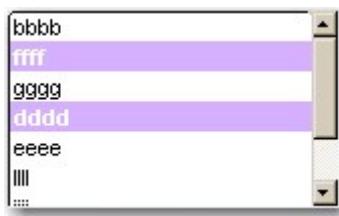




LAF TList PJC - public properties

These properties can be set or read from any List Item of type TList whose Implementation Class property is set to:

`oracle.forms.fd.LAF_XP_TList`



Content

CLEAR_LIST_SELECTION	3
GET_LIST_ORIENTATION	3
GET_LIST_SELECTION	3
GET_MULTI_SELECTION	4
GET_SORTED_LIST	4
SET_ENHANCED	4
SET_ENHANCED_ALL	5
SET_LIST_ORIENTATION	5
SET_MULTI_SELECTION	6
SET_SORTED_LIST	6
SET_DEBUG	7
Events raised to Forms	7



CLEAR_LIST_SELECTION

Clear the selected values.

```
Set_Custom_Property( '...', 1, 'CLEAR_LIST_SELECTION', '' ) ;
```

GET_LIST_ORIENTATION

Returns the list orientation (only with enhanced Tlist).

```
varchar := Get_Custom_Property( '...', 1, 'GET_LIST_ORIENTATION' ) ;
```

The possible return values can be:

- VERTICAL
- VERTICAL_WRAP
- HORIZONTAL_WRAP

GET_LIST_SELECTION

Returns the comma delimited list of selected indexes.

```
varchar2 := Get_Custom_Property( '...', 1, 'GET_LIST_SELECTION' ) ;
```

The first indice is 0, so if the user has selected the 1st, 3rd and 5th value the method will return : 0,2,4

```
Declare
    LT$Coll    PKG_LOOK_AND_FEEL.TYP_TAB_STRINGS ;
    LC$Sel     Varchar2(4000) ;
    LC$Label   Varchar2(128);
    LC$Value   Varchar2(128);
    i          pls_integer := 1 ;
Begin
    -- get the selected values --
    LC$Sel := Get_Custom_Property( 'BL.LIST4', 1, 'GET_LIST_SELECTION' ) ;
    If LC$Sel is not null Then
        -- transform comma delimited string into collection --
        PKG_LOOK_AND_FEEL.To_String_Collection( LC$Sel, LT$Coll ) ;
        LC$Sel := '' ;
```



```

-- loop through the collection --
For i In LT$Coll.First .. LT$Coll.Last Loop
    -- list label --
    LC$Label := get_List_Element_Label( 'BL.LIST4', LT$Coll(i)+1 );
    -- list value --
    LC$Value := get_List_Element_Value( 'BL.LIST4', LT$Coll(i)+1 );
    If i > 1 Then
        LC$Sel := LC$Sel || CHR(10);
    End if ;
    LC$Sel := LC$Sel || LC$Label || ' (' || LC$Value || ')';
End loop ;
End if ;
Set_Alert_Property( 'AL_INFO', ALERT_MESSAGE_TEXT, LC$Sel ) ;
i := Show_Alert( 'AL_INFO' ) ;
End ;

```

GET_MULTI_SELECTION

Returns a varchar2 that indicates if the multi-selection is allowed for the list item.

Returned value van be TRUE or FALSE

```
varchar2 := Get_Custom_Property( '...', 1, 'GET_MULTI_SELECTION' ) ;
```

GET_SORTED_LIST

Returns a varchar2 that indicates if the enhanced list is sorted.

Returned value van be TRUE or FALSE

```
varchar2 := Get_Custom_Property( '...', 1, 'GET_SORTED_LIST' ) ;
```

SET_ENHANCED

Set/unset the enhanced (Swing) Tlist.

property value : true|false

Set the value to false to get the native Forms Tlist widget.

Set it to true to overload the native Forms Tlist with a Swing list widget.

```
Set_Custom_Property( '...', 1, 'SET_ENHANCED', 'true' ) ;
```



SET_ENHANCED_ALL

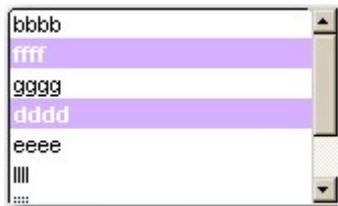
Item as SET_ENHANCED but for every Tlist of the current module.

SET_LIST_ORIENTATION

Set the enhanced Tlist orientation.

Allowed values:

- VERTICAL



- VERTICAL_WRAP



- HORIZONTAL_WRAP



Without indication, the Tlist is created with a default value of : **VERTICAL**

```
Set_Custom_Property( '...', 1, 'SET_LIST_ORIENTATION', 'VERTICAL_WRAP' ) ;
```



SET_MULTI_SELECTION

Set the multi-selection flag.

Allowed values:

- **true** : you can select more than one value in the list
- **false** : you cannot select more than one value

Without indication, the Tlist is created with a default value of : **false**

```
Set_Custom_Property( '...', 1, 'SET_MULTI_SELECTION', 'true' ) ;
```

If the value is set to true, you can get the list of selected indexes with the *GET_LIST_SELECTION* method.

SET_SORTED_LIST

Set the sort flag (on enhanced Tlist only).

Allowed values:

- **true** : the Tlist is sorted
- **false** : the Tlist is not sorted

Without indication, the Tlist is created with a default value of : **false**

```
Set_Custom_Property( '...', 1, 'SET_SORTED_LIST', 'true' ) ;
```



SET_DEBUG

Turn ON/OFF the debug messages.

By default the all logs are OFF.

Allowed values are : 'true' or 'false'.

```
Set_Custom_Property( '...', 1, 'SET_DEBUG', 'true' ) ;
```

Events raised to Forms

While a new value is selected, the Tlist raises an event to Forms, via the MOUSEEVENT_MESSAGE event:

```
-----  
-- Mouse events --  
-----  
ELSIf (EventName = 'MOUSEEVENT_MESSAGE') then  
    get_parameter_attr(eventValues,'ITEM_NAME',eventValueType, p1);  
    get_parameter_attr(eventValues,'MOUSE_EVENT',eventValueType, p2);  
    if p2 = 'TLIST_INDEX' And p1 is not null then  
        get_parameter_attr(eventValues,'ITEM_VALUE',eventValueType, p3);  
        get_parameter_attr(eventValues,'ITEM_SELECTION',eventValueType, p4);  
        go_item(p1);  
        copy(get_List_Element_Value( p1, p3+1 ),p1);  
        execute_trigger('WHEN-LIST-CHANGED');  
    End if;
```

So, the When-List-Changed trigger of the List Item is invoked.

Of course, to have the Items raising events back to Forms, you have to indicates at form startup:

```
-- for 9i and 10g versions only --  
Set_Custom_Property('LAF_BLOCK.LAF_BEAN', 1, 'SEARCH_TEXT_ITEMS','') ;  
  
-- for all version --  
Set_Custom_Property('BL.LIST_ITEM', 1, 'ENABLE_EVENTS','BL.LIST_ITEM,true');
```

Whith Forms 11g, the first instruction is not required.



Oracle Forms Look & Feel project

Created and maintained by Francois Degrelle

[Oracle Forms L&F Web site](#)



Oracle Forms Look & Feel project