Prompt and Selection Lists with the new the Grids

John Tuohy

Prompt Lists

- Prompt lists are designed for making a selection
- The selection process
  - User does something to invoke the prompt list
  - A modal panel is presented and the user makes a choice
  - The panel is closed and the underlying object is updated
- The prompt update process can be:
  - Automatic – where the selection is updated back to the invoking object with little or no extra code
  - Manual – where the selection is made and the underlying update must be updated with code

Invoking and Non-Invoking Prompt Lists

- Invoking prompt lists
  - The basic process for all of these modes are:
    - Store any default set up behaviors that might be changed
    - Based on invoking context, prepare the list (update mode, initial column, etc.)
    - Load the data
    - Seed the list and move to seeded row
    - Popup the prompt list
    - Make a selection
    - Update the invoking object
    - Restore any default set up behaviors you might have used
    - Close the prompt list
  - This process is complicated but much of this is done automatically
- Non-Invoking prompt lists
  - The basic process for this mode is:
    - Load Data
    - Seed the list and move to seeded row
    - Popup the prompt list
    - Make a selection(s)
    - Close the prompt list
    - Process the selection(s)
  - This process is less complicated but requires more coding

Prompt List Update modes

- Four styles of prompt lists are supported
  - Relational
    - A selection is made and an underlying DD record is updated
    - The invoking object and prompt list must use the same table
  - Move value out
    - A selection is made and the underlying form is updated with the selection
  - Custom
    - A selection is made and the underlying update must be manually updated within a callback
  - Non-invoking
    - A selection is made and the underlying update must be manually updated

Update Mode and interface requirements

- The Update Mode (peUpdateMode) determines the interface requirements between the invoking object and the prompt list

Update Mode	Prompt Object	Invoking Object	
Relational *umPromptRelational*	Send Popup	Must be focus Get Value Get Data_File Get Data_Field Get Server Send Prompt_Callback	Can be fully automatic with no custom code. Custom code in Invoking object * Only available with cDbCJGridPromptList
Move Value Out *umPromptValue*	Send Popup	Must be focus Get/Set Value Set Item_Changed_State Send Prompt_Callback	Can be automatic with little or no custom code. Custom code in Invoking object *
Custom *umPromptCustom*	Send Popup	Must be focus Send Prompt_Callback	Requires custom code. Custom code in Invoking Object *
Non-Invoking *umPromptNonInvoking*	None – create your own custom method	None – call the prompt list custom method	Requires custom code in the prompt object. Invoking only needs to understand the custom method

* Requires custom code in the invoking object's Prompt_Callback, which requires knowledge of additional prompt object interfaces (mostly properties)

Prompt_Callback

- Prompt_Callback
  - Is sent from the prompt object back to the invoking object, giving the invoking object a chance to customize the popup instance
  - Most of these changes are made by changing properties:

```
Set peUpdateMode of hPrompt to umPromptValue
Set peUpdateColumn of hPrompt to 1
```

  - These properties are saved before the popup and restored after prompt is closed
    - See OnStoreDefaults and OnRestoreDefaults
  - If the update mode is non-invoking, none of this happens

The “Db” or data aware prompt list

- Based on the cDbCJGridPromptList class
- Requires a DD server
- Is used for loading and selecting database data
- Data can be loaded automatically
- Supports all modes including relational mode
- Can be dynamic or static
- Can be single select or multi-select
  - Multi-select only supported with static list
  - Multi-select will probably use non-invoking update mode
- This is your classic selection list

The regular (non Db) prompt list

- Based on the cCJGridPromptList class
- Does not requires a DD server
- Is used for loading and selecting any kind of data
- Data must be loaded manually
- Supports all modes except relational mode
- Is static
- Can be single select or multi-select
  - Multi-select will probably use non-invoking update mode
- Can be used with DEO and non-DEO invoking objects
- This is new to 16.1

Examples

- Now let’s look at some prompt lists in greater detail