Browse_List_Control

Description

Manages browse list functionality for a specific databound form.

Syntax

```
Ans = Browse_List_Control(Window, BrowseType, IDList)
```

Returns

Text display of the record position within the total number of records listed.

Parameters

Parameter	Description
Window	Current window in that has focus
BrowseType	The type of Browse command is being given (See Remarks below)
IDList	An @FM or @VM delimited list of keys for the browse list

Remarks

OpenInsight provides a built-in method for managing browse lists within a databound form: Query By Form (QBF). By default, the QBF menu appears on new forms created with the Form Designer. However, the functionality of the browse capabilities is sometimes too limiting, especially since the special QBF based events (e.g. QBFFIRST, QBFNEXT, QBFPREV, and QBFLAST) do not trigger the READ event.

To provide an alternative to QBF, the Browse_List_Control procedure was created. It uses the FrameWorks provided promoted event handlers for record management, therefore any special logic that needs to execute when a record is read from the disk is guaranteed to happen. The developer is also free to to modify this procedure to customize how browse list management should work if the default handling is not adequate.

Browse_List_Control does not provide form based searching capabilities like the QBF feature does. However, QBF is limited because not all data fields are necessarily represented on the form. For instance, some fields might be displayed in an ActiveX control which is not databound.

Record searches can be accomplished using the SRP Record Search Utility (RSU), a free utility that ships with FrameWorks. By default, this utility is activated by the Open button that appears on FRW_MAIN. The RSU can be configured to return single or multiple results, thus allowing the developer to use the Browse_List_Control procedure to handle the display of the results.

BrowseType Values:

Value	Function
Initiate_List	Clears the browse list and disables the browse navigation buttons. Used in Promoted_Close_Event
Create_List	Creates the browse list using the delimited list of Key IDs passed through the IDList parameter
Check_List	Updates the browse status of the current window. Used in Promoted_Activated_Event and Promoted_Read_Event
Forward_List	Moves the browse pointer one Key ID forward in the list. Used by the PUB_TB_BL_FORWARD button on the FRW_MAIN form
Backward_List	Moves the browse pointer one Key ID backward in the list. Used by the PUB_TB_BL_BACKWARD button on the FRW_MAIN form
Delete_Item	Removes the Key ID from the browse list. Used in Promoted_Delete_Event

Only the **Create_List** value needs to be added by the developer to utilize the features of the Browse_List_Control procedure. The other values have already been added to the appropriate promoted and FRW_MAIN button events.

Example

```
Param = ""
Param<5> = Yes$ ; // Enable multi-select
ID = Dialog_Box("SRP_RSU", @Window, Param)

 If ID GT "" then
   NumItems = Count(ID, @VM) + (ID NE "")
   If NumItems EQ 1 then
       // If only one key ID then populate the key controls and read the record
       rv = Set_Window_Key(ID, Yes$, @Window)

   end else
       // If more than one key then use Browse_List_Control
       rv = Browse_List_Control(@Window, "Create_List", ID)

   end
end
```

See Also

FRW_Main_Events, Promoted_Activated_Event, Promoted_Close_Event, Promoted_Delete_Event, Promoted_Read_Event, Update_Window_Text