

# Randolph Does Do Windows

By William Baker

**Editor's note:** The ResBuilder™ and GUIBuilder™ files discussed in this article are available for download at <ftp://ftp.basis.com/pub/advantage/v2n4/prefs.exe>

"Randolph," I said, "Months ago, I predicted you'd be well advanced in GUIBuilder by winter. Now it's ready to snow--have you made any progress, or are you going to destroy my reputation for predicting the future?"

"I've made good progress but not the sort you want to hear about," Randolph answered. "I had almost succeeded in forgetting about GUIBuilder until you mentioned it. I remember you showed me a program that displayed a button and a 'Hello, world' message. I suppose since then you've done something more interesting?"

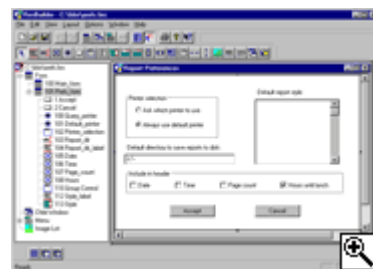
"I decided the next thing I would tackle after a 'Hello, world' program is one of those ubiquitous Windows preferences screens."

"You quit what?"

"I didn't say I quit; I said, 'ubiquitous'--you see preferences screens in Windows applications all over the place."

"Oh. I suppose you're going to say that I have to put preferences screens all over the place since you see them everywhere."

"Absolutely. Look at this screen in ResBuilder." I said while pointing. "You've seen preferences screens like this, haven't you? This preferences dialog will be invoked from the applications main menu. I created the main form with nothing on it but a tiny menu, just to get started. The top-level menu says View, and the only option under it is Preferences. When the user clicks the Preferences menu, the preferences form comes up. You can see that I'm working on preferences for creating reports. I have different kinds of controls for the different types of information the user can enter. There are radio buttons to pick the printer selection method, a text edit control for the default path for reports to disk, checkboxes for what fields to include in the headings, and a list box to select the default report style."



*Preferences screen in ResBuilder.*

"I'm amazed that someone as serious as you would put 'Hours until lunch' as a default report header field," Randolph said.

"That's just to keep the audience amused."

"Audience? What audience?"

"Don't worry, I just mean whoever is reading the program."

"Oh."

"The key to getting information to and from dialogs in GUIBuilder is to use the screen-handling functions that are included with the product. These eliminate a lot of the tedium of setting controls one at a time with mnemonics and querying controls one at a time with CTRL() or SENDMSG() functions. These functions work with a template for each form, so in my initialization section I set up my own template to hold the user preferences, and I structured it as closely as possible to the form template."

```
rem ' -----
rem ' Init
rem ' -----

rem Hide SYSWINDOW
print 'hide'

rem prefs$ string template stores user preferences. Its structure
rem is similar to prefs_form$ template that is used to pass values
rem to and from the form.

dim prefs$:"query_printer:n(1*=0),default_printer:n(1*=0),"+
: "report_dir:c(64*=0),date:n(1*=0),time:n(1*=0),"+
: "page_count:n(1*=0),hours:n(1*=0),style:c(255*=0),"+
: "default_style:n(1*=0)"

rem Set defaults. In real life, these defaults would probably
rem be read from a file.
prefs.query_printer=1
prefs.default_printer=0
prefs.report_dir$="c:\"
prefs.date=0
prefs.time=0
prefs.page_count=0
prefs.hours=1
prefs.style$="Regular report"+$0a$+"Full detail report"+$0a$+
: "Graphs"+$0a$+"Summary report by region"+$0a$+"Summary report"+
: " by state"+$0a$
prefs.default_style=0
```

"Then I wrote a section of code to be executed when the user selects Preferences from the menu. At that point, I want to switch from the main form to the preferences form and then use a function to set up the `prefs_form$` string template, which will be used to pass values to and from the form."

```
rem ' Menu selection made

rem ' Switch to preferences form
gb_win_id$="101"
gb_context=fngb_focus_win_id(gb_win_id$)
print (gb_sysgui)'show'(0),'raise','focus'(0)

rem ' Set up string template for passing data to and from form
dim prefs_form$:fngb_template$(gb_win_id$)
```

"Then each field in the `prefs_form$` template gets set with the corresponding value in the `prefs$` template. There is a handy utility program with GUIBuilder called `gb_rec` that copies fields of the same name from one template to another. Using this utility saves me from having to write a bunch of statements to set each field one at a time:"

```
rem ' Initialize template with default values stored in prefs$

call "gb_rec",prefs$,prefs_form$,"";
rem ' copy fields from prefs to prefs_form

prefs_form.report_dir_label$="Default directory to save reports to:"
prefs_form.style_label$="Default report style:"
```

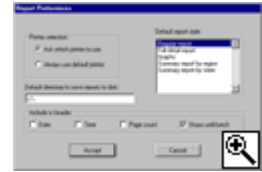
"And now the payoff--I can manipulate all the controls on the form with this one function:"

```
rem ' Use function to display values on screen
prefs_form$=fngb_put_screen$(gb_win_id$,prefs_form$)
```

"List boxes are a special case. The Put Screen function doesn't highlight a selected item, it just displays a list of items in the list box. So we have to use a mnemonic to highlight an item in the list box. Note the neat trick of querying the `prefs_form$` template to get the control ID of the list box:"

```
rem ' fngb_put_screen won't handle highlighting list box item,
rem ' so this must be done with a mnemonic.
control_id$=fattr(prefs_form$,"STYLE","ID")
print (gb_sysgui)'listsel'(num(control_id$),prefs.default_style)
```

"So now I have the code to display the preferences form and the correct values in it. So really all that's left to deal with is how to get out of the form. For that, I provided a Cancel button and an Accept button. When the user hits the Cancel button, all we have to do is hide the preferences form because we can ignore the user's input. When the user hits the Accept button, we query the values entered in the form with another function. Once again, the list box has to be handled as a special case: we use the CTRL() function to find out which item the user selected."



*Preferences screen with default values.*

```
rem ' Push button operated
prefs_form$=fngb_get_screen$(gb_win_id$,prefs_form$)
rem ' Copy fields from prefs_form to prefs
call "gb_rec",prefs_form$,prefs$,""
rem ' Store index of report format selected in list box
control_id$=fattr(prefs_form$,"STYLE","ID")
prefs.default_style=dec(ctrl(gb_sysgui,num(control_id$),2))
rem ' Hide this form and return to main form
print (gb_sysgui)'hide'(0)
gb_win_id$="100"
gb_context=fngb_focus_win_id(gb_win_id$)
```

Randolph was skeptical that all this code to get eight pieces of information from a user was really saving me any work. "Well, I could show you how much work it would be to write the whole program at the console without GUIBuilder," I replied, "but that would surely make you fall asleep, and besides, it's zero hours until lunch."

