

AutoNavigator for FileMaker Pro™ 2.0 User Manual

AutoNavigator 2.0v1 © 2006 Foundation Database Systems/David Kachel, all rights reserved.

Foundation Database Systems and its functionaries provide absolutely no warranty, express or implied, for the enclosed software product and shall not be held liable for any damage or loss resulting from the use of AutoNavigator or distribution of products or software solutions created using AutoNavigator.

Due to the fact that AutoNavigator is provided in an open and completely modifiable state, under no circumstances will Foundation Database Systems issue a refund of the purchase price. All sales are final.

CONTACT INFO:

Foundation Database Systems
www.foundationdbs.com
support@foundationdbs.com

3945 S. Wasatch Blvd.
Suite 266
Salt Lake City, UT 84124
801-566-1340

RE: THESE INSTRUCTIONS: - (Print these instructions so you can more easily follow along when working with the files.)

We **STRONGLY** recommend that you read this document in its entirety before proceeding.

If you just can't stand it and absolutely **HAVE** to explore one of these files before reading this entire document, jump down the page to "GETTING STARTED & UNDERSTANDING THE HIERARCHY" and follow the instructions there. But then be sure to come back and read everything. You'll need it!

TABLE OF CONTENTS:

1. PACKAGE CONTENTS
2. UPDATE & RELEASE HISTORY
3. DESCRIPTION
4. DEVELOPER OVERVIEW
5. DEVELOPER SCRIPTS
6. NAMING LAYOUTS
7. TAB LABELS
8. CREATING USER LAYOUTS
9. DELETING USER LAYOUTS
10. CREATING DEVELOPER LAYOUTS
11. HOW TO THINK ABOUT LAYOUTS
12. CHANGING USER LAYOUT NAMES
13. GETTING STARTED & UNDERSTANDING THE HIERARCHY
14. WHEN THE SYSTEM IS CONFUSED
15. SUBSTITUTING YOUR OWN TAB GRAPHICS
16. BUILDING A NAVIGATION SYSTEM AUTOMATICALLY
17. BUILDING A NAVIGATION SYSTEM MANUALLY
18. PRE- & POST- TAB SCRIPTS
19. CONTROLLING FILEMAKER TAB OBJECTS
20. PRECAUTIONARY NOTES

1. PACKAGE CONTENTS:

In addition to this instruction file, there are 3 AutoNavigator files enclosed:

AutoNavigator2.0v1.fp7 — The actual AutoNavigator file you will build with.

AutoNavigatorSample1.0v4.fp7 — A version 1.0v4 demo file with lots of layouts for you to explore and experiment with.

AutoNavigatorStripped1.0.fp7 — A version 1.0 faux completed project, with everything stripped out of it except for the navigation system. (Some scripts and Custom Functions missing because this stripped file was created in an earlier version.)

DO NOT OPEN THE ABOVE FILES DIRECTLY!

Instead, always use duplicates. AutoNavigator2.0v1.fp7 is the actual template file you will use to start new projects. The other two files are for tutorial/practice/demonstration purposes. We suggest burning all 3 files, along with this text file, to a CD for safe storage.

If you should damage or lose any of the original "AutoNavigator" files, we will replace them free.

2. UPDATE & RELEASE HISTORY:

10-13-06

New release, version 2.0v1. AutoNavigator will now handle layouts up to 7 levels deep (formerly 5) and up to 20 tabs per level (formerly 10). AutoNavigator 2.0v1 now contains pre- and post- scripts you can alter to fit your needs for all tabs at all levels (see section "18. PRE- & POST- TAB SCRIPTS"). In addition, AutoNavigator will now track built-in FileMaker tab panels under all circumstances, both while navigating between layouts and when logging back into a solution. (see section "19. CONTROLLING FILEMAKER TAB OBJECTS").

7/11/05

Still getting some reports of users having to manually edit Navigation records when they have used the same word in two different hierarchies, particularly when the word is used in a top-level layout and a default second level layout. To avoid this problem, when you know you are using a word for the second time, first use a temporary substitute word to name the layout, like XXXXXXXXX. Run the TabsControl script as usual, then rename the layout with the word you really want to use, and run TabsControl again.

AutoNavigator will replace XXXXXXXXX with the correct word without disturbing the other Navigation records.

6/29/05

Minor update to this document.

4/18/05

This is the revised instruction manual for version 1.0v4. This version of AutoNavigator contains several minor bug fixes, and a substantial upgrade to the process for automated creation of Navigation records. It is now almost impossible to break AutoNavigator's routines for the automated creation of Navigation records. You may also now freely use the same word in different hierarchies without worrying that AutoNavigator will change that word in hierarchies other than the one where you wish it to be changed. (Words can be repeated anywhere but at top-level tabs and within the same hierarchy; unlikely events.) We have added 2 new scripts and 2 new fields, deleted one field, added 3 new developer layouts, and substantially altered several existing scripts.

4/10/05

This is not the AutoNavigator 1.0 file we had originally planned to release. This is actually a hybrid version combining the features of the originally completed version 1.0 and some of the features slated for the 2.0 version. In the original 1.0 version, all Navigation records had to be created manually, typing in all the required information in each Navigation record for each layout. As we found ourselves with more time than expected after development was complete and before the release date for version 1.0, we decided to see if we could add some of the 2.0 features to the 1.0 file before the release deadline. We succeeded, and hope you will feel you have benefited from that success.

While you can still build/edit Navigation records manually, and may wish to do so under some circumstances, AutoNavigator 1.0 will now build and modify all of your Navigation records for you on the fly, with just one keypress. It will also automatically propagate layout name changes throughout all of your Navigation records.

3. DESCRIPTION:

AutoNavigator is a simple FileMaker Pro™ version 7/8 file. It does not require any plugins or other extras. It is intended to be used as a starting template for all of your FileMaker projects. AutoNavigator can build a complex, multi-level tab and navigation system into your solutions quickly and easily, turning many hours of work into minutes.

AutoNavigator comes preloaded with tab sets you can copy and paste onto your layouts (this is now automated also) for up to 7 levels of tabs, with up to 20 tabs at each level. In other words, you can have up to 140 tabs on each layout!

AutoNavigator places no restrictions on layout design whatsoever. Your "tabs" can be any size or appearance, and can be placed anywhere on your layouts. And you can use a

different set of tab graphics for each level in the hierarchy (for those who like that "angry fruit salad" appearance in their solutions).

AutoNavigator also keeps track of the last layout every user was on in every hierarchy. No users will ever find themselves on an "unexpected" layout, even after logging out and back in.

And best of all, AutoNavigator is completely open. Change the code in any way you like to accommodate the special needs of your solution. (Note: This does not mean "Open-Source". AutoNavigator is fully copyrighted and cannot be... well, you know the drill: If you paid for it, you can use it, but you don't own it, can't sell it, can't distribute it, or claim to have created it.) You are NOT required to credit AutoNavigator or Foundation Database Systems anywhere in your solutions, but it would be a nice gesture.

4. DEVELOPER OVERVIEW:

AutoNavigator 2.0v1 consists of:

- 3 tables, (total of 10 table occurrences)

- Main** - where your tabbed layouts are seen, and where you will start your solution

- Navigation** - contains 1 record for each layout used by Main, plus all tab graphics

- Preferences** - stores tracking information of last layout visited by each user in all hierarchies

- 9 relationships

- 46 scripts

- 102 fields

- 9 developer layouts

- 1 user layout

The "Navigation" relationship between the Main and Navigation tables is what allows AutoNavigator to know which tab graphics and tab labels to display on each layout, and which layout to go to when a tab is clicked. (The primary key is the field "gCurrentLayout". The foreign key is the "Layout_Name" field.)

For each user layout, there must be a Navigation record in the Navigation table containing 28 repeating fields (4 for each level in the hierarchy) of 20 repetitions each, plus one field for the name of the layout it represents (the foreign key).

CAUTION:

Under no circumstances will you need to alter any field definitions, scripts or buttons in order for AutoNavigator to work. If you are changing one of the aforementioned for any reason other than adapting the solution for special needs, you are making an error.

ADMONITION:

This is not shrink-wrapped software for consumer use. This is a developer tool! It is not guaranteed to be flawless, or to exactly fit your development needs or style, or to anticipate every error a developer might make. It is assumed that you are a qualified developer and that if the software won't work the way you want it to, you have the skills to change it; or you may opt to pay Foundation Database Systems to build a custom version. We solicit and will be grateful for your bug reports and feature suggestions, but we do not in any way guarantee this software to work in a specific way or to fix "bugs" on demand.

Reverse Engineering:

You do not need to understand how AutoNavigator works to use it, but to get the most out of AutoNavigator, we recommend you determine how it works internally. Any reasonably experienced FileMaker developer could have built AutoNavigator. There is nothing particularly difficult or new in AutoNavigator from a programming standpoint. You probably won't see any techniques or ideas you haven't seen before. What we've done is simply to suffer a lot of programming tedium in order to build a tool that, well... helps us avoid a lot of programming tedium. We sincerely hope it will do the same for you.

5. DEVELOPER SCRIPTS:

There are six developer scripts named: "Tabs Control", "Copy Tab Set", "Tabs Storage", "Developer Layout Change", "Reset", and "Send BugReport/FeatureRequest", respectively. (Uncheck the "Include in menu" checkboxes for these scripts before you deliver your solution.) You will use the first five scripts in creating layouts and Navigation records. (The purpose of the sixth script is obvious.)

1. Use the "Tabs Control" script (Command-1, Mac) (Control-1, Windows) to switch back-and-forth between the user layout you are working on and the "Developer-Tabs-Control" developer layout, which is where you will enter/review/edit the Navigation record for that layout. (If the user layout you are on is a new (duplicated) layout, or you have changed its name, this script will create a Navigation record for the new layout, or propagate the name change through the other Navigation records.) If you do not want AutoNavigator to automatically create a new Navigation record, or propagate a layout name change automatically, hold down the Shift key while selecting this script from the Scripts menu. You will be taken to the Navigation record for the original layout you copied, or changed the name of, where you can manually create new Navigation records or alter other records. (See the section titled "BUILDING A NAVIGATION SYSTEM MANUALLY" for more information.)

2. Use the "Copy Tab Set" script (Command-2, Mac) (Control-2, Windows) to automatically copy to the clipboard the correct tab set (level and length) for your purposes. A dialog will ask you for the tab level and tab set length, and will then place

that tab set on the clipboard for you. Then just enter layout mode and paste. This script will not work if you are on a Developer layout. (See the section "SUBSTITUTING YOUR OWN TAB GRAPHICS" for information on how to change tab graphics and tab set sizes.)

3. Use the "Go To Tabs Storage" script (Command-3, Mac) (Control-3, Windows) to switch back-and-forth between the layout you are working on and the "Developer-Tabs-Storage" layout, which is where you can go to manually copy the pre-set tab sets for your layouts. (You should seldom if ever need to do this because the "Copy Tab Set" script above is one of the version 2 features we included at the last minute. It is much more convenient.) If you need to change the size of the fields and buttons used in your tab sets, you will do so on this layout and then paste the altered sets into the repeating fields on the "Developer-Tabs-Containers" layout which is where the "Copy Tab Set" script goes to grab a tab set for you. (See the section "SUBSTITUTING YOUR OWN TAB GRAPHICS" for information on how to change tab graphics and tab set sizes.)

4. Use the "Developer Layout Change" script (Command-4, Mac) (Control-4, Windows) IMMEDIATELY after creating OR deleting each and every layout for DEVELOPER use (not user layouts). This is VITAL, or AutoNavigator may damage your Navigation records the next time you use the "Tabs Control" script. Also use this script after deleting (BUT NOT AFTER CREATING) a user layout. (More on this below)

5. Use the "Reset" script (Command-5, Mac) (Control-5, Windows) Whenever the system becomes "confused", i.e, the tabs disappear: this can only happen during development (it has never been known to happen with end-user use) and is an indication of a developer error. (More on this below)

6. NAMING LAYOUTS:

The naming of layouts must follow strict rules. NO EXCEPTIONS. You must employ WHOLE-WORD, hyphenated names for your layouts. For example; a layout can be named "myLayout" but not "my Layout". No numbers, spaces or punctuation of any kind are permitted: only upper and lower case letters. The only exception to this rule is the hyphen (-). This character is reserved, and must be used to differentiate layouts that are at lower levels in the hierarchy.

There is one FORBIDDEN WORD in the naming of user layouts: "Developer". This word must not be used in naming any user layout at any level. It is reserved (and required) for the naming of developer layouts. (More on this below.)

All layout names must be unique! OK, that's fairly obvious, but it still had to be said.

AVOID USING THE SAME WORD TWICE AT LOWER LEVELS OF THE SAME HIERARCHY. Layout names such as; "Home-Detail-Addresses" and "Home-List-Addresses" may confuse AutoNavigator (Home-Addresses-Detail and Home-Addresses-List will not), as both layouts are in the same ("Home") hierarchy and "Addresses" appears at the lower level of both layout names. You MAY use the same word in different hierarchies. Give careful thought to naming layouts before creating them, and be extra cautious about renaming a layout. If in doubt, use a temporary word such as "XXXXXXXX" in your layout name, create your layout(s) using the "Tabs Control" script, then change the temporary word to the name you really want and run the "Tabs Control" script again.

Also, remember that FileMaker does not allow any scripted access to objects in layout mode. Therefore if you have a hierarchy of "Home" layouts, and want to change "Home" to "Main", AutoNavigator will change all of your Navigation records to reflect this change, but not the actual names of the layouts. You must manually change the word "Home" to "Main" in all your layouts.

Permitted Layout Name Examples:

Accompanying Tab Labels (OK to change):

Home	(1st level layout)	Home
Home-List	(2nd level layout)	List
Home-Detail	(2nd level layout)	Detail
Home-Detail-Phones	(3rd level layout)	Phones
Home-Detail-Addresses	(3rd level layout)	Addresses

Non-Permitted Layout Name Examples:

Home1

Home List

Home-Detail Phones

Home 27

(Home)

Home\$Detail@Phones
Developer
Main-Developer

The number of words, separated by hyphens, in a layout name indicates the hierarchy/tab level at which the layout exists. For example:

Home	indicates a first-tab-level layout
Home-Detail	indicates a layout that is at the second level of tabs
Home-Detail-Phones	indicates a layout at the third level of tabs, etc.

The naming of layouts must follow the above convention in order for the navigation system to work.

Follow the same naming convention if you add any new developer layouts. **ALL LAYOUTS IN YOUR SOLUTION, REGARDLESS OF THEIR PURPOSE MUST FOLLOW THE ABOVE RULES**, with the exception that developer layouts do use and are required to use, the word "Developer" in their names (see below).

7. TAB LABELS:

Tab labels are the text that actually appears over the tab on your layouts (not the layout name). AutoNavigator automatically sets each tab label to be the last whole word in the name of the layout to which the tab navigates. If your layout name is "Home-Detail-Addresses", the tab label will read "Addresses". If you change a layout name, AutoNavigator will automatically change the corresponding tab labels throughout the system, to correspond to the altered layout name. The tab labels that appear to the viewer do not have to retain these automatically generated names. If you wish, when your layout creation is complete, you can manually alter (or search and replace) any or all tab labels. Tab labels do not have to follow the layout naming rules, but leave them "as generated" by AutoNavigator until you are sure you have finished building layouts. Otherwise, AutoNavigator will alter them for you when you add another layout, or enter a layout name change. Another good reason for this is that if you change your tab labels before your project is near completion, you may get a bit confused while moving around through the project if the labels do not correspond to the names of the layouts they represent.

8. CREATING USER LAYOUTS:

CAUTION: This does not apply to creating developer layouts. See below.

NEVER select "New Layout/Report..." from the Layouts menu when creating a layout on which AutoNavigator tabs will appear. ALWAYS create a new user layout by DUPLICATING and renaming an already existing user layout. Duplicate and rename only ONE layout at a time. Then IMMEDIATELY run the "Tabs Control" script (If you do anything else before running the "Tabs Control" script, especially going to another layout, AutoNavigator will create a new Navigation record for the wrong layout!). Being on the layout in question is the only way the "Tabs Control" script has of knowing how to correctly create a new Navigation record for your new layout. It will also automatically alter all the other Navigation records to reflect your new layout. Again, remember to always run the "Tabs Control" script immediately after creating a new user layout. Only then, should you alter the new layout or create another.

If you absolutely insist on creating a new layout by using the "New Layout/Report..." command; create your layout, then hold down the Shift key while selecting the "Tabs Control" script from the Scripts menu. Without creating a new Navigation record, the "Tabs Control" script will take you to the Navigation record for the LAST layout you were on BEFORE creating the new one. Either duplicate this Navigation record or create a new one. Then have fun correcting/completing it manually. Don't forget to go to all the other existing Navigation records and change them to reflect your new layout. The more layouts you already have, the more time consuming it is to manually alter all the Navigation records. Manually creating and altering Navigation records in AutoNavigator is still much, much faster than building a navigation system without AutoNavigator, but letting AutoNavigator do the work for you is a couple orders of magnitude faster still.

* Vital Note: In order to make tabs appear in your user layouts, you must create one universal relationship from the AutoNavigator table named "Main" to each table that will be directly represented in a user layout ("directly"; table data shown only in a portal does not require this relationship). In the relationship dialog box, connect the field "MainID" from the table "Main" to the serial ID number of each indicated table (preferably the first table occurrence for each table), using a Cartesian (universal) join (the "X" in the define relationships dialog box). You need to do this for the first table occurrence only, for each relevant table.

* Vital Note: Please remember that every time you create a new table in FMP, FMP automatically creates a new table occurrence, and more importantly, a new layout. You must immediately rename this automatically generated layout to "Developer-layoutName", then run the "Developer Layout Change" scrip, or run the risk of confusing AutoNavigator.

9. DELETING USER LAYOUTS:

We recommend you do not delete user layouts, but "repurpose" them instead. Deleting a layout is a simple task if you have only a few layouts, but can become quite tedious if you already have a lot. The more Navigation records you have that reference the layout you wish to delete, the more time consuming will be the task of correcting those records.

The deletion of user layouts and their Navigation records is not automated. To delete a user layout, use the "Tabs Control" script to go to the layout's Navigation record. Delete the Navigation record, then manually correct all Navigation records that refer to it. Return to the layout you wish to delete via the "Tabs Control" script and delete it, then drag a copy of the "gCurrentLayout" field onto the user layout you end up on and type in the user layout's name. Now you MUST immediately run the "Developer Layout Change" script, because AutoNavigator now thinks there are more layouts than there actually are and needs to be corrected. Use the "Developer Layout Change" script ONLY after DELETING a user layout, NEVER before or after CREATING one. These steps should re-orient AutoNavigator correctly.

10. CREATING DEVELOPER LAYOUTS:

ALL DEVELOPER LAYOUT NAMES MUST BEGIN WITH THE WORD: "Developer", and must follow the naming convention described previously. No exceptions. A developer layout is any layout that will have no tabs.

If you create OR delete any developer layout (a developer layout is any layout not within the tab system, such as, hidden search layouts, report printing layouts, etc... ANY LAYOUT THAT WILL HAVE NO TABS ON IT), IMMEDIATELY run the "Developer Layout Change" script after each and every developer layout is created, renamed OR deleted. The "Tabs Control" script decides what to do for you based on the existing number of layouts as stored in the "gCountOfExistingLayouts" field. But if you manually create a layout outside of the AutoNavigator tab system, this setting will be wrong and the "Tabs Control" script will make errors in your Navigation records the next time it is run. This is the reason for the "Developer Layout Change" script. These errors will have to be corrected manually.

If you need a new developer layout and are not already on a developer layout, use the "Tabs Control" script to get to the Navigation records layout ("Developer-Tabs-Control"). Create your new developer layout manually, return to the "Developer-Tabs-Control" layout, then use the "Tabs Control" script again to return to your user layout. (More on this below.)

11. HOW TO THINK ABOUT LAYOUTS:

Think of your solution as containing two separate layout "spaces": user layouts, and developer layouts, with a barrier between.

Your solution may have ONLY the above two classes of layout... User layouts with tab navigation that AutoNavigator has built, and developer layouts the names of which always start with "Developer-" having no tabs and that can be reached by the user only through your scripts. Both classes of layout MUST follow the previously described naming convention (letters and hyphens only).

While working with user layouts you must navigate using your AutoNavigator tab system. Never the Layouts popup.

While working with developer layouts, you must navigate using the Layouts popup, never the developer scripts.

If you are working with user layouts and want to work on a developer layout, use the "Tabs Control" script to get to the "Developer-Tabs-Control" layout. You are now safely in the developer layout "space". (Think of the "Tabs Control" script as the "bridge" that lets you cross over safely between the user layout "space" and the developer layout "space".) Now you can use the Layouts popup to navigate among developer layouts. You may also use the "New Layout/Report..." command from the Layouts menu. When you are ready to go back to work on user layouts, use the Layouts popup to return to the "Developer-Tabs-Control" layout and then use the "Tabs Control" script to move back into the user layout "space". (You may also use the "Go To Tabs Storage" script as a "bridge". Remember to return to the "Developer-Tabs-Storage" layout by way of the Layouts popup when done working on developer layouts and to then use the "Go To Tabs Storage" script to move back into the user layouts "space".)

If you use either of these scripts ("Tabs Control" or "Go To Tabs Storage") while on a developer layout to which they do not pertain, AutoNavigator may become confused. (See the section titled; "WHEN THE SYSTEM IS CONFUSED")

Don't forget the admonition given previously. Whenever you create a new developer layout or delete an existing developer layout, IMMEDIATELY run the "Developer Layout Change" script.

12. CHANGING USER LAYOUT NAMES:

You may change the name of a user layout at any time. But there are some restrictions:

You may change only one word at a time, then run the "Tabs Control" script to propagate the name change throughout your Navigation records. The word you changed will be altered on EVERY Navigation record, including those not in the hierarchy in which you are currently working. (This is why you should avoid using the same word twice in naming layouts.)

Unfortunately, scripting cannot affect the names of the layouts themselves, so after "Tabs Control" has finished changing your Navigation records, you must manually change the name of every other layout in the hierarchy containing the word you just replaced. AutoNavigator will have done the hard part for you, but you will have to do a little work yourself. If you plan your database carefully, you should not have to worry about changing layout names.

DO NOT run the "Tabs Control" script during the process of manually changing the names of the other layouts.

To manually change a number of layout names, you will of course have to use the Layouts popup menu, which we've admonished you not to do. (After changing a word used in a layout name, AutoNavigator's navigation system will not work until the actual names of the layouts have been changed to match the changes made to your Navigation records.) When you have finished changing the necessary layout names, go back to the layout you originally started from. AutoNavigator will now work again. (Or, run the "Reset" script, or drag a copy of the "gCurrentLayout" field onto any user layout and enter the name of that layout into the field. Then delete the field from the layout.)

13. GETTING STARTED & UNDERSTANDING THE HIERARCHY:

There is a maximum of 20 tabs at each level of the hierarchy, 140 tabs per layout. But each tab does not necessarily represent a layout at that level and you cannot navigate to 140 layouts from every layout! Here's why:

Let's presume that you want to build a system with four tabs at the top level...

Starting with the one layout already built into AutoNavigator ("Untitled"), you will first wish to rename it; let's say to "Home".

Make a duplicate of the AutoNavigator2.0v1 file and open that duplicate now. It will open to the "Untitled" layout. Read the text box there first. Go to layout mode, delete the text box, then select the currently existing tab set (it is 7 tabs long, and therefore too long for our purposes). If the "Object Size" floating window is not there, open it, then note the position of the tab set and delete it. Run the "Copy Tab Set" script (Command-2, Mac) (Control-2, Windows), and enter "1" for the level and "4" for the number of tabs you want in the set. Reenter Layout mode and paste the new tab set. Position it where the deleted set was located and send it to the back. Now change the name of the "Untitled" layout to "Home" (no quotes). Run the "Tabs Control" script immediately. You will be asked if you wish to view the Navigation record or return to the layout. Select "Return". (If you click the "Review" response by mistake, just run the "Tabs Control" script again to get back to your user layout.)

CAUTION: Remember NOT to use the Layouts popup built into FileMaker during any part of this process. Navigate ONLY with Tabs and scripts.

Now we need 3 more layouts. Enter layout mode and DUPLICATE the "Home" layout. (DO NOT use the "New Layout/Report..." command in the Layouts menu.) Rename the newly duplicated layout "Contacts"; run the "Tabs Control" script again. You now have 2 layouts, and will find that using the tabs, you can navigate between them. DUPLICATE the "Contacts" layout, rename it "Sales" and run the "Tabs Control" script yet again. Repeat the process for a fourth layout; name it "Inventory". This time click "Review" instead of "Return". Then select "show all records" and take a look at the 4 Navigation records for your 4 layouts. Notice that only fields in the section called "First Level Tabs" have been filled in. There are no tabs yet below the first level on any of your layouts.

Run the "Tabs Control" script again to return to your last created user layout. Now you will see that you can navigate back-and-forth between the four user layouts with ease, using your new tab navigation system. While on any of the layouts, run the "Tabs Control" script again, then again. You will see that you are taken to the Navigation record for each layout you are on, then back to the layout in question. Once a Navigation record has been created for a layout, the "Tabs Control" script only takes you back-and-forth from a layout to its Navigation record. It does not create or alter any records.

Now we have 4 layouts with 4 top-level tabs on each layout. Think of each top-level tab as the patriarch of a family: the top of a hierarchy of tabs/layouts. There is a maximum of 20 hierarchies (20 top-level tabs) available in AutoNavigator.

Now we can move on to the creation of a set of second-level tabs/layouts...

Click on the "Contacts" tab (now that you can navigate within your system). Run the "Copy Tab Set" script (Command-2, Mac) (Control-2, Windows). Enter 2 for second level and 3 for the number of tabs. Go to layout mode. DO NOT duplicate the layout. Paste your new tab set onto the "Contacts" layout and position it wherever you like. RENAME the existing Contacts layout to: "Contacts-Business", appending the hyphen and the word "Business". Run the "Tabs Control" script and select "Review" to examine

your layout's Navigation record. Since this Navigation record already existed (it was created when you built your initial four layouts), AutoNavigator has simply altered it to reflect the change from a first-level layout to a second-level layout. You will see an entry for one tab in the "Second Level Tabs" area. Now examine the Layout Name fields at both the second level and first level. Notice that BOTH fields contain the layout name "Contacts-Business". This is because both tabs now represent the same (one) layout. You have turned your first-level "Contacts" layout into a second-level "Contacts-Business" layout. There no longer exists a first-level layout for this hierarchy; only a first level tab. For each tab that has related tabs lower in the hierarchy, there is a default tab one level lower and the corresponding layout is also moved one level lower in the hierarchy. This concept is a little difficult to grasp so, study your layouts and Navigation records until you "get it".

Now return to your new second-level layout (remember NOT to use the Layouts popup, use the "Tabs Control" script). We are now going to create a second tab and layout at the second level. Enter Layout mode and duplicate the "Contacts-Business" layout. Change the name from "Contacts-Business..." to "Contacts-Personal", run the "Tabs Control" script and Return to your new layout. Enter layout mode again, again duplicate the layout and change the name to "Contacts-Other". Run the "Tabs Control" script once more. Return to your layouts. Now you have a total of 6 user layouts (not 7). Remember that the Contacts layout became the second-level "Contacts-Business" layout when you appended to its name and ran the "Tabs Control" script. By appending to a layout name instead of changing it, you are telling AutoNavigator that you want to create a new lower level set of tabs/layouts.

Now that you have an idea how it is done, how long do you suppose it would take you to start over with a fresh copy of AutoNavigator and build the same 6 layout, 2 level navigation system you just made? A minute? Two? How long would it take you to do the same thing from scratch without using AutoNavigator, starting with a blank FMP file...? (Hint: use a calendar, not a watch.) ... You're welcome!

And don't forget, when building layouts and their navigation system without AutoNavigator, the more layouts you already have, the more tedious and time-consuming it is to add just one more layout. With AutoNavigator, the 200th layout is just as simple to add as the 2nd.

Now select two or three layouts and add some of FileMaker's built-in tab objects to each. Nest several layers of tab objects within tab objects, being sure to name each tab pane in the "Object Info" dialog. (Naming objects is only possible in version 8.5 of FileMaker Pro™; see "19. CONTROLLING FILEMAKER TAB OBJECTS" below.)

Now close the AutoNavigator2.0v1 file you've been using. Throw it away if you wish and open a duplicate of the AutoNavigatorSample1.0v4 file (demo files were created in an older version of AutoNavigator). This is a file with 50+ layouts already built in. The purpose of this file is tutorial in nature. You can explore this already built layout/navigation set to see how it works and practice adding some of your own layouts.

Click the "Companies" tab. You will see 2 tabs at the second level, plus one each at the third and fourth levels, and 2 at the fifth level. Though you see 7 tabs on this layout (not counting the other 6 top level tabs aside from "Companies"), they represent a total of only three layouts.

Use the "Tabs Control" script to go to the Navigation records layout, enter Find mode and type "Companies" into the "Layout_Name" field at the top. Run the search. Notice that only 3 records are found for the "Company" hierarchy. There are only 3 layouts in this hierarchy, despite the 7 tabs. Use the "Tabs Control" script again to return to the "Companies" layout.

Here's what happened:

We started with a "Companies" layout, then decided we needed additional "child" layouts under "Companies", and therefore a lower level tab set. Since any tab having lower level tabs under it requires a default lower level tab/layout, the layout which that original tab represents, must now become a lower level layout. So we turned "Companies" from a first-level layout to a second-level layout by appending "-Parent" to its name and running the "Tabs Control" script. Then we created the "Companies-Uncle" layout by duplicating the "Companies-Parent" layout, renaming the duplicate and again running the "Tabs Control" script. When we wanted a third level of layouts and tabs, one of the second level layouts had to become the default layout for the third level. We chose the same layout as before and "Companies-Parent" became "Companies-Parent-Child". Then for the fourth level it was changed to "Companies-Parent-Child-Grandchild" and finally to "Companies-Parent-Child-Grandchild-GrtGrndSon" for the fifth level. The same single layout kept moving down in the hierarchy as we needed to build additional levels into our tab navigation system.

The three actual layouts represented by these tabs are:

"Companies-Parent-Child-Grandchild-GrtGrndSon"
"Companies-Parent-Child-Grandchild-GrtGrndDghtr"
"Companies-Uncle"

Be sure you understand. Look at the tabs. There is no first level layout. Only one second level layout("Companies-Uncle"). No third or fourth level layouts, and 2 layouts at the fifth level("...-GrtGrndSon" and "...-GrtGrndDghtr")... Even though there are tabs at all levels.

Now make sure you are on the "Companies-Parent-Child-Grandchild-GrtGrndSon" layout, and run the "Tabs Control" script so you can view the Navigation Record for this layout. Follow the white tabs down the page and look at the layout name at each level. You will notice that the name is exactly the same at every level of tabs; "Companies-Parent-Child-Grandchild-GrtGrndSon". Study this carefully to fully understand what is

going on. Then use the "Tabs Control" script to go back to your user layout to see if it makes more sense.

Of course, what we have built here under the "Companies" hierarchy doesn't make a lot of sense from a practical standpoint. There would be no reason to move a layout all the way down to the fifth level when there are no third and fourth level layouts, but it is a good way to demonstrate how the hierarchy works. Each successively lower tab set must have one tab that is the default for the corresponding tab in the next level up. In other words, if you click on "Companies" and it has tab levels/layouts "beneath" it, one of them must be the default layout or the "Companies" hierarchy would have nothing to show you. If you have previously built multi-level tab navigation systems then you already know they all must work this way, but this fact does escape a lot of people's attention, even after they've built one.

14. WHEN THE SYSTEM IS CONFUSED:

If the Navigation system is not working correctly, one of the following fields may have an incorrect entry:

In the "Main" table:

- gCurrentLayout
- gTabsControlLastLayout
- gTabsStorageLastLayout

In the "Navigation" table:

- Layout_Name

These errors most commonly result from building Navigation records manually (mistaken data entry or typos), or using the Layouts popup when you shouldn't and then running the "Tabs Control" script, or building Developer layouts without following the instructions given previously.

The Navigation system built into AutoNavigator is quite robust and virtually impossible for the end-user to damage or break if he is locked out of layout mode and the script editor, and has no access to the Layouts popup. But during the process of development it is possible to misstep and throw the system off by using the navigation system some of the time, and the Layouts popup other times. Using the Layouts popup is a hard habit to break, and you are likely to take a while to grow accustomed to not using it when working on user layouts.

First, don't panic. Any mistake you make can be fixed. Second, make frequent backups while developing. If for some reason you should damage your Navigation records to the point that it would take too much time to repair them manually, just delete them, and import a clean set of Navigation records from your last backup; and if necessary,

manually reenter any Navigation records that were added since that backup. (A trick that requires a little care: if you had to reimport your Navigation records and had say, 3 new user layouts that your backup had no Navigation records for; drag a copy of the field "gCountOfExistingLayouts" onto THE LAYOUT YOU COPIED to make the first of those 3 layouts (be sure to get there using the tab system) and enter a number that is 1 less than the number contained in the field. Go to layout mode and use the Layouts popup to transfer to the first of the 3 new layouts. This will make AutoNavigator think you just created the layout. Now run the "Tabs Control" script. This should rebuild your Navigation record. Repeat the process for the other 2 new layouts.)

CAUTION: It is strongly advised that when you decide to rename a layout in a solution containing more than a dozen or so user layouts, you make a temporary copy of the solution before making the changes. This way you have a backup in case of error.

If you find you have inadvertently confused the system (no tabs showing, or incorrect tab colors or tab labels showing in the wrong color), run the "Reset" script, or drag a copy of the field "gCurrentLayout" onto your layout and type the name of the current user layout into it. Your tabs should now appear correctly. If they do not, use the "Tabs Control" script (or the Layouts popup if the script won't work) to go to the Navigation record for your layout and make sure the entries for the layout are correct.

If you are on the "Developer-Tabs-Control" layout, and AutoNavigator won't return you to your last user layout. Drag a copy of the "gTabsControlLastLayout" field onto your layout (DO NOT RUN THE RESET SCRIPT), type the name of the user layout you wish to return to into the field. Then DELETE THE "gTabsControlLastLayout" FIELD FROM THE LAYOUT and run the "Tabs Control" script again. Deleting the "gTabsControlLastLayout" field from the layout first is important! If you don't, the next time you run the "Tabs Control" script for a new or altered layout, the script will change the contents of this field too and you will have to correct it manually once again.

If you are on a user layout and the "Tabs Control" script will not take you to your Navigation record, this means that the contents of the "gCurrentLayout" field do not match the foreign key field "Layout_Name" in the Navigation record. Check this field, then go to the Navigation record manually and change it if necessary. Return to your user layout, also manually, and try again.

If you are on a user layout and one of the tabs does not work, this means that the entry for that tab in the layout's Navigation record is incorrect (It may also mean that the wrong button is under the tab, though this is unlikely). Use the "Tabs Control" script to go the Navigation record and correct the entry.

15. SUBSTITUTING YOUR OWN TAB GRAPHICS:

You can use any graphics you like for your tabs. To substitute yours for ours, just go to the "Developer-Tabs-Control" layout using the "Tabs Control" script. There at the bottom of the layout are the 7 repeating fields (3 repetitions each) that store the graphics. Paste or drag and drop your graphics into the fields. Then return to your user layouts using the "Tabs Control" script to see how your new graphics look.

You can use a different set of graphics for each hierarchy level. (A little imagination and a few changes to the code would allow you an even broader set of options.)

The only difficulty you may encounter is the possibility that your tab graphics may not display well in the size fields we used to create the included AutoNavigator tab sets, because yours are smaller or larger. This is simple to fix. Adjust the size of one of the tab sets (displaying your graphics instead of ours) on a user layout, until they look right. Note the changes you had to make. Then use the "Go To Tabs Storage" script to go to the "Developer-Tabs-Storage" layout where the tab sets are stored. There are 133 tab sets there (19 sets for each of the 7 hierarchy levels). Adjust the size of each of these (you will have to ungroup each, then regroup them when you are done; don't forget to also resize and regroup the invisible buttons behind them). (A simpler, less tedious approach is to alter only the longest sets you expect to need for each of the seven levels, then always use those sets and trim them to the required length in your layouts. This way you only have to alter 7 instead of 133 tab sets.)

Normally, the adjustment you need to make is as follows: Use the "Object Size" floating window to get the size of your tab graphic(s). Set the repeating fields to exactly the width of your graphic times the number of repetitions, plus two pixels; set the height to exactly two pixels taller than your graphic. Start with the tab sets that are 10 repetitions long. Then for the succeeding sets, just set the number of repetitions to one less successively. The fields will automatically be sized correctly (+ 2 pixels longer, and + 2 pixels taller.)

There is one more step. Remember that the "Developer-Tabs-Storage" layout is for manual retrieval of tab sets. You'll probably want to be able to continue using the very convenient "Copy Tab Set" script, but the tab sets stored in the fields on the layout which that script accesses ("Developer-Tabs-Containers") still contain our original tab sets, not yours. So:

While still on the "Developer-Tabs-Storage" layout, in layout mode, copy the first group (First Level Tabs, 20 through 2) to the clipboard. Then use the Layouts popup (you're in the developer layout space now, remember) to go to the "Developer-Tabs-Containers" layout. Paste the group of tab sets you just copied onto the layout underneath the fields you find there, in layout mode. Cut the first tab set (20 repetitions) to the clipboard, enter Browse mode and paste it into Level One, Repetition 20. Switch back and forth between

layout and browse mode, cutting one tab set at a time and pasting it into succeeding lower repetitions of the field. When they are all gone, go back to the "Developer-Tabs-Storage" layout, copy another set and return to paste them into the repetitions of Level Two, etc. OR, open a second window and drag and drop between "Developer-Tabs-Storage" in layout mode and "Developer-Tabs-Containers" in Browse mode. Don't forget to close the second window when done.

When all done, return to the "Developer-Tabs-Storage" layout using the Layouts popup, then use the "Go To Tabs Storage" script to move back into the user layout space.

In the future, if you need tab sets of this size again, don't repeat the same tedium. Drag and drop (or import) the contents of the fields on the "Developer-Tabs-Containers" layout into a fresh copy of AutoNavigator and you're set to go.

16. BUILDING A NAVIGATION SYSTEM AUTOMATICALLY

AutoNavigator will be happy to build your tab and navigation system for you on the fly. You do not need to understand the inner workings of AutoNavigator in order to use it for even the largest tab and navigation system. If you follow instructions carefully, you will probably never even have to look at a Navigation record, much less edit one.

To make the process as simple and effortless as possible, before starting to create new layouts make sure all of your basic elements are in place. Substitute your tab graphics for ours, if that is what you want, and make sure all the tab sets have been resized (see above) to fit your graphics. Place all of the recurring elements you know will appear on every (or most) of your layout(s) on the original "Untitled" layout that comes with AutoNavigator. It is easier to delete an element that is not needed on a particular layout than it is to copy and paste between layouts. Resize the layout to fit your needs and make certain everything is precisely positioned. If you know how many levels of tabs and the length of tab sets you will need in a particular hierarchy, place those also. Again, it is simpler to delete a tab set from a layout than to add and position tab sets on a number of layouts.

When your basic layout design is complete, you can start creating new layouts.

Rename the "Untitled" layout and run the "Tabs Control" script.

For each additional first-level tab and layout you need, duplicate the last first-level layout, rename the duplicate and run the "Tabs Control" script. Repeat this until you have all the tabs/layouts you need at the first level of the hierarchy.

When you are ready to build lower levels in any of the hierarchies, use your new tabs to navigate to the layout you want to build under.

Use the "Copy Tab Set" script to grab tab sets if you haven't already placed them. Position them.

For the FIRST LAYOUT at the new level, append a hyphen and the word you have chosen to identify this second-level layout to the name of the existing layout (remember, you are converting a first level layout into a second level layout and the number of words in the layout name indicates its position in the hierarchy: DO NOT DUPLICATE THE LAYOUT. The existing first-level layout BECOMES the new second level layout). Run the "Tabs Control" script.

For additional tabs/layouts at this second level, DUPLICATE your newest second-level layout. Change only the last word in the name to the one you want to use, and run the "Tabs Control" script again.

Repeat the above process to add third, fourth, fifth, sixth and seventh levels to your hierarchy.

The foregoing is all just this simple: If you want to start a new lower level of layouts, append a hyphen and another word to the name of the existing layout under which you want the new level to appear and run the "Tabs Control" script. If you want to add a new layout to an existing level of layouts/tabs, duplicate one of the layouts at that level, change the last word in its name and run the "Tabs Control" script.

CAUTION: If you wish to create a new first-level layout (this is true at all levels), do not duplicate a layout with two or more words in its name. This is NOT a first level layout, even though it may have a tab at the first level. Duplicate only a layout with the same number of words as the level at which you want the new layout to appear. For example: if you have 3 first-level tabs; "Contacts-Business", "Invoices" and "Reports", and you want a new first-level layout "Settings", do not duplicate the "Contacts-Business" layout. This is a SECOND-LEVEL layout! Duplicate either "Invoices" or "Reports".

(For Developer layout creation, see the appropriate section above.)

Make frequent backups. It may be easier to revert to a backup than to correct a mistake by editing Navigation records if your system is already complex.

17. BUILDING A NAVIGATION SYSTEM MANUALLY

AutoNavigator is set up so that if you hold down the "Shift" key when you run the "Tabs Control" script, you will be taken to the "Developer-Tabs-Control" layout without creating a new Navigation record. You will be taken to the Navigation record for the user layout you were on if it is not new, or the last user layout you were on before creating a new layout. AutoNavigator will NOT create or alter any Navigation records for you. You may now manually edit/duplicate existing Navigation records or add new ones.

CAUTION: If you have created a new user layout and then bypass AutoNavigator's automated Navigation record building routines by using the "Shift" key while running the

"Tabs Control" script, you CANNOT go back and get AutoNavigator to build a new Navigation record for that layout automatically.

You will see 7 areas on the "Developer-Tabs-Control" layout titled "XXXX Level Tabs". These are the fields for each level of tabs possible on your user layout. Each level shows four fields (20 repetitions each) labeled, "Tab Label", "Layout Name", "Select Tab Graphic" and "Preview Tab Graphic".

In the "Tab Label" field, type in the Label you wish to appear on the tab. In the "Layout Name" field, enter the name of the layout to which that tab should navigate. Then click on the field "Select Tab Graphic" until the tab graphic you want appears in the field below it (in the "Preview Tab Graphic" field). (You may have to click outside the "Select Tab Graphic" field for the tab graphic to appear.)

Each tab position can be represented by one of three different graphics (look at the "Tab Graphics Sets" section at the bottom of the "Developer-Tabs-Control" layout) :

Active Tab this tab can be clicked to go to another layout
Selected Tab this tab represents the layout you are on; clicking it will have no effect
Inactive Tab this tab represents an existing layout you are not allowing your user to navigate to (preferable to hiding tabs). There is no method in AutoNavigator for changing an Active Tab to an Inactive Tab, or vice versa, on the fly. You will have to code this yourself.

Be very careful to enter data correctly into the "Layout Name" field. Think of the hierarchy. A tab that has layouts lower in the hierarchy will have a multiple-word, hyphenated name longer than its level would indicate. The number of words indicates at what level in the hierarchy the layout for that tab actually exists. Study existing Navigation records in the accompanying "AutoNavigatorSample1.0v4.fp7" file to see how this works. It is helpful to keep in mind that the Navigation record is a diagram of the existing tabs on a layout and what they are supposed to do. It is not a picture of the layout, but of where the tab system on that layout can navigate to!

Getting the layout names and Navigation records correct will prove a little frustrating until you get accustomed to making ALL necessary changes. For example:

If you start your solution with a set of first level tabs that represents layouts named, for example; "Home", "Reports" and "Preferences"... your layout names and Navigation records will be quite simple and straightforward. After all, you have only one tab level, but...

Let's say that now you wish to add some second-level layouts under the "Home" tab. You'll add "List" and "Detail" layouts. This means that the "Home" layout will no longer exist. Instead of a "Home" layout, there will now be a "Home-List" layout (your original "Home" layout renamed) and a new "Home-Detail" layout. You must now go into the Navigation Records for each layout and change "Home" to "Home-List" in each relevant

layout record. The Navigation record for the "Reports" layout for example, will still show the layout name "Home" for the first tab. This must now be changed to "Home-List", etc. You must also go to the Navigation record for the original "Home" layout and change its name to "Home-List". Most important, you must remember to change the name of the layout itself.

When you have finished manually creating/altering a Navigation record, click the "Reset Navigation" button (this is not the same as the "Reset" developer script that can only be used in the user space). This will wipe the Preferences records (which contain the layout navigation history) clean. Be sure not to use the Layouts popup to return to your user layout. Use the "Tabs Control" script. You can't do any serious damage if you forget this, but your tabs will be temporarily "confused" if you do. Just click several tabs for the system to re-orient itself. (Your end-users will never have this problem because they won't have access to the Developer layouts.) If this doesn't work, use the "Reset" developer script, or go to any user layout, drag a copy of the field "gCurrentLayout" onto it and enter the name of the current layout in Browse mode. Delete the field from your layout and try your tabs again. They should work. If they do not. Check the Navigation record for the layout you are on to be sure all entries are correct.

CAUTION: Be very careful while building layouts and Navigation records manually. Remember that the "Tabs Control" script not only switches you back and forth between your user layouts and Navigation records, but it also checks to see if you have created a new layout, and builds a Navigation record for you automatically if you have (unless you hold down the "Shift" key while running the script). It is VERY IMPORTANT to click the "Reset Navigation" button when you are finished creating/altering Navigation records manually. This script also runs the "Developer Layout Change" script as a subscript and will keep AutoNavigator from creating new Navigation records you don't want. If you create your new layouts AFTER you create Navigation records for them, run the "Developer Layout Change" script after creating each new layout in addition to clicking the "Reset Navigation" button after altering/adding Navigation records.

(For Developer layout creation, see the appropriate section above.)

Make frequent backups. It may be easier to revert to a backup than to correct a mistake by editing Navigation records if your system is already complicated.

18. PRE- & POST- TAB SCRIPTS:

AutoNavigator now has built-in pre- and post- tab scripts called near the beginning and end of all 7 Tab Scripts (). To activate a pre- or post- tab script, just remove one step from the script and add your own code where appropriate.

19. CONTROLLING FILEMAKER TAB OBJECTS:

CAUTION: This feature will only work with FileMaker Pro™ 8.5 and above. If you have purchased AutoNavigator 2.0v1 and find you need to develop a project in FileMaker 7 or 8, just let us know and we will either provide you with a copy of AutoNavigator 1.0v4 at no charge, or give you instructions for turning this feature off in AutoNavigator 2.0v1.

Most developers using AutoNavigator are also using the very convenient tab object introduced in FileMaker Pro™ 8.0. This is a very handy tool when multiple field sets are required but an additional layout would be overkill. However, these FileMaker tab objects have a serious failing: when you return to a layout containing a tab object(s), you generally find the tab panels that were frontmost when you left are no longer frontmost on your return.

With the addition of three (3) built-in custom functions, fifteen (15) new fields and two (2) new scripts (along with some changes to existing scripts) in AutoNavigator 2.0v1, coupled with the new ability in FileMaker Pro™ version 8.5 and above to name objects on your layouts (in particular, FileMaker tab object tab panels), AutoNavigator can now automatically bring the last FileMaker tab panel(s) forward when you return to any AutoNavigator user layout. Your users will find upon returning to any layout, that all of the tab panels that were frontmost on the layout when they left it, are still frontmost when they return.

It doesn't matter how many tab objects you have on a layout, or how many levels deep tab objects are nested (tab objects inside tab objects), AutoNavigator will take you back to exactly where you left off on every layout. And, AutoNavigator will remember the last tab panels (in addition to the last layout) for each user when they log back in to your solution.

All you have to do to take advantage of this new AutoNavigator feature is to simply name each tab panel (in the "Object Info" dialog) on every FileMaker tab object you create. That's it. You do not need to do any programming, create or set any fields... absolutely nothing! Just name your FileMaker tab panels and you're done. You can even re-use the same names on any other layouts and AutoNavigator will not be confused.

FileMaker tab objects also cause problems when non-navigational scripts have to temporarily navigate to other layouts behind the scenes, such as for printing or creating a new record in a related table. Upon return to the original layout, users find different tab panes are brought forward. The new tab object control feature in AutoNavigator can handle this too. The two new scripts that are key to the functioning of AutoNavigator's ability to control FileMaker tab objects ("SetFMptabObjectHistory" and "GetFMptabObjectHistory") can be called at the beginning and end, respectively, of any

script you write and will automatically make sure your users always see the same tab panels they previously selected, all the time.

Finally, most of the work performed by AutoNavigator in controlling FileMaker tab objects is performed by three (3) custom functions. These routines are lightning fast and therefore there is no practical limit to the number of layouts and tab objects it can handle.

20. PRECAUTIONARY NOTES:

AutoNavigator is designed to do almost all the work for you, but because it is a developer tool and you must therefore have developer access, it is NOT foolproof. You CAN create quite a mess for yourself if you are careless and undisciplined. When you create a new layout and use the "Tabs Control" script to create a matching Navigation record (each layout must have a Navigation record), AutoNavigator will do its best to build an accurate Navigation record for you and propagate changes throughout the other Navigation records. But you must do your part: build your layouts in a rational, top-down order; first top-level layouts, then 2nd level, then 3rd, etc.

Consider each tab at the top level to be a separate hierarchy. You can build all or part of any hierarchy at any time. For example, you can build the second and third level of tabs/layouts for the first (top-level) tab hierarchy. Then you can go build levels 2 through 5 for the fourth (top-level) tab hierarchy. You do not have to build all second-level tab/layout sets in all hierarchies before you can build a third-level tab/layout set in any hierarchy. But you must work from the top down in each hierarchy, and not skip levels.

It is **STRONGLY** recommended that you follow the advice given in every book ever published on building databases (and software programming in general)... "Plan your database on PAPER first." THEN begin to code in AutoNavigator. Build your layouts in an orderly, top-down fashion. Do not attempt to build a 3rd (or 4th) level layout until you already have a 1st, 2nd (and 3rd) level above it (AutoNavigator won't let you do this automatically). Always create layouts by making a duplicate of an existing layout in the same hierarchy. In other words, if you need a "Father-Son-Granddaughter" layout(3rd level), go to the "Father-Son-Grandson" layout and duplicate it. Do NOT create a duplicate of the "Uncle-Nephew-Grandnephew" layout in order to create a "Father-Son-Granddaughter" layout. AutoNavigator WILL let you do it, but your Navigation records will be thoroughly incorrect and you will have to edit many of them manually!

DO NOT allow your end-users to navigate between layouts using anything other than your tab sets and AutoNavigator's scripts. Make certain your end-users do not have access to the Layouts popup menu, Layout Mode, or the Script Editor. You probably already know from experience that these things are not wise, but it doesn't hurt to remind you.

The Startup and Close scripts contain steps that are vital to the proper operation of AutoNavigator. Do not delete them or alter the existing steps. You will no doubt wish to add steps unique to your solution to these scripts, but be careful where you place your additions.

BACKUP, BACKUP, BACKUP... Ignore this advice and you WILL regret it. In addition to regular and robust backup schemes to protect your data after deployment and your database structure during development (ALL databases crash and burn sooner or later), make additional temporary copies of your file(s) during development (in addition to standard "builds") every time you make changes to AutoNavigator layouts or Navigation records that are "risky", such as changing the name of a user layout when you already have dozens of user layouts and Navigation records. It is so much easier to make a quick copy and revert to it if necessary, rather than having to manually change dozens of Navigation records. Damaging an AutoNavigator file during development to the point of being beyond repair is extremely unlikely, but having to manually correct dozens of Navigation records can happen if you make a mistake. Save yourself a lot of potentially tedious manual editing.

END
