

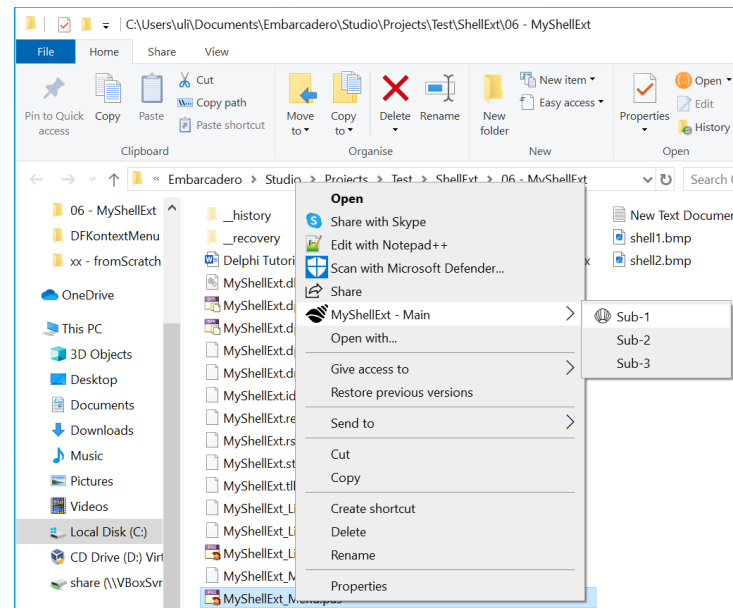
Tutorial- Delphi- Shell Extension- Context Menu- Part 3 (sub-menus)

Scope

Part 3 builds on part 1 of this tutorial and allows to structure the menu with sub-menus and icons to visualise the functionality of your menus. At the end of this part 3 of the tutorial you will be able to sub-menus and add icons to the menu items.

Make Context Menu with Sub-Menus and Icon

The current context menu is flat. Although we can add as many menu items as we wish (almost), we should avoid adding more than 2 or 3 items in the main context. Adding sub-structure and icons provides a more effective use for the user, which is the whole reason why we do a context menu. In this section we create the shown menu structure with a main entry which links to 3 sub menu items and visual icons for Main and Sub-1.



Build the Menu Structure

Remember the context menu is build by the function *QueryContextMenu*. So we need to adapt the code a bit to add substructure with a Main entry with 3 Sub-menu items. First we need to create the sub-menu items by creating an empty menu, *CreatePupUpMenu*, and add then our items with *AppendMenu*. The handle *hMnu* is the link to this sub-structure menu with the 3 items. Then we insert the main menu item (MyShellExt – Main) to the context menu, define it as a *MF_POPUP*, and link to our created sub-menu structure with its handle *hMnu*. At the end we need to provide as Result the number of menu items created, so 4 in our case here.

```
function TMyShellExt.QueryContextMenu(Menu: HMENU; indexMenu, idCmdFirst, idCmdLast, uFlags: UINT): HRESULT; stdcall;
var
    hMnu: HMENU;
begin
    // If the flags include CMF_DEFAULTONLY then we shouldn't do anything
    if (uFlags and CMF_DEFAULTONLY) = CMF_DEFAULTONLY then Result := 0
    else begin
        // create sub-menu
        hMnu := CreatePopupMenu(); // create sub-menu structure
        AppendMenu(hMnu, MF_STRING, idCmdFirst+1, 'Sub-1'); // 1st sub-menu
        AppendMenu(hMnu, MF_STRING, idCmdFirst+2, 'Sub-2'); // 2nd sub-menu
        AppendMenu(hMnu, MF_STRING, idCmdFirst+3, 'Sub-3'); // 3rd sub-menu
        InsertMenu(Menu, indexMenu, // insert menu with the sub-menus
            MF_STRING or MF_BYPOSITION or MF_POPUP, hMnu, 'MyShellExt - Main');
        // Return number of menu items added
        Result := 4;
    end;
end;
```

Propose to test the code and ensure the display of the menu and sub-menus works as expected, before adding the icons.

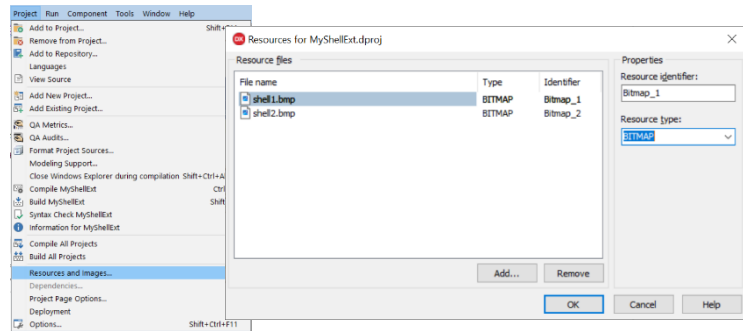
Add Icons to the Menu Items

In this example project we add 2 bitmaps as icons to the Main and Sub-1 menu items. For that we need a bitmap with the size of maximum 32x32 pixels. We add those as resources to our Delphi project, link them to two TBitmap variables and then link them to the red marked code above to display them with the menu items.

GSCL//Unclassified//

Insert resources to the DLL

So first you need 2 bitmap files you want to use for the menu icons. Save them with the size of 32 x 32 pixels in your project folder. In Delphi under "Project -> Resources and Images ..." you can add the 2 bitmaps. By default Delphi will give them the identifiers Bitmap_1 and Bitmap_2. Leave the defaults.



Load the Bitmaps during Runtime

Under var add the 2 TBitmap variables, bmp1 and bmp2.

```
var
  fFiles: TStringList;
  bmp1, bmp2: TBitmap;
```

Under *initialization* create the bitmap structures for bmp1 and bmp2 and load the 2 bitmaps from the resources.

```
initialization
  TMyShellExt_Factory.Create(ComServer, TMyShellExt, Class_MyShellExt,
  'MyShellExt', 'testing', ciMultiInstance, tmApartment);
  fFiles := TStringList.Create;
  bmp1 := TBitmap.Create; bmp1.LoadFromResourceName(hInstance, 'Bitmap_1');
  bmp2 := TBitmap.Create; bmp2.LoadFromResourceName(hInstance, 'Bitmap_2');
finalization
  fFiles.Free;
  bmp1.Free;
  bmp2.Free;
end.
```

Under finalization free the bitmap structures.

Assign the Bitmaps to the Menu Items

Finally we can enable the red highlighted code in the QueryContextMenu function. After ensuring the bmp has been created we can assign it to the menu items. Bmp2 is assigned to the sub-menu item Sub-1, and bmp1 to the main-menu item.

```
function TMyShellExt.QueryContextMenu(Menu: HMENU; indexMenu, idCmdFirst, idCmdLast,
uFlags: UINT): HRESULT; stdcall;
var
  hMnu: HMENU;
begin
  // If the flags include CMF_DEFAULTONLY then we shouldn't do anything
  if (uFlags and CMF_DEFAULTONLY) = CMF_DEFAULTONLY then Result := 0
  else begin
    // create sub-menu
    hMnu := CreatePopupMenu(); // create sub-menu structure
    AppendMenu(hMnu, MF_STRING, idCmdFirst+1, 'Sub-1'); // 1st sub-menu
    if bmp2.Handle <> 0 then // icon for Sub-1
      SetMenuItemBitmaps(hMnu, idCmdFirst+1, MF_BYCOMMAND, bmp2.Handle, bmp2.Handle);
    AppendMenu(hMnu, MF_STRING, idCmdFirst+2, 'Sub-2'); // 2nd sub-menu
    AppendMenu(hMnu, MF_STRING, idCmdFirst+3, 'Sub-3'); // 3rd sub-menu
    InsertMenu(Menu, indexMenu, // insert menu with the sub-menus
    MF_STRING or MF_BYPOSITION or MF_POPUP, hMnu, 'MyShellExt - Main');
    if bmp1.Handle <> 0 then // icon for Main
      SetMenuItemBitmaps(Menu, indexMenu, MF_BYPOSITION, bmp1.Handle, bmp1.Handle);
    // Return number of menu items added
    Result := 4;
  end;
end;
```

Now test the code and ensure that the icons are properly shown.

GSCL//Unclassified//