Mobile Design: Listbox Usability

List or list box is a powerful GUI (Graphical User Interface) tool which is used to display more then one items at a time to the user, sometimes with the intent of allowing the user to select one or more then one items from the list, while at other times the intent is just to display a non selectable list of content. List allows the user to view a logical collection of related items like Contacts, SMS, and Settings etc.

For more details on list and list boxes check the links below:-

Wikipedia

S60 UI Component: Lists

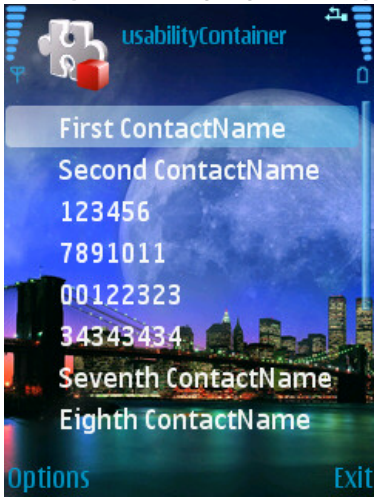
While list boxes are a powerful GUI component by themselves but from a usability perspective it is very important to ensure that the user doesn't feel overwhelmed or irritated in using the list box. Every aspect of the application has to ensure a nice user experience and list boxes are no different.

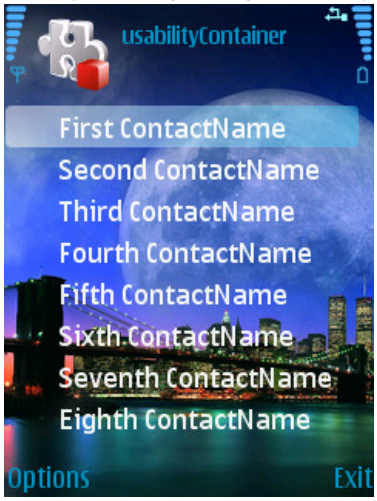
Some common things to remember while using a list box are:-

- **Display only logical related contents in the list box**

The list box by its definition is a GUI element to display a logically related set of items/entities. Always ensure that this golden rule of usage is followed, for instances don't mix content which doesn't belong together into a list, as this would end up confusing the user. For instance if you are developing a log application, make sure that you show only missed calls in the missed call lists, received in received call lists and so on. Mismatching would not only be an irritant for the user it would also make the program logic and later code maintenance a tougher task from a developer team's perspective.

Example of wrong logical Grouping **Example of Right Logical Grouping**





- **Compress display content on the list box item**

Although the lists allow you to display a lot of content in the individual items, but it is always a good idea to compress the text to only the most important/relevant content. This is all the more critical on the constrained display area of the mobile device, as you don't want a single item hogging most of the real estate leaving hardly any space for the other items. It would also be a good idea to show the long/detailed text in a separate detailed view which can be opened by pressing the Select/Center Soft Key of the device. While on the list view you can truncate the content by displaying ellipsis at the end to denote to the user that there is more text which is not being displayed because of space constraints.

If you think that beyond a certain length, text doesn't make sense you should possibly bring up an information note or some other dialog to notify the user of the same rather then just blocking user input while the user is entering the text.

- **Enable marquee in case you have lot of content in the list**

(C) Copyright Nokia 2014. All rights reserved.

You should consider using a marquee in case you think that the entire content has to be displayed on the list itself and you don't want the user to navigate to a detailed view to see the content neither do you want to truncate the text for one reason or the other.

Example of listbox with no marquee

Example of listbox with marquee

For more on marquee:-

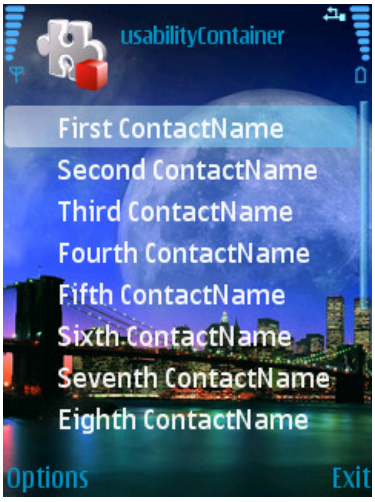
Marquee Scrolling

- **Implement scroll bars in case you have more content**

More often then not the list could have more items then what can be displayed on the view in one shot, in those cases you should allow the user the ability to scroll up/down the list so that they can view all the items. In cases of a custom drawing implementation make sure that the content being displayed on the view is scrolled consistently i.e. at the same speed and gives the same consistent look and feel.

Example of listbox without scrollbar Example of listbox with scrollbar

(C) Copyright Nokia 2014. All rights reserved.



For more details on scrollbar implementation in Symbian C++, check:-

Scroller

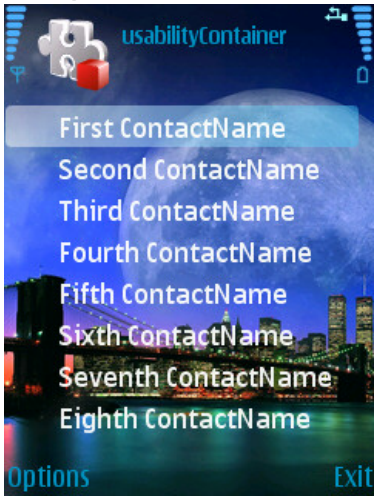
ScrollBar on List

- **Implement a search box specially if you have lot of items**

From a user's stand point it is imperative to provide them with the option to search a particular item, so that they don't have to scroll up/down and waste precious time in trying to locate an item. This is all the more crucial in applications like contacts/chat clients etc where you are most likely to have lots of contacts/buddies and the user might want to dial/chat quickly with a particular person.

Where-ever possible you should also provide the user to customize the search options like search by name/number etc. In some cases even implementing an auto-complete approach would give a nice experience to the user where they don't have to manually type in the entire name before the search results are displayed.

Example of listbox without search-box Example of listbox with search-box



For more details on search field implementation in Symbian C++, check:-

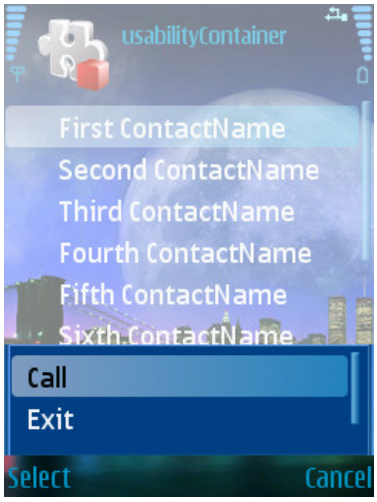
Search Field

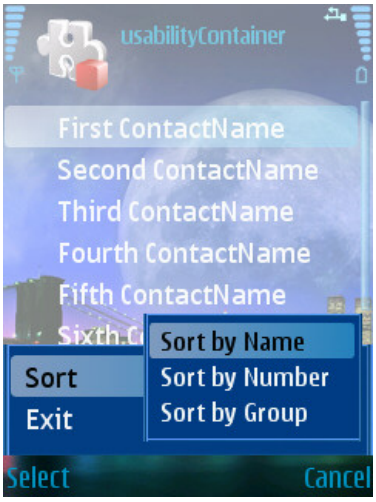
- **Allow user the option to sort the list in some orders**

Sorting is also a standard option that you should provide the user in cases of long lists. Sorting could especially be helpful in cases where the user wants to analyze the last usage patterns, recent dialed, most dialed, most viewed content etc. Sorting can be more aptly defined as not a must have but rather a good to have feature. Sometimes the good to have feature are what differentiate your product from a competitor's offering. For instance in case of a log application if you allowed the user to sort based on most talked, least talked, most received calls from number options it would be highly appreciated by the user and it can be very well used as a USP for the product.

Example of listbox without sort option Example of listbox with sort option

(C) Copyright Nokia 2014. All rights reserved.





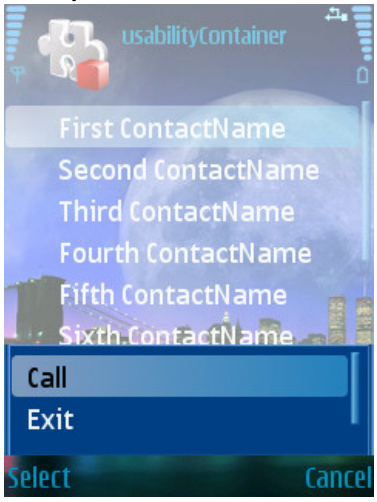
- **Use the right list box depending on the content being displayed**

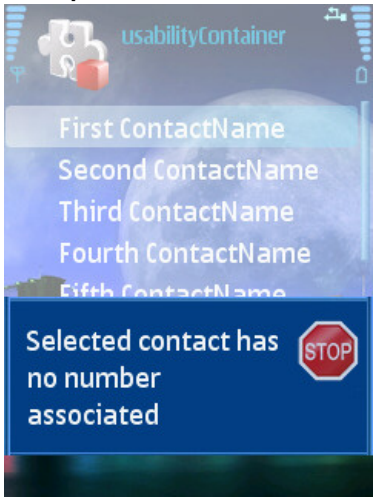
Always use the right list box from the options available instead of using a blanket “one fits all” approach. For instance if the intent is to just display text and no icons, do not use a list box which supports icons as well.

- **Show/hide menu options based on current list box selection**

It is very important to change the visibility of the menu items depending upon the currently selected item. You do not possibly want to show an option to the user which is not applicable for the current item and then either throw an error dialog stating ‘request not applicable’ or do nothing, both the cases are likely to be equally bad from a usability stand point.

Example of listbox with invalid commands **Example of listbox with invalid commands**



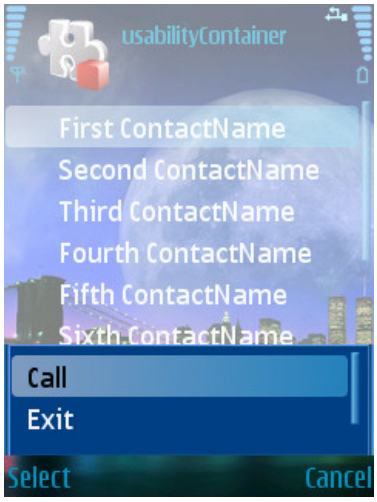


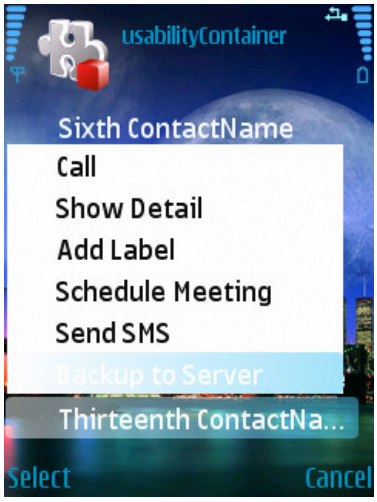
- **Show drop down menu options**

Sometimes it would be useful and desirable to show a menu in place itself rather then using the default menu layout where it pops up at the bottom of the screen. One should be careful in deciding the layout of the drop down menu and also have a mechanism to select the menu options and allow the user to cancel the drop down menu if they so desire.

Example of listbox with default menu **Example of listbox with drop down menu**

(C) Copyright Nokia 2014. All rights reserved.





- Icons/different backgrounds should be used to help the user quickly differentiate between types of items.
- While activating a view containing list, make sure the currently selected item is visible.
- If the list does not have any item to be displayed, inform the user about the empty list by displaying some text in the main pane of the application. In case there is a search box implementation, ensure the searchbox is hidden in this case.

Links

- How to display drop down/fly out menu
- How to create a simple listbox in Symbian C++

---- Added by Mayank on 11/06/2009 ----

(C) Copyright Nokia 2014. All rights reserved.