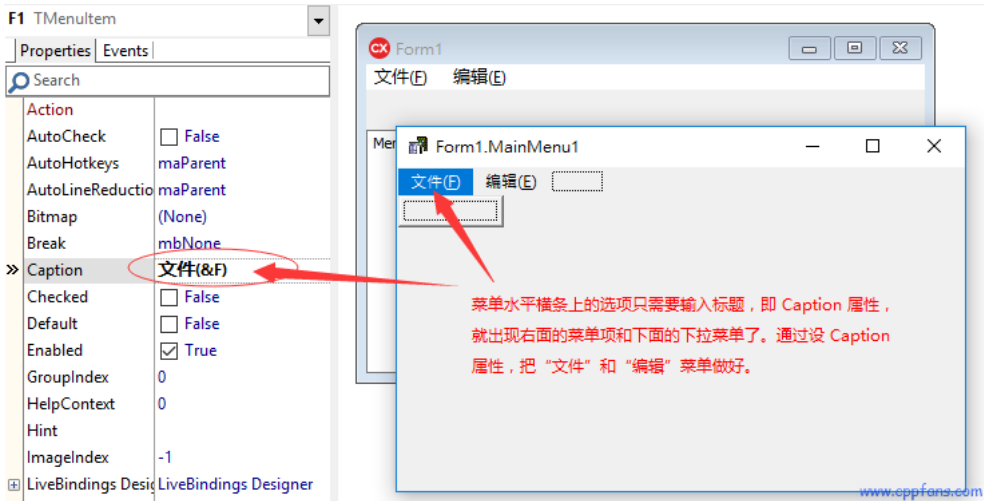
如果让界面以汉语显示，需要修改每个 Action 的标题，即 Caption 属性：



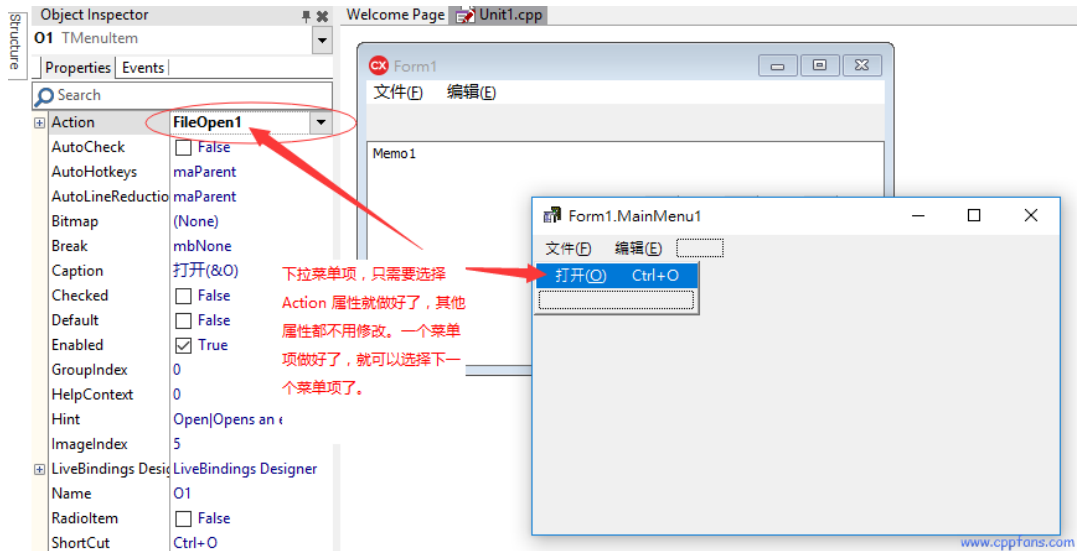
标准 Action 控件	标题	备注, 无备注表示这个 Action 是完整功能
EditCut1	剪切 (&T)	
EditCopy1	复制 (&C)	
EditPaste1	粘贴 (&P)	
EditSelectAll1	全选 (&A)	
EditUndo1	撤销 (&U)	
EditDelete1	删除 (&D)	(Shortcut 属性会影响正常的编辑时的 Del 键功能, 清空 Shortcut 属性可以解决)
FileOpen1	打开 (&O)	需要实现 OnAccept 事件
FileSaveAs1	另存为 (&A)	需要实现 OnAccept 事件
FileExit1	退出 (&X)	
SearchFind1	查找 (&F)	
SearchFindNext1	下一个 (&N)	需要把 SearchFind 属性关联到 SearchFind1 控件来实现功能
SearchReplace1	替换 (&R)	

6. 设计主菜单

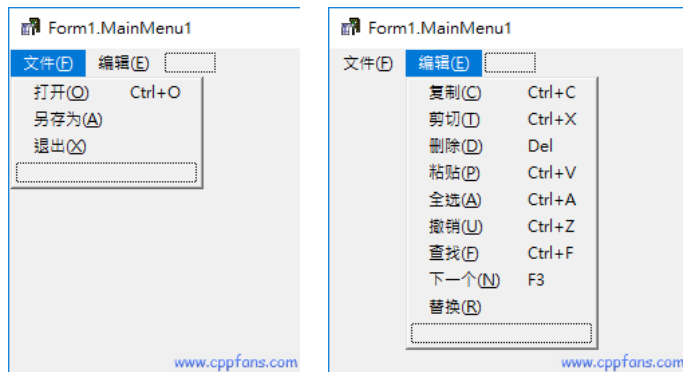
双击主菜单控件 MainMenu1, 打开编辑主菜单的画面。
主菜单水平横条上的项目只需要输入标题, 即 Caption 属性就可以了, “文件” 和 “编辑” 两个菜单项的 Caption 属性设好, 如下图所示:



主菜单的每个下拉菜单的项目, 只需要关联 Action 属性到 ActionList1 里面的, 刚刚加入的那些 Action, 就完成菜单项的设置了, 菜单项和 Action 的属性都是同步的, 可以显示出来菜单项的标题为每个 Action 的标题。
在菜单设计画面里面, 没有看到每个 Action 关联的小图标, 这不影响运行的效果, 在运行的时候是可以看见小图标的。

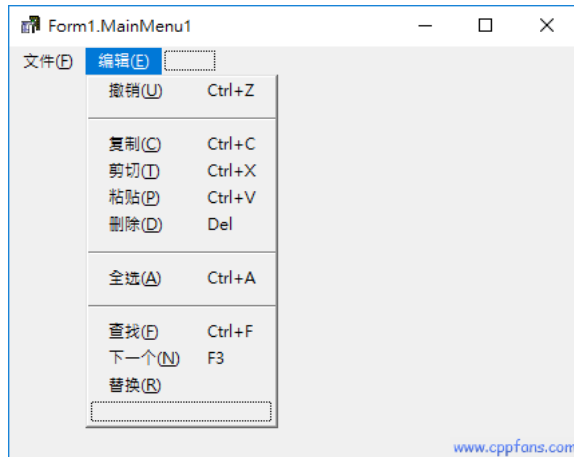


把所有的菜单项都关联到 Action 控件，如下图所示：



菜单项调整顺序：用鼠标左键按住并且拖动菜单项，可以调整菜单项的顺序。

加入分隔线：菜单项的标题只有一个减号“-”的项目是分隔线，新的空白菜单项的标题输入减号，并且调整位置，如下图所示：

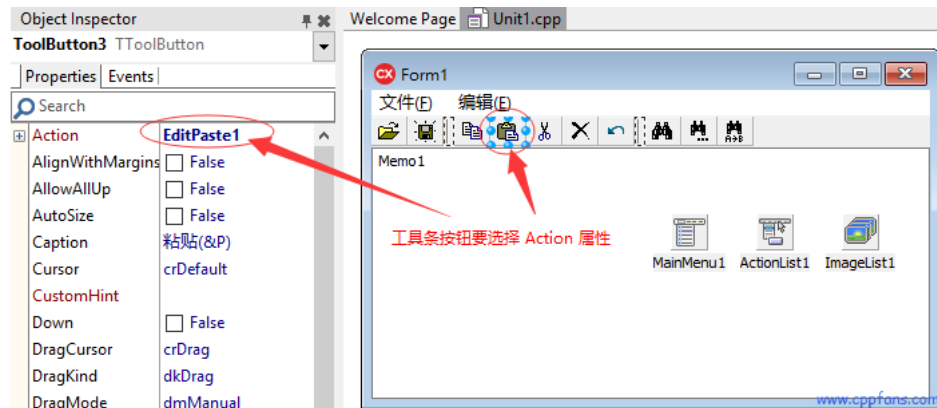


7. 设计工具条

在工具条 ToolBar1 上点击鼠标右键，选择“New Button”可以新增一个按钮，选择“New Separator”可以新增一个分隔线。

新增的按钮，即 ToolButton 的 Action 属性，选择希望点击这个按钮实现的功能，就完成了按钮的设置工作。

把所有希望放在工具条上的功能都对应的添加到工具条按钮上，选择 Action 属性，工具条的设计就完成了，如下图所示：

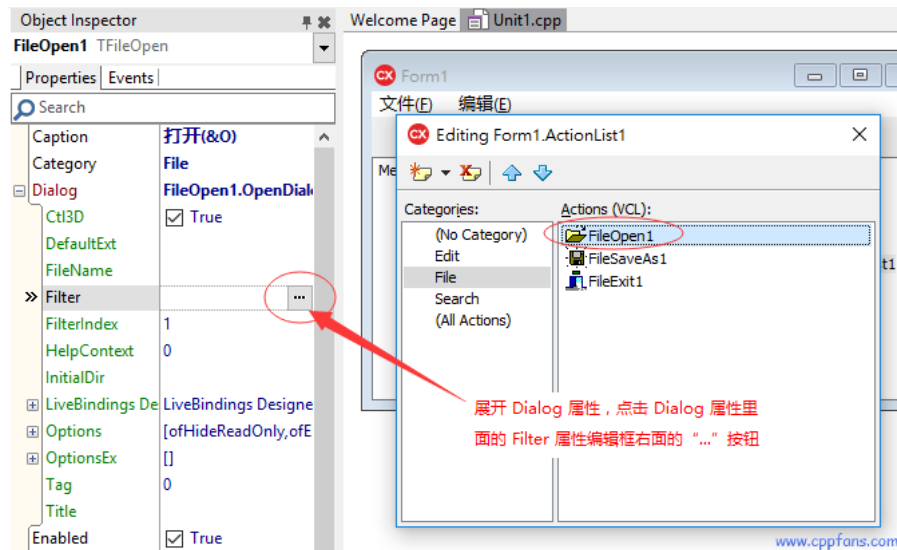


8. 完成功能

通过前面添加的**标准 Action 表格**可以看到: FileOpen1、FileSaveAs1、SearchFindNext1 这三个 Action 需要处理。

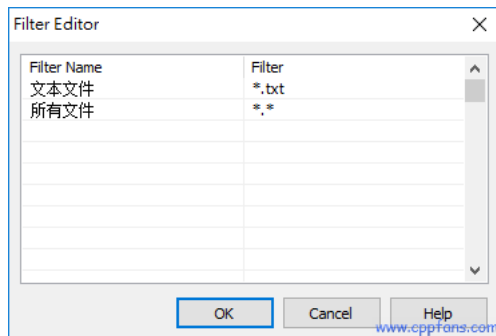
FileOpen1:

双击 ActionList1 控件, 打开 Action 编辑对话框, 找到并且点选 FileOpen1,
先选择文件类型过滤: 如果希望在打开文件对话框里面, 只显示文本文件的列表, 可以如下图方法处理:

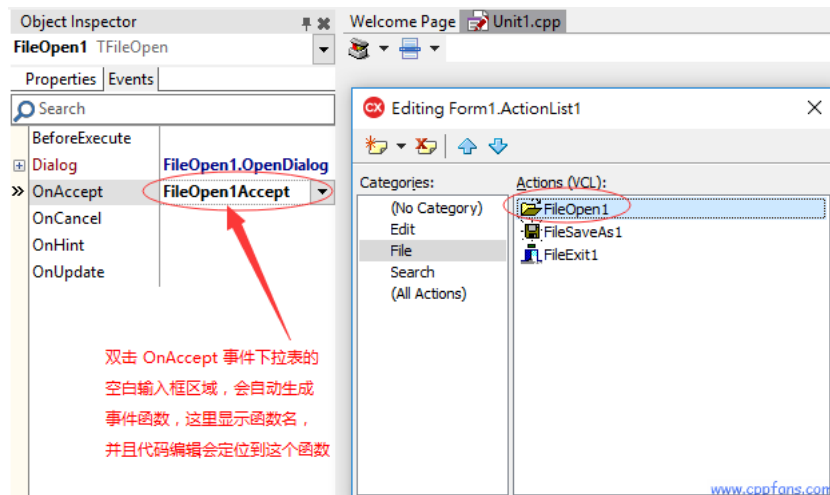


可以在 Filter Editor 里面输入多项内容, 在打开对话框里面以下拉表形式选择。

Filter Name 可以随意输入, 例如 "文本文件", Filter 为 "*.txt" 就是过滤之后, 只显示后缀为 *.txt 的文本文件;
Filter Name 可以随意输入, 例如 "所有文件", Filter 为 "*" 就是不过滤文件, 显示所有文件。



双击 FileOpen1 事件页面里面的 OnAccept 事件下拉编辑框里面的空白区域, 会自动生成函数, 在这里显示出来函数名, 并且在**代码编辑**里面定位到这个函数。以后在事件页面里面的函数名上双击, 都可以定位到**代码编辑**里面的这个函数位置。

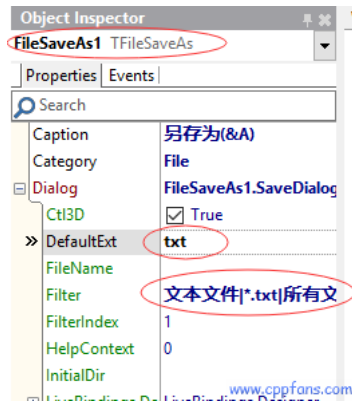


在**代码编辑**里面生成的函数里面, 添加打开文件的代码, 完成之后是这样的:

```
void __fastcall TForm1::FileOpen1Accept(TObject *Sender)
{
    Memo1->Lines->LoadFromFile(FileOpen1->Dialog->FileName);
}
```

FileSaveAs1:

和 FileOpen1 一样, 需要设 Dialog 里面的 Filter 属性, 按照前面 FileOpen1 叙述的做就可以了,
另外 FileSaveAs1 需要设置 Dialog 里面的 DefaultExt 属性, 如果用户没有输入文件名后缀, 用这个默认的后缀, 在这里输入 "txt":



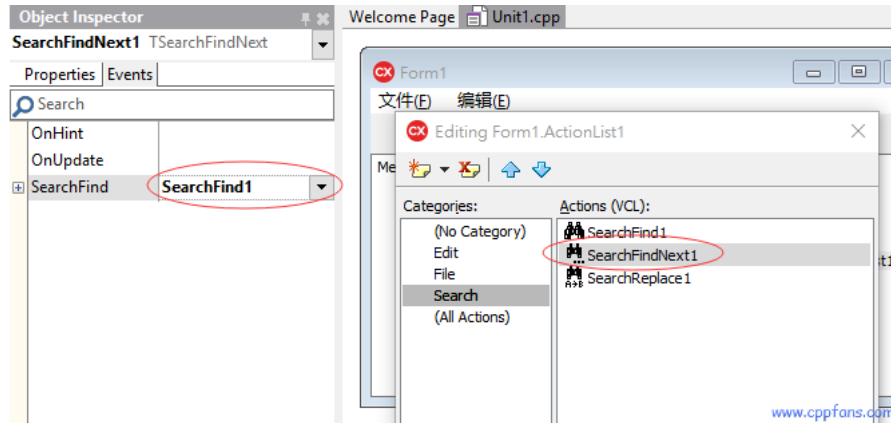
双击 FileSaveAs1 事件页面里面的 OnAccept 事件下拉表编辑框里面的空白区域，自动生成函数，和 FileOpen1 的方法一样，生成的 OnAccept 事件代码完成之后是这样的：

```
void __fastcall TForm1::FileSaveAs1Accept(TObject *Sender)
{
    Memo1->Lines->SaveToFile(FileSaveAs1->Dialog->FileName);
}
```


SearchFindNext1:

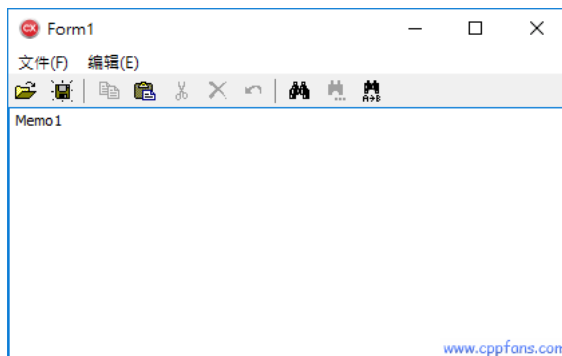
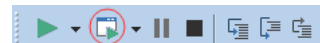
SearchFindNext1 “查找下一个” 控件要和 “查找” 控件 SearchFind1 关联：

双击 ActionList1，在打开的 Action 列表里面找到并且点选 SearchFindNext1，在 SearchFind 属性的下拉表里面，找到并且选择 SearchFind1，如下图所示：



9. 编译和运行

点击工具条上的运行按钮 “”，或者使用键盘快捷键 F9 进行以调试方式运行程序。虽然我们只写了两句代码，可是程序所有的功能都是可用的！下面是程序运行的截图：



这个程序缺少“保存”和“新建”两个项目，请继续阅读“使用自定义 Action 控件”。

参考链接：[创建 VCL 项目](#) • [生成独立运行的 exe 文件](#) • [运行 exe 项目](#) • [调试 exe 项目](#) • [选择显示主题](#)

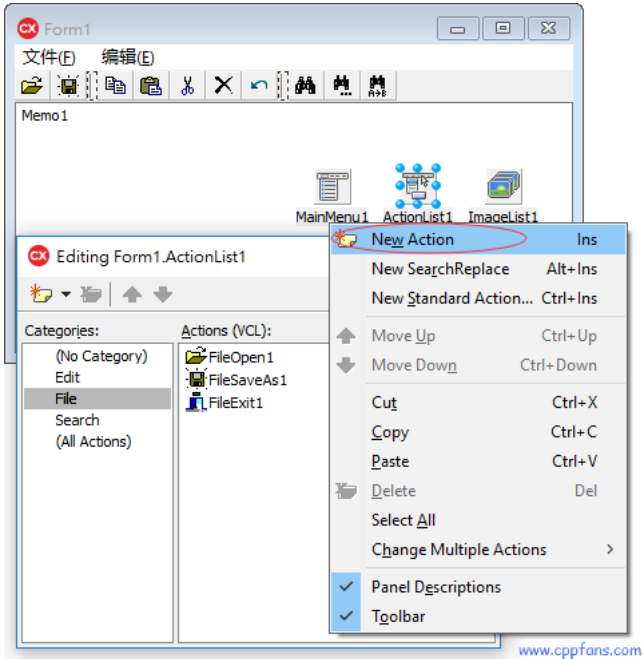
使用自定义 Action 控件

1. 在 Form 上放一个 TActionList 控件 ActionList1 和一个 TImageList 控件 ImageList1
2. ActionList1 的 Images 属性关联到 ImageList1
3. 双击 ActionList1 打开编辑对话框，在打开的对话框里面，点击鼠标右键，选择 New Action...
4. 可以重复第三步添加多个 Action 控件
5. ActionList 里面的每个 Action 控件都可以设定标题、Enabled 和 Visible 属性，或者通过 ShortCut 属性设定快捷键，实现 OnExecute 事件等
6. 菜单项、工具条按钮、或者其他的控件，都可以通过 Action 属性选择加入 ActionList1 里面的 Action 了，关联到 Action 的控件的属性和事件都和关联的 Action 同步

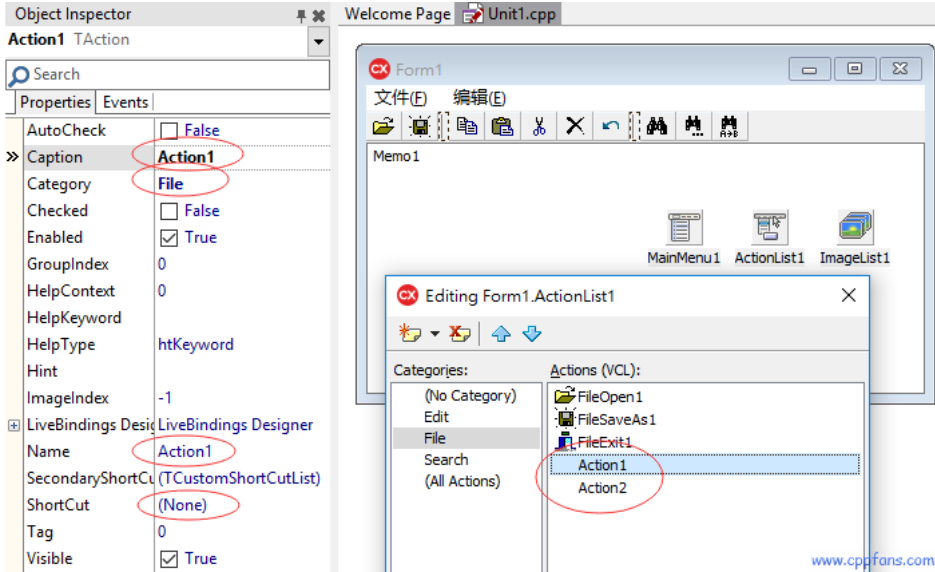
例：“制作一个简单的记事本”基础上增加“保存”和“新建”功能

在 “使用标准 Action 控件实现标准功能” 的例子: “制作一个简单的记事本” 基础上, 增加 “保存” 和 “新建” 功能

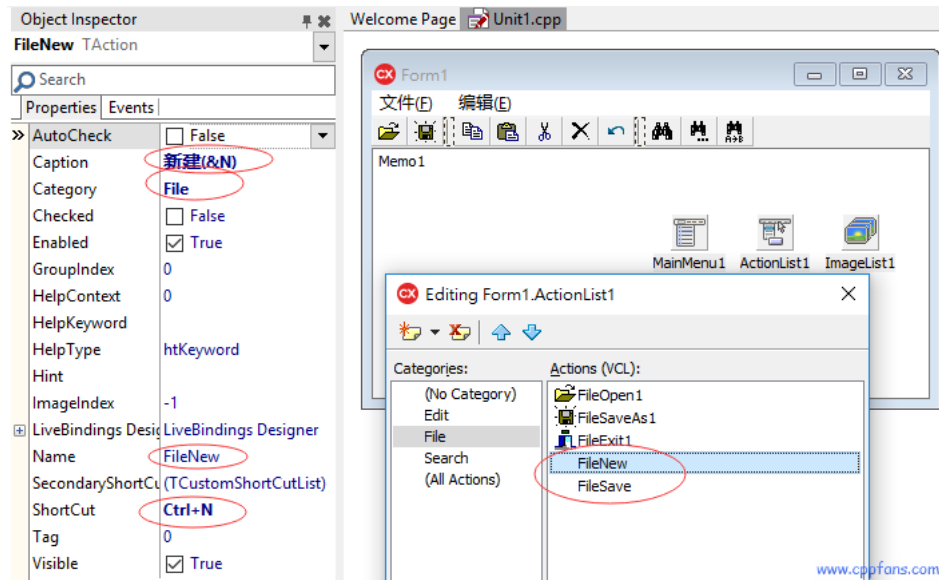
1. 添加 “保存” 和 “新建” 两个 Action:
- 双击 “制作一个简单的记事本” 例子 Form1 上的 ActionList1 控件, 打开编辑对话框, 我们希望在 File 分类里面添加这两个 Action, 就先点选 File 分类, 如果选错了分类, 以后也可以调整。在 Actions 里面点击鼠标右键, 选择 “New Action” 添加自定义的 Action 控件。



添加两个 Action 之后的截图:

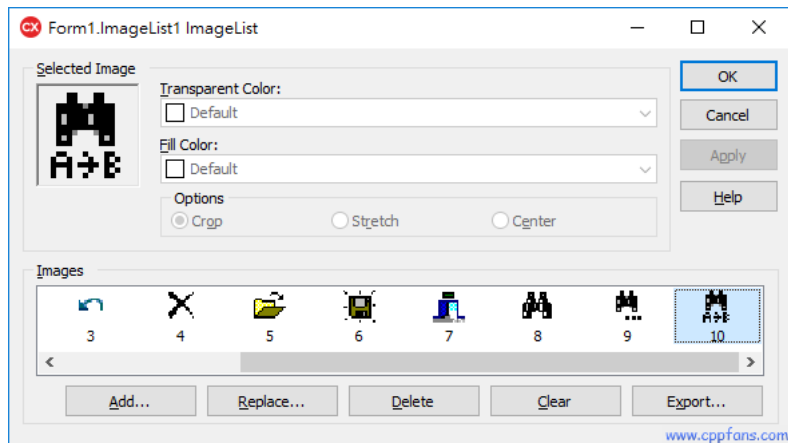


- 新增的 Action 需要修改的项目如上面截图所示:
- Caption: 标题, 例如 “保存(&S)”、“新建(&N)”;
- Category: 分类, 可以下拉表在已有分类里面选择, 也可以直接输入一个新的分类名称;
- Shortcut: 快捷键, 例如保存的快捷键为 “Ctrl+S”、新建的快捷键为 “Ctrl+N”。可以在下拉表里面选择, 也可以直接输入新的快捷键。
- Name: 控件的名称, 默认的 Action1、Action2 名称不容易区分, 可以改成 FileSave 和 FileNew 或者其他自己容易记忆和区分的名字。
- 修改之后的截图:

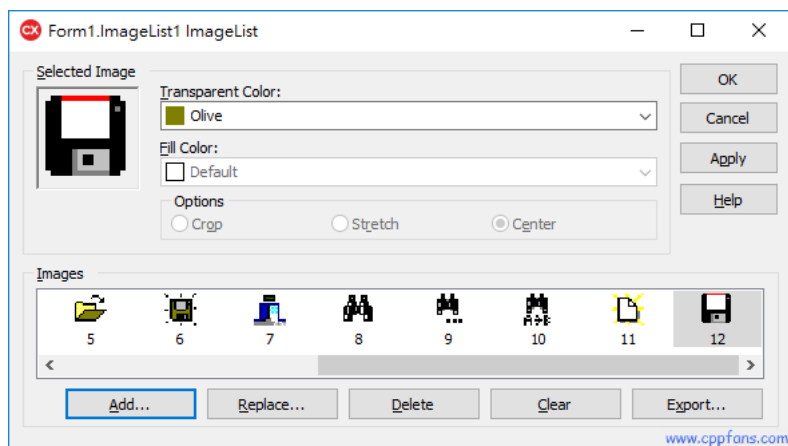


2. 给 Action 添加图标

双击 ImageList1 控件, 打开编辑画面:

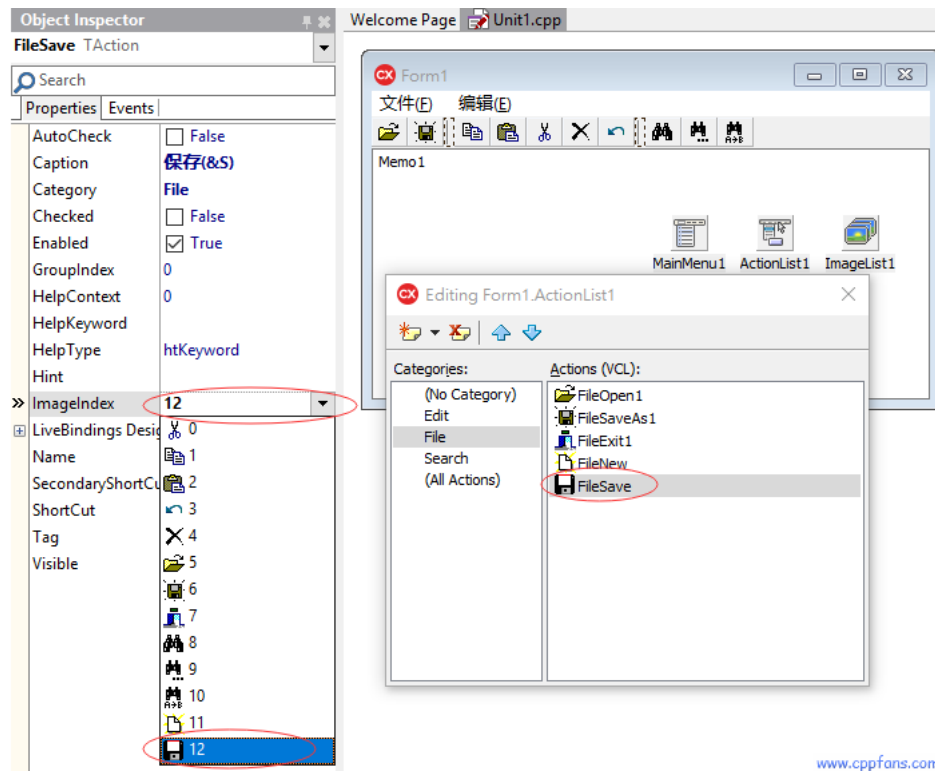


由于前面的小图片都关联到其他控件了, 是根据顺序号关联的, 所以不能改变这些小图片的顺序, 只能在最后添加新的小图片: 点选最后一个小图片, 如上面的截图, 然后点击 “Add...” 按钮, 选择要加入的图片, 添加完成的截图如下:



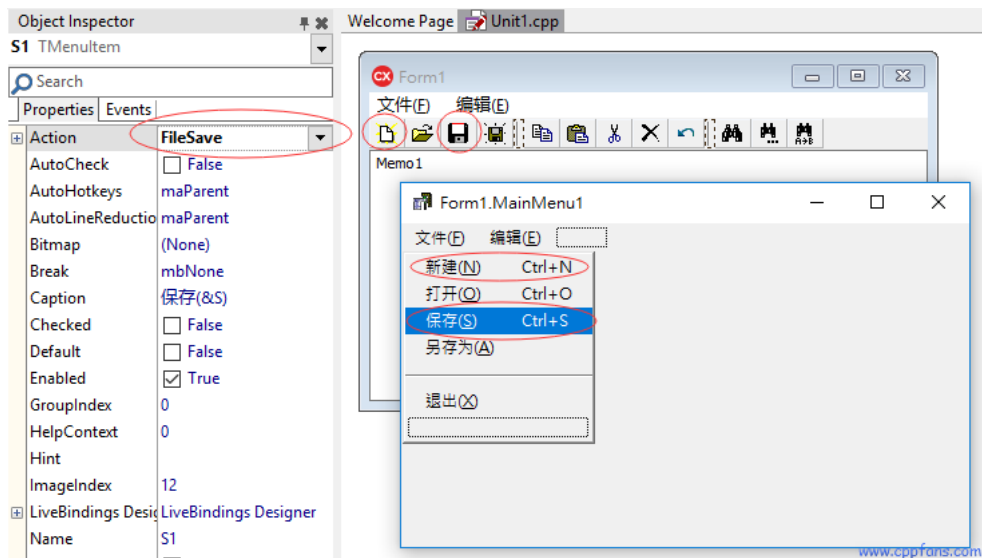
第 11 个和第 12 个小图片就新加入的, 在所有其他图片的最后。如果找不合适的小图片, 可以把这两个另存下来使用: “📁”, “💾”。如果顺序错了, 可以用鼠标拖拽调整, 也可以点击 “Cancel” 按钮取消操作, 然后重新添加。如果检查没有问题了, 点击 “OK” 按钮确定修改生效。

双击 ActionList1 控件, 打开 Action 列表, 选择要加入或修改图标的 Action, FileNew 的 ImageIndex 属性: 下拉表选择刚加入的第 11 个小图片, FileSave 的 ImageIndex 属性: 下拉表选择更加入的第 12 个小图片, 如下图所示:



3. 添加到菜单和工具条上

和“制作一个简单的记事本”例子的第 6 步“设计主菜单”和第 7 步“设计工具条”一样，新增加两个菜单项和两个工具条按钮，把菜单项和工具条按钮的 Action 属性分别关联到 FileNew 和 FileSave 这两个 Action 就可以了



4. 实现功能

保存:

如果要实现保存功能，必须记住最后一次操作“打开”或“另存为”选择的文件名，点击“保存”的时候使用记住的这个文件名。

如果记住的文件名为空，即最开始运行，或者新建之后的状态，点击“保存”需要调用“另存为”，是为了让用户起文件名和选择存放位置。

在代码编辑里面，翻到 Unit1.h 头文件页面，在 private: 里面添加保存文件名的字符串和显示文件名的函数：

```
private: // User declarations
    UnicodeString sFileName;
    void ShowFileName(void);
```

在代码编辑里面，翻到 Unit1.cpp 文件页面，

在 TForm1 的构造函数里面添加显示文件名，和显示文件名的函数，还要加上记住文件名的代码：

```
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Memo1->Lines->Clear(); // 清空编辑内容
    ShowFileName();        // 显示文件名
}
//-----
void TForm1::ShowFileName(void)
{
    if(sFileName.IsEmpty()) // 文件名为空
```

```

    Caption = L"未命名 - 记事本";
else
    Caption = sFileName + L" - 记事本";
}
//-----
void __fastcall TForm1::FileOpen1Accept(TObject *Sender)
{
    sFileName = FileOpen1->Dialog->FileName; // 记住打开的文件名
    Memo1->Lines->LoadFromFile(sFileName);    // 打开文件
    ShowFileName();                          // 显示文件名
}
//-----
void __fastcall TForm1::FileSaveAs1Accept(TObject *Sender)
{
    sFileName = FileSaveAs1->Dialog->FileName; // 记住保存的文件名
    Memo1->Lines->SaveToFile(sFileName);        // 保存文件
    ShowFileName();                          // 显示文件名
}
//-----

```

双击 ActionList1 打开编辑画面，找到并且点选 FileSave，在事件页面里面，OnExecute 事件右面的空白位置，生成事件函数，这里显示函数名，并且定位到代码编辑的生成的函数的位置。以后再双击函数名，都会定位到代码编辑的这个函数的位置。

```

void __fastcall TForm1::FileNewExecute(TObject *Sender)
{
    sFileName = L""; // 清空文件名
    Memo1->Lines->Clear(); // 清空编辑内容
    ShowFileName();    // 显示文件名
}

```

新建：


双击 ActionList1 打开编辑画面，找到并且点选 FileNew，在事件页面里面，双击 OnExecute 事件右面的空白位置，生成事件函数，这里显示函数名，并且定位到代码编辑的生成的函数的位置。以后再双击函数名，都会定位到代码编辑的这个函数的位置。

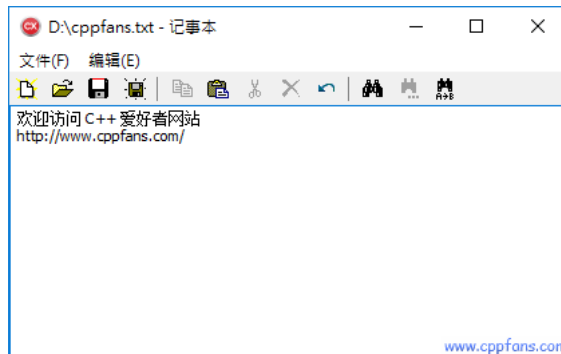
```

void __fastcall TForm1::FileNewExecute(TObject *Sender)
{
    sFileName = L""; // 清空文件名
    Memo1->Lines->Clear(); // 清空编辑内容
    ShowFileName();    // 显示文件名
}

```

5. 编译和运行

点击工具条上的运行按钮 “”，或者使用键盘快捷键 F9 进行以调试方式运行程序。



参考链接：[创建 VCL 项目](#) • [生成独立运行的 exe 文件](#) • [运行 exe 项目](#) • [调试 exe 项目](#) • [选择显示主题](#)

下一页：[其他程序文字拖拽到Memo里](#)