Best way to show screens to user on application

Asked 15 years, 9 months ago Modified 15 years, 9 months ago

Viewed 518 times

1

I'm developing a Winforms application which has been running for years with an explorer view (TreeView left, screen right). I means that:

- All the screens have an hierarchy organization
- All the nodes on TreeView have one and only one screen related.
- A screen gets activated when a node on treeview gets selected.

One of the advantages is that the user has an ordered stucture and one of the inconveniencies is that with hundreds of screens the user gets confused.

I see other options: use classical menus, use tabs or a mix of everything.

Any advice for a good way to show a lot of screens to user in a user-friendly way?

Update: I'm changed "hundreds screens" by "a lot of screens". The most important thing is not show all at time but that the user can find what they need easily.

Update2: In this proposal, the user only see one screen at time.

Update3: I'm talking about *handling* multiple screens not showing multiple screens. No MDI, **only one ontime**.

winforms user-interface user-friendly screens

Share

Improve this question

Follow

edited Mar 10, 2009 at 13:41

asked Mar 8, 2009 at 22:36

FerranB

36.7k ● 18 ● 69 ● 86

"show hundreds of screens to user in a user-friendly way" Seems like a contradiction... – SAMills Mar 8, 2009 at 22:42

Google shows billions of pages in an user friendly way. – FerranB Mar 8, 2009 at 22:49

It is unclear if you want the information in the "hundreds of screens" visible simultaneously. – RedBlueThing Mar 8, 2009 at 22:59

5 Answers

Sorted by: Highest score (default)

I have used other applications similar to this is the past, and the major problem is trying to find the exact screen

4

you want. There are two common solutions to this problem, shortcut codes and favorites menu.

With the shortcut codes, allocate a short code (5 or 6 characters) to each screen. The user then inputs this shortcut code into a text box which will then jump to the correct screen. Users will create their own list of often used codes.

For the favorites menu, allow users the ability to be able to create their own menu list in the structure they want. They will find things easier, if they organize it themselves.

Share Improve this answer

Follow

answered Mar 8, 2009 at 23:22

Craig T

2,744 ● 5 ● 25 ● 34

Expanding on these idea: search? Common for web browsers (and the web itself!) to have a search for bookmarks/favourites and history. – strager Mar 10, 2009 at 1:25

1

Why do you need to show so many seprate screens at once? Why not just show the screen for the currnetly selected node, why are all needed at once?

If it is all tabular data is is probably too much to be consumed all at once, if it is graphical data, could it not be combined?

There may be a valid reason to show all the data at once or there may not, hard to tell from what is provided in your question. With that said, better to keep it simple than overload the user. MDI apps are never easy to use.

Tabs may work for a small set of items but still is not a good UI for hundreds of items.

If you are only showing one element at a time, out of hundreds possible on the tree nodes, then that is fine. The one screen showing at a time would be contextual to the item selected as the user moves through the nodes. Think of the Outlook approach where what is selected in the left pane is displayed in the right pane in whatever form fits the data being displayed.

edited Mar 8, 2009 at 23:15

answered Mar 8, 2009 at 23:10

schooner
3,077 ● 8 ● 32 ● 39

Have you considered the Office Ribbon?

The Ribbon gives you a lot of flexibility on how to show and organize functions and it's highly visual.

Here is a good link about the Ribbon and also here

To use the Ribbon you have to license it from Microsoft. You can do that online.

Providing the user with ketboard shotcuts is usually a good thing too.

I also like to provide the user with an "autocomplete" field on the menu so that they can can find the function by name (or part of it) and be able to navigate directly to where they want to go.

Share Improve this answer Follow

edited Mar 8, 2009 at 23:27

answered Mar 8, 2009 at 23:18

Klinger

4,970 ● 1 ● 31 ● 35

0

I general I find trees to be a bad idea, especially if your "hierarchy" is of a small fixed depth.

If you have a small fixed depth, consider replacing the tree with a list. At the top of the list can be drop-downs for filtering based on the node-level properties. It will use up less screen real-estate because it is vertical-only, with no horizontal component.

Clicking on an item can display it in the view (like currently), but it may be a good idea to allow a user to double-click on more than one item which could launch more windows, or tile with the existing displayed items. (I

am assuming that currently, the user only sees a single detailed view at once in any given window.)

Share Improve this answer Follow

answered Mar 8, 2009 at 23:18

oxbow_lakes

134k ● 56 ● 320 ● 450

0

Actually, it's hard to beat a hierarchy for organizing large numbers of items. I wouldn't favor a classical pulldown menu for vast numbers of windows because it would be even harder to keep track of where you are than in a tree (e.g., a tree lets you look in multiple branches at once). But here's a few alternatives:

I'm not clear how you ended up with so many windows, but maybe it comes from combinations of classes, views, content, and detail, or maybe it comes from using a task-centered UI structure for something far too complex (I've more on that at http://www.zuschlogin.com/?p=3). For complex apps, you want a different primary window for each significant class of data object (e.g., invoices, employees). These are listed on one menu, and typically there's few enough (15 or less) that it can be single non-cascading pulldown menu. The content of each window is set by a separate menu, perhaps by a menu item that opens a dialog that may include a list box (like an Open dialog) or other controls for querying/searching. The "view" of each window (how the data objects are shown, e.g., table versus form) is set by menu items in the View menu. Details for any given object in a window can be

shown in a separate pane within the window in a master-detail relation, essentially turning you data objects into a menu for details. A single window can have multiple detail panes for the user to open and close to select the specific detail to show. Tabs may also be used within a single pane to fit subdivisions of content.

You say it’s not important to show all window options at once, but often showing all options at once makes it easiest for users to find what they need. Maybe you need a “home” window that lists all the other windows in organized, labeled, and separated categories. This is will be easier to use than the tree if your users select a window then stick with it for most of the session. Your tree is better if there's frequently selection of windows throughout the session, owing to the overhead of getting to the home window. If all windows/options don’t fit on a single home window, then show only selected common windows for each category on the home window and provide a button or link to show an exhaustive list.

If you’re talking 100’s of windows, maybe you should have Search, perhaps in addition to a menu-based browse approach to getting to a window.

In any case, providing easy access to the few most commonly used windows is a good idea. Such windows can be explicitly selected by the designer, based on user research, or selected by the the user (favorites), but it also typically works well to make it automatic with an

algorithm that uses some combination of frequency and recency of use.

Share Improve this answer Follow

edited Mar 10, 2009 at 11:21

answered Mar 10, 2009 at 1:20

Michael Zuschlag

4,760 ● 17 ● 14