**TAxTaskDialog Document**

* **ClassName: TAxTaskDialog**
  + **This is the main object which you have to create and execute it.**

Here is some more Important Properties Descriptions:

* **DialogCaption: string;**
  + The Caption of the dialog window.
  + If left void, the Caption of the application’s main form is used.
* **Title: string;**
  + The dialog's Title content.
  + Any '\n' will be converted into a line feed.
  + If left empty, the Basic standard titles from Windows dialog classes will be used.
* **MainText: string;**
  + Main information text that you would like to inform the user.
* **CommandLinkButtons: Array of TCommandLinkButton;**
  + An array list of custom TCommandLinkButton expected here.
  + By default, the buttons will be created at the bottom of dialog form, just like the common buttons but If tdfUseCommandLinks flag is set, the custom buttons will be created as big button in the middle of the dialog window
* **RadioButtonsText: string;**
  + A '|' separated list of custom radio buttons expected here.
  + They will be identified with an ID number starting at 200
  + aDefaultRadio parameter can be set to define the default selected RadioButton
  + '\n' will be converted as shown as popup hint of the RadioButton.
* **ExpandableText: string;**
  + The Expandable information content text
  + Any '\n' will be converted into a line feed
  + By default, This Text will be shown in the Middle of form and under the MainText but if tdfExpandFooterArea is set the Expandable text will be shown at the bottom of form above the footer.
* **ExpandedButtonTooltip: string;**
  + The button ToolTip to be displayed when the information is ready to collapse.
* **CollapsedButtonTooltip: string;**
  + The button ToolTip to be displayed when the information is ready to Expand.
* **FooterText: string;**
  + The footer content text
  + Any '\n' will be converted into a line feed
* **ComboBoxItems: string;**
  + A '|' separated list of custom Text expected here.
  + If set, a Combo Box will be displayed to select
  + If tdfUserInputeTextBox is in the flags, the combo box will be in edition mode and the user will be able to edit the Items or fill the field with one item of the selection
* **UserInputedText: string;**
  + Some text to be edited (TAxEdit)
  + if tdfUserInputeTextBox is in the flags, the created EditBox will contain the default value.
* **RadioResult: integer;**
  + The selected radio item
  + First is numerated 0
* **ComboBoxSelectionResultIndex: Integer;**
  + After execution, contains the selected item from the Selection list
* **VerifyChecked: Bool;**
  + Reflect the bottom most optional checkbox state
  + If Verify is not ' ', should be set before execution
  + After execution, will contain the final checkbox state
* **Execute**
  + launch the TaskDialog form
* **ClassName: TExtraButtonParentClass (TBitBtn)**
  + A generic Button to be used in the User Interface
  + It is always a Themed button, Notice that under Delphi 6, since TBitBtn is not themed, It will be a raw TButton with no glyph.
* **TExtraButton = class(TExtraButtonParentClass)**
  + CreateEx
    - Is a constructor and Create a standard instance of button?
    - ModalResult/Default/Cancel properties will be set as expected for this kind of button
  + SetBitmap
    - Set the glyph of the button
    - Set nothing under Delphi 6
* **Flag List**
  + (tdfEnableHyperLinks, tdfUseCommandLinks, tdfUseCommandLinksNoIcon,

tdfExpandFooterArea, tdfExpandByDefault, tdfVerificationFlagChecked,

tdfShowProgressBar, tdfShowMarqueeProgressBar, tdfCallbackTimer,

tdfPositionRelativeToWindow, tdfRtlLayout, tdfNoDefaultRadioButton,

tdfCanBeMinimized, tdfUserInputeTextBox, tdfExpandButton,tdfEmailButton)

* **function UnAmp(const s: string): string;**
  + Return the text without the '&' characters within
* **Function CR(Const s:string): string;**
  + Return the text with lines feed instead of any ‘\n’.
* tdfEmailButton
  + if tdfEmailButton is set, a CommandLinkButton will be added automatically and the developer doesn’t have to make ‘Send Mail’ button manually.
* **Usage: Take a look to sample project.**