BPLUS (6): BASIC Plus Applications

v2.4 / 6 of 18 / 01 feb 99 / greg goebel / public domain / bp06_app

* The BPlus product includes an applications environment that runs in the HP BASIC environment. It features an "Application Manager" that allows use of several built-in applications. This chapter describes the applications environment and the standard applications in detail.

* Contents:

[1] THE APPLICATIONS ENVIRONMENT & APPLICATIONS MANAGER

* The BASIC Plus applications environment provides access to the following standard applications:

- Notepad editor.
- Clock.
- Help utility.
- Help compiler.
- Screen Builder.

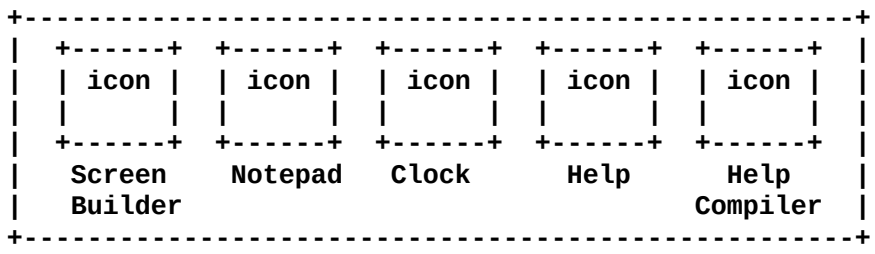
There was no applications environment in BPlus before the 2.0 revision, and the only application that could be used was an earlier and different version of the Help utility. The applications environment also does not apply, as mentioned in earlier chapters, to HBW; the only application supported is the Screen Builder (HBW online Help uses the standard MS-Win Help utility).

* If you have BPlus installed, and you enter at the RMB prompt:

```
APP
```

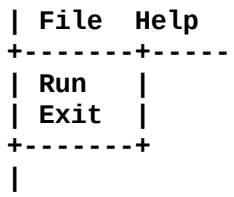
-- then the standard RMB display will go away to be replaced with a gray background, with the following panel floating in the upper-left corner:

```
+---+-----------------------------------------------+
| = |             Application Manager               |
+---+-----------------------------------------------+
| File  Help                                        |
```



This is the Application Manager; it is somewhat similar in concept to (but simpler than) MS-Windows Program Manager in that it can be used to run programs identified at sets of icons.

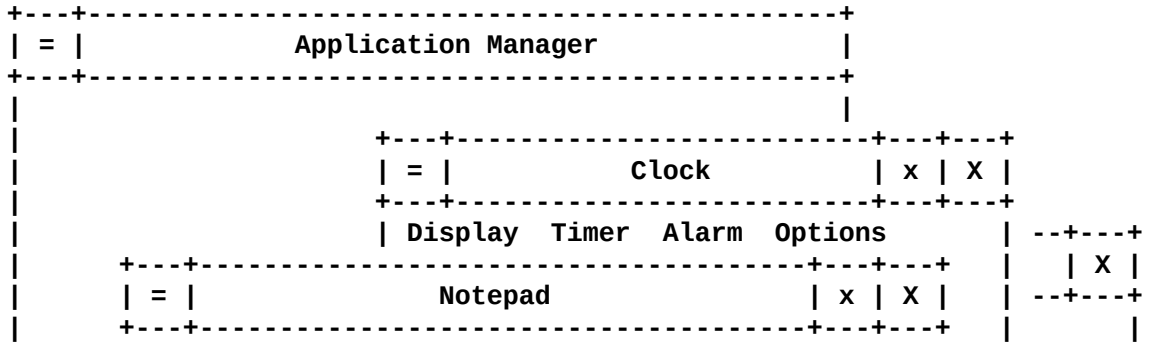
There isn't much to using it; you can either double-click on an icon to run the application, or run it from the "File" menu:

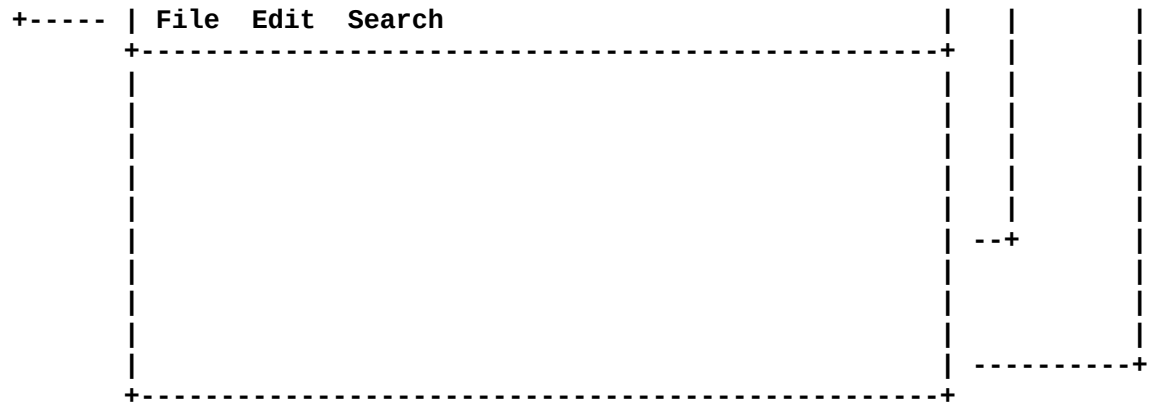


-- as well as bail out of the applications environment and return to the standard RMB environment. The other menu, "Help":

-- can also be used to bring up the Help utility, as well as identify the rev level of the Application Manager.

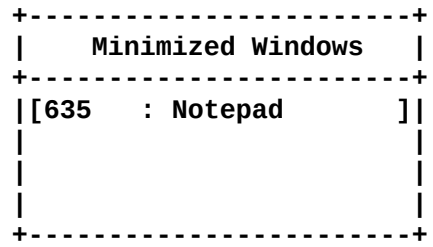
* Anyway, you can use the application manager to run the BPlus apps, and you can run multiple copies of them:





You can "maximize" an individual app by clicking on the first button from the right; this expands the app to full screen. You can reduce it back to "normal" size by clicking it again.

You can "minimize" indivdual apps by clicking on the second button from the right; this reduces the app to an entry in a list box, the Minimized Windows Manager (which is actually part of the Application Manager and not, strictly speaking, an app itself, though it's convenient to think of it as one). For example, if we iconize the Notpad editor in the illustration above, we get:



Note that apps also have a "toaster box" in the upper-left corner; usually this just gives a short menu to allow you to conveniently close the app.

* Now for some fine details. In the example above, we ran the applications environment and invoked apps from it; but you can actually bypass the environment and run them directly if you like, by invoking "APP" along with the application name:

```
APP "APCLOCK"    Run Clock application.
APP "APHCOMP"    Run Help Compiler.
APP "APMGR"      Run Application Manager.
APP "APNOTEP"    Run NotePad.
APP "APSCRBLD"   Run Screen Builder.
```

Note that you can run Help by simply invoking HELP from the RMB prompt. All these commands can be executed programmatically, allowing you to integrate them into RMB applications ... though about the only app that is really useful in that context is the NotePad editor.

Applications have life of their own, separate from RMB programs -- running, pausing, stopping,

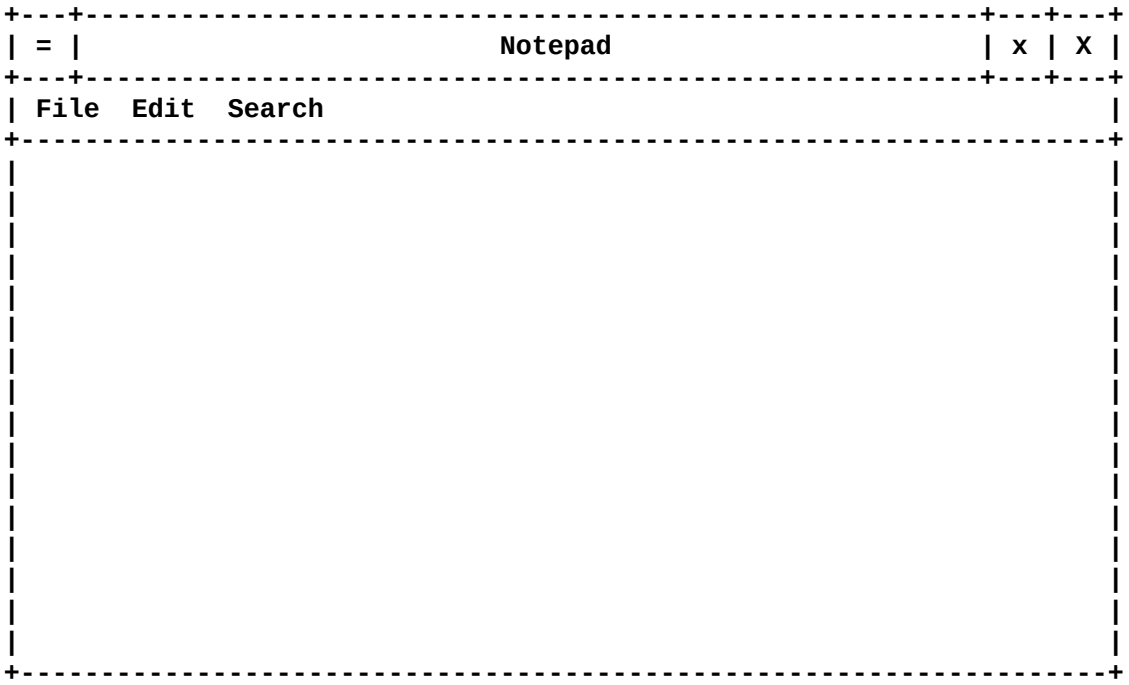
editing, scratching, loading, or getting a program has *no* effect on apps. When apps are active, BPlus "owns" system resources -- the keyboard, display, and so on. An application continues to live until the user closes it, or until the RMB environment undergoes a Shift-reset or a SCRATCH A.

The background color or a tiled bitmap for the application environment can be specified in the CONFIG.

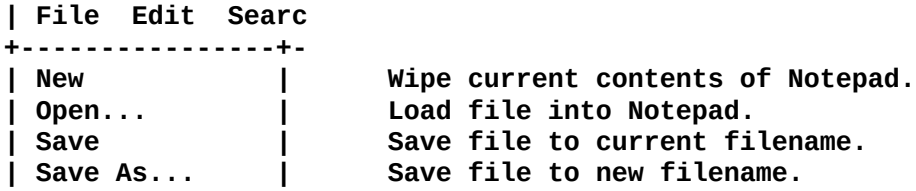
Note that you cannot execute an RMB program from the applications environment; that is basically attempting to run such an RMB program out of a CSUB, and RMB won't let you do that. Apps are written in C code, using MS-Windows-like calls; simple Windows programs can be in theory ported to the RMB application environment, but unfortunately HP does not offer tools to allow users to build their own apps.

[2] NOTEPAD

* The Notepad application is a simple text editor with mouse-selectable cut, copy, paste, and replace functions; you can use the mouse to select text for cutting, copying, and pasting. It has a simple user interface:



The "File" menu provides various file-I/O functions (as well as allows you to quit Notepad):



```
| Read...        |      Merge file into current text.
| Quit           |      Quit Notepad.
+----------------+
|
```

The "Edit" menu is used to perform various operations on text within the editor that has been selected with mouse click-and-drag operations:

```
   Edit  Search
 -+-----------+-
  | Cut       |          Cut text out of editor, put in paste buffer.
  | Copy      |          Copy text from editor and put in paste buffer.
  | Paste     |          Insert paste buffer contents into editor.
  | Replace   |          Replace selected text with paste buffer contents.
  | Clear     |          Cut text out of editor, don't put in paste buffer.
  +-----------+
```

Finally, the "Search" menu provides the functions:

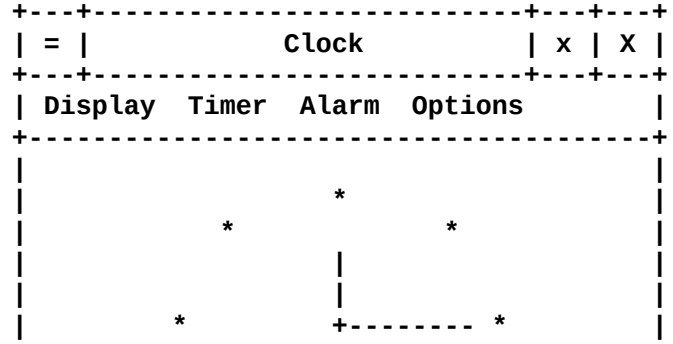
```
   Search
 -+-------------------+-
  | Line Number...    |  Move cursor to designated line number.
  | String...         |  Search for a string.
  | Next occurrence   |  Repeat search.
  +-------------------+
```

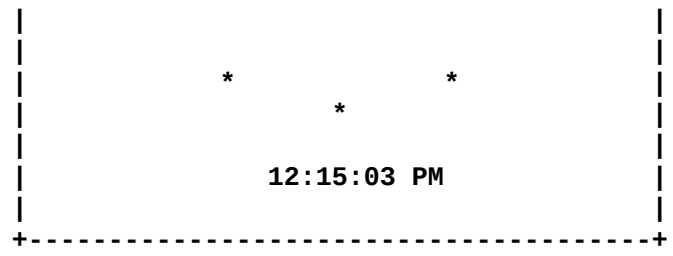
Cutting and copying place the selected text in a hidden "clipboard" accessible by other Notepads -- you can run more than one -- or by the Bplus 2.0 STRING widget.

The Notepad is *not* intended as a replacement for the RMB editing environment! You can edit RMB programs as text files, of course, but it provides no syntax checking and you can't execute a program from it.

[3] CLOCK

* The Clock app is simple in concept:





The "Display" menu has the following entries:

```
| Display  Ti
+----------+-
|  Analog  |               Use analog (HMS-hands) display.
|  Digital |               Use digital (numeric) display.
| *Mixed   |               Use combined analog and digital display.
|  Timer   |               Set digital timer/stopwatch mode.
+----------+
|
```

The "Timer" menu sets timer operation modes:

```
   Timer  Alarm  Options
-+----------------------+-
 |  Set Value...        |    Set timer count value.
 |  Set Update Rate...  |    Set the timer count rate.
 | *Timer counts down   |    Set timer count direction.
 |  Timer repeats       |    Set whether timer counts cyclically.
 |  Notification        |    Set timer notification mode.
 |  Start               |    Start timer.
 |  Reset               |    Clear timer.
 |  Pause               |    Pause timer.
 +----------------------+
```

The "Alarm" menu sets alarm operations:

```
   Alarm  Options
-+-----------------+-
 | Set Time...     |         Set alarm time.
 | Notification    |         Set alarm notification mode.
 +-----------------+
```

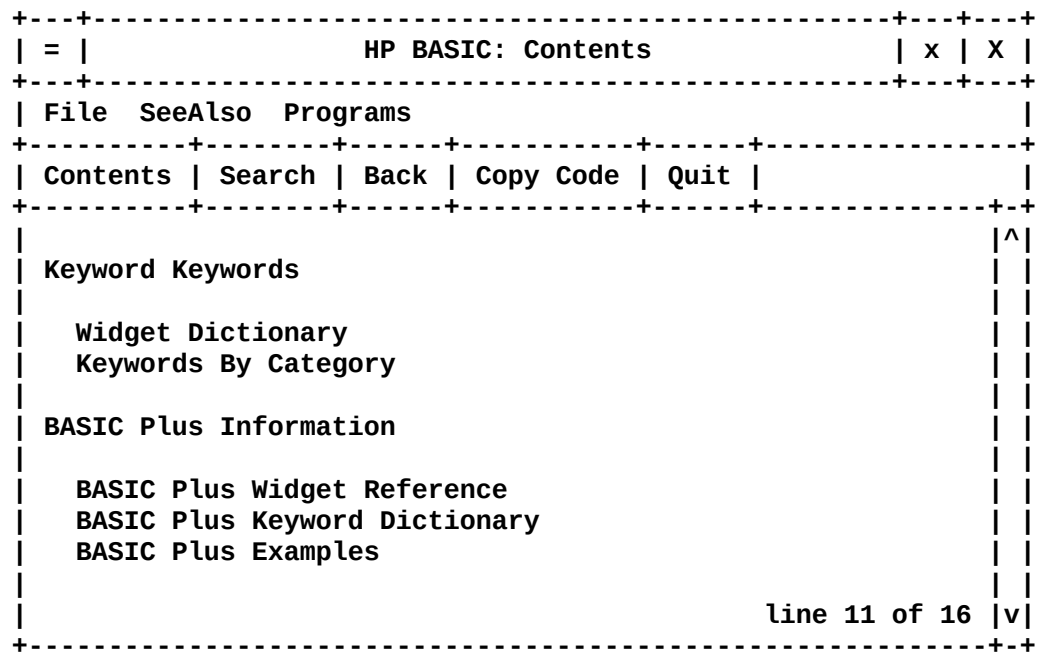
The "Options" menu sets a few minor functions:

```
   Options
-+-----------------+-
 | *Show seconds   |         Enable or disable seconds hand/digits.
```

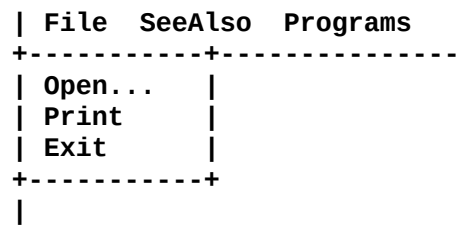
```
|  Military time  |         Enable or disable 24-hour count.
+-----------------+
```

[4] HELP

* The most useful app in the BPlus set is the Help utility (remember, once more, that HBW uses the MS-Win Help utility). The panel selected by HELP looks like this:

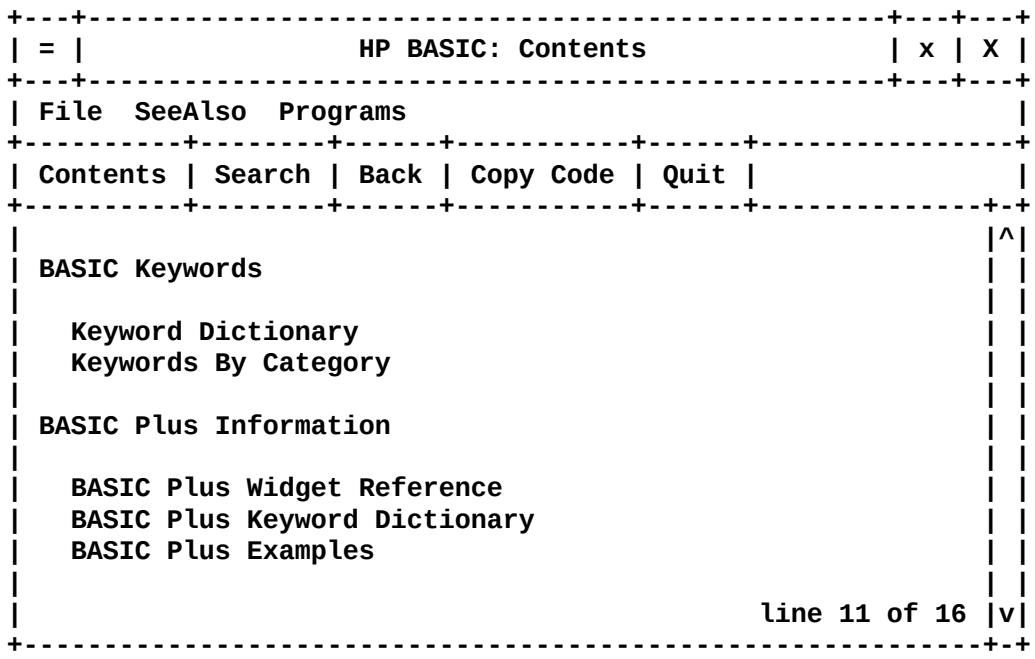


The "File" menu allows you to open a Help file (you can make your own, as the next section shows), print the current help topic to the current RMB PRINTER IS device, or exit the utility:

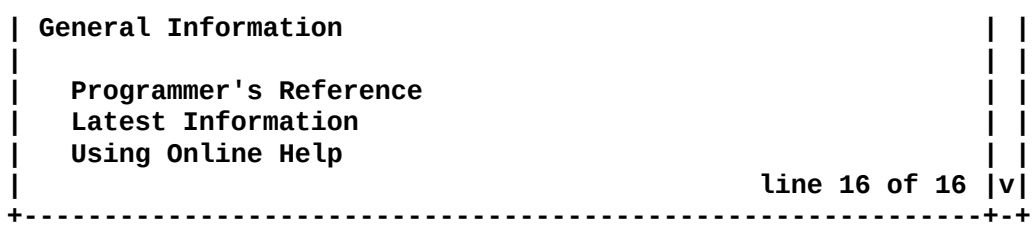


The "SeeAlso" and "Programs" menus are defined by the current Help file; they allow you to cross-reference the current Help topic to other help topics, and to get a list of example programs for the Help topic.

* Using the Help Utility is straightforward. When you first bring up the utility using the standard RMB help file, you get the master menu:

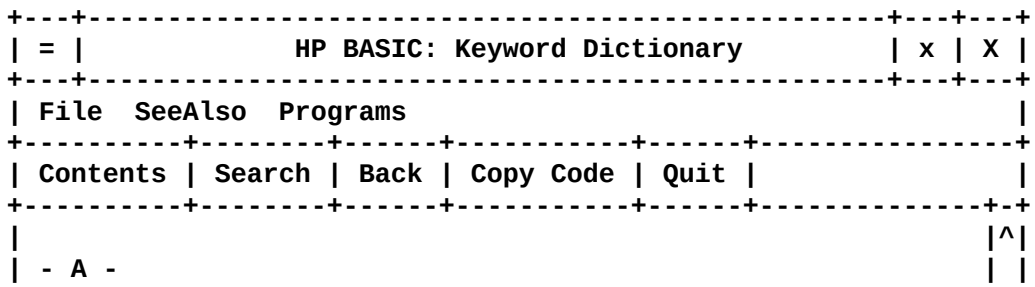


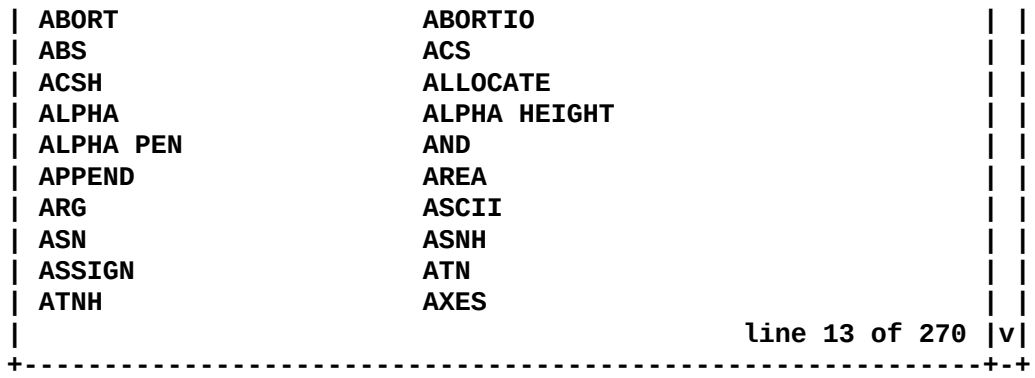
The help topics appear in blue on a color display; anything in blue is an index to another topic, all you have to do is click on it and you get that subject. (The non-blue text won't do anything if you click on it.) Note that all the topics are not displayed, the line on the bottom indicates that there are five more lines scrolled off the bottom of the screen; you can use the scrollbar at the right of the panel to see the rest:



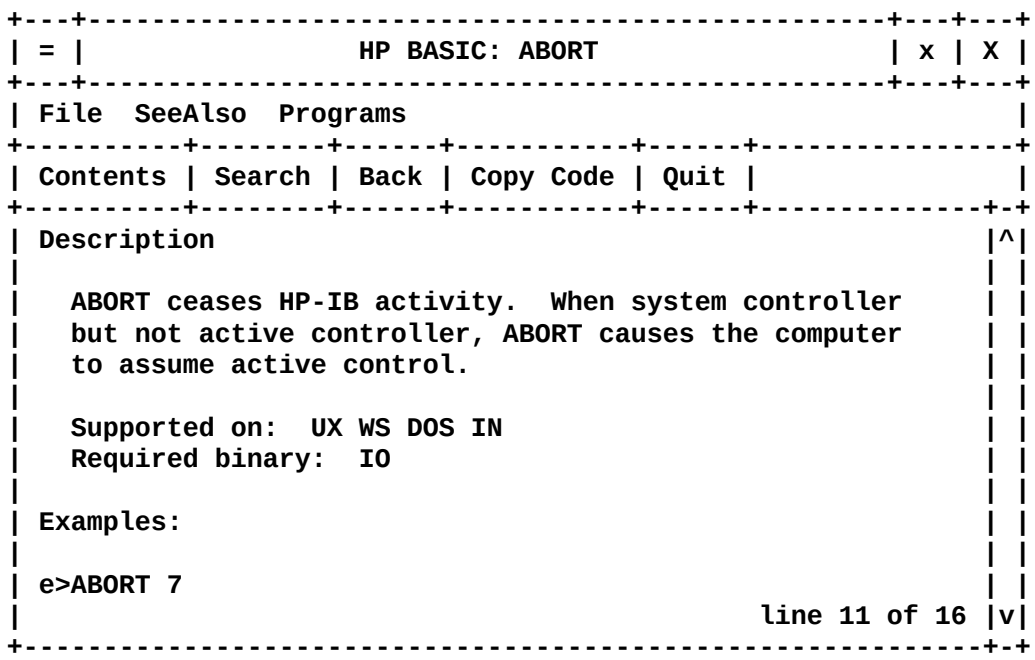
You can scroll one line at a time by clicking on one of the arrows on the scrollbar, or you can grab the marker with your mouse and move text up and down.

In any case, you can then click on the subject you want to investigate; for example, you could go to the "Keyword Dictionary" entry:



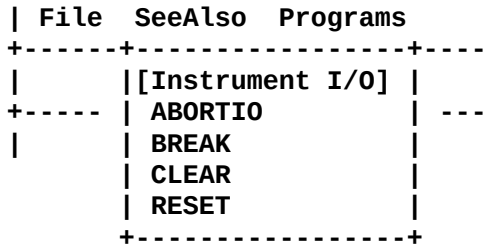


Click on the "ABORT" to get that entry:



Note the example statement at the bottom -- it's marked with an "e>" and is in black. If you click on it, that statement will be copied into the RMB editor.

You can get a list of related topics by selecting the "SeeAlso" menu:



Click on one of these entries and you will get the Help entry for it.

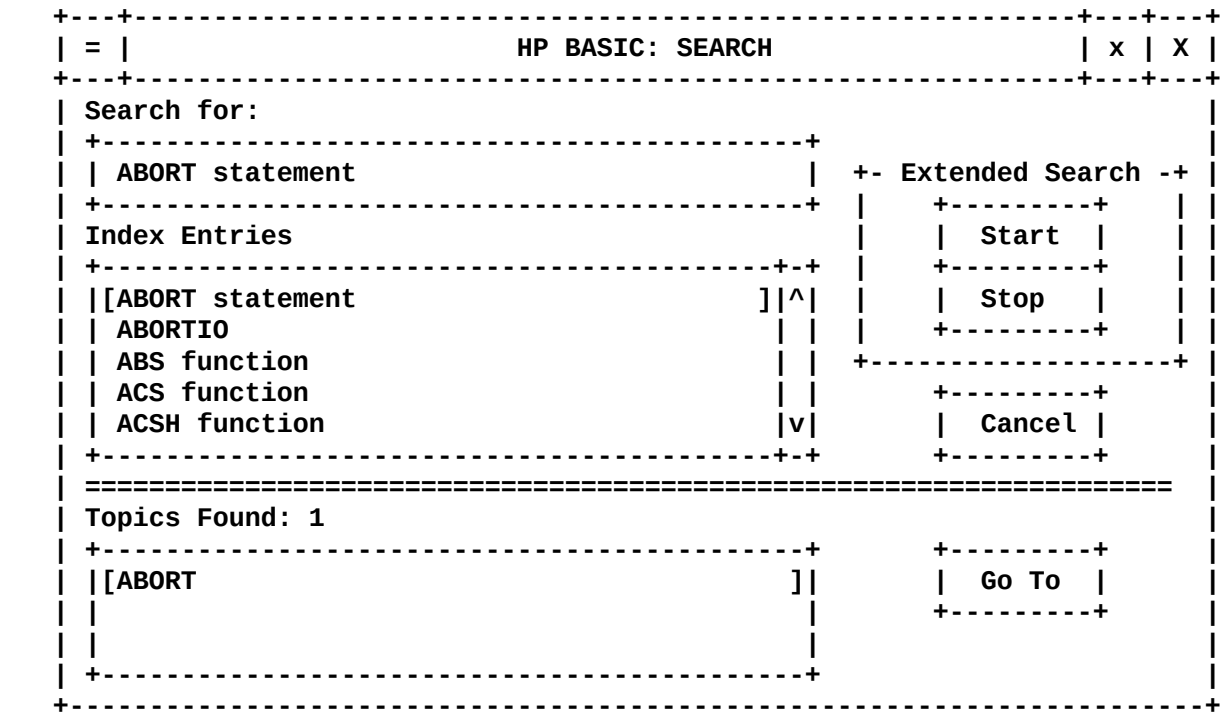
* That should give you a general idea of how the Help utility works. Now for a few other details.

The "Programs" menu allow you to jump to program examples in the Help utility related to the current entry, much like "SeeAlso" -- though relatively few entries have example programs linked to them.

The buttons below the menu:

-- give fast access to a number of functions. When you click on the "Contents" button, you're popped back to the table of contents (as seen at the beginning of this section); the "Back" button allows you to retrace through previous screens; if you have accessed an entry that consists of a program, the "Copy Code" button dumps the code into the RMB editor.

I skipped the "Search" entry because it's a little more complicated than the others ... click on it and you get a dialog:



You can enter topics in the "Search for" field and "Start" an "Extended Search" to see if there is any such topic indexed into the Help file; all indexed entries are listed in the "Topics Found" box, and you can then "Go To" each as you like. You could also search through the "Index Entries" to see all the

indexed entries available ... note that these index pointers have to be specifically defined in the Help file -- you may not be able to find some topics even if they exist in the Help file, if they haven't been indexed.

* The Help system has some other features. Suppose you wanted help on the IMAGE statement; from RMB, you could enter the statement:

```
HELP "ABORT"
```

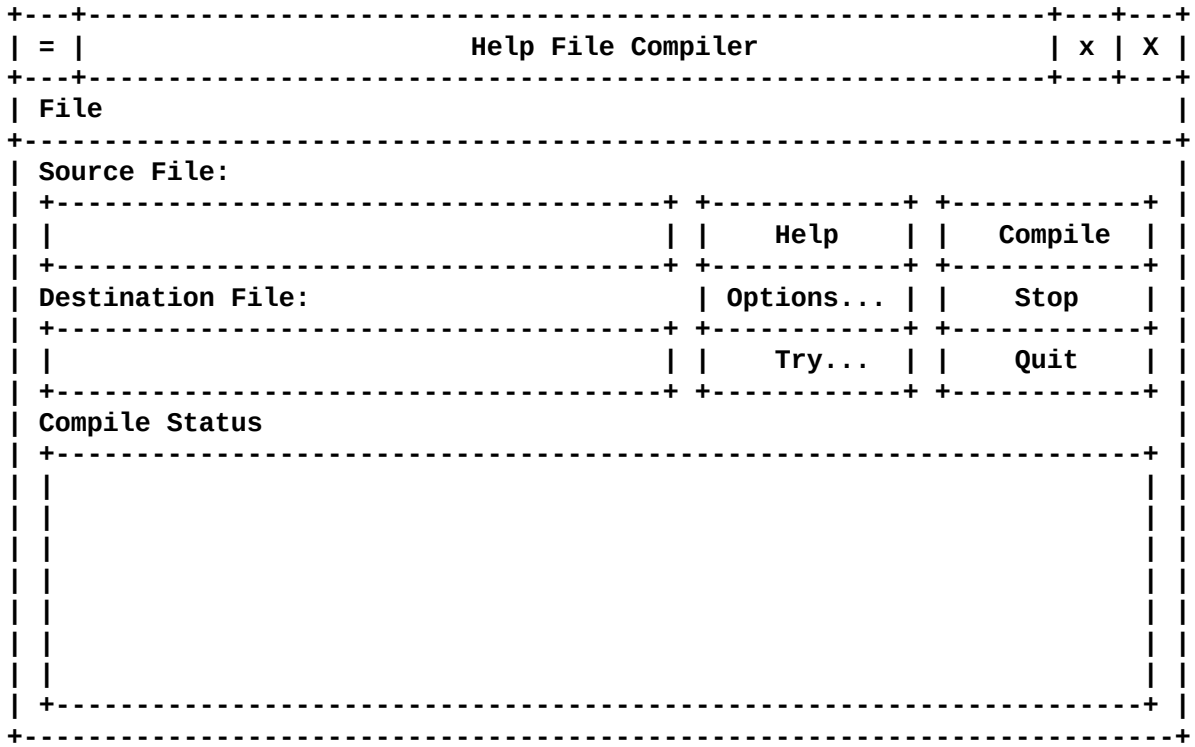
-- and you would get the entry for ABORT as we investigated earlier. You can also optionally invoke Help from RMB using a different file:

```
HELP "SaveEntry","DBASE.HLP"
```

As noted elsewhere, you can also "connect" a widget in a user interface to a Help file and Help topic using the HELP FILE and HELP TOPIC attributes.

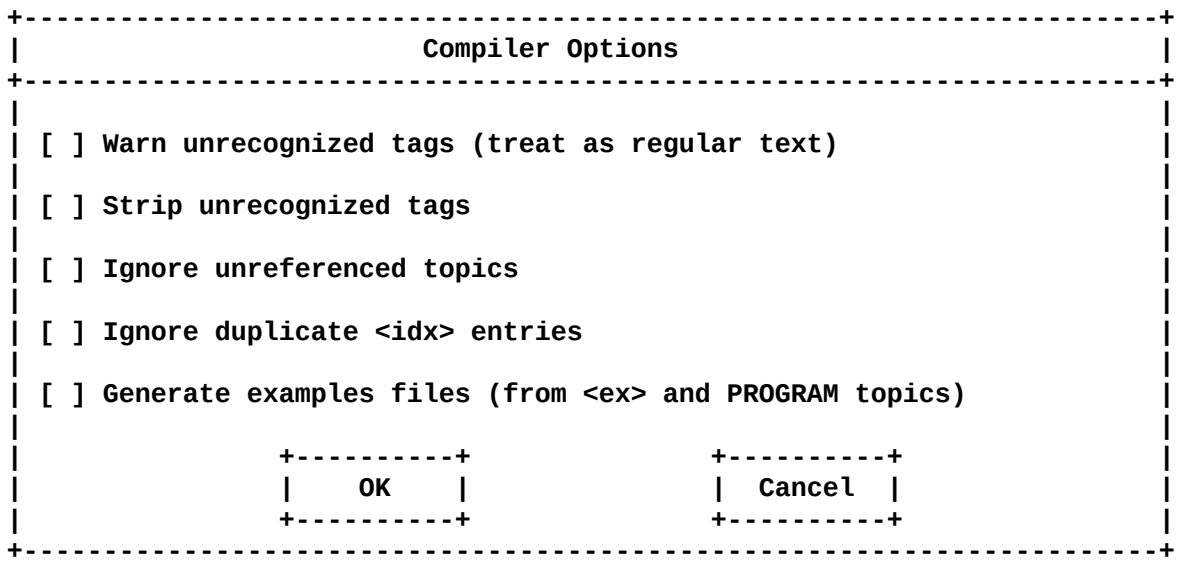
[5] HELP COMPILER

* The Help Compiler allows you to create your own help files for the BPlus Help app (such files are not compatible with the MS-Win help utility used with HBW). It is simple in appearance:

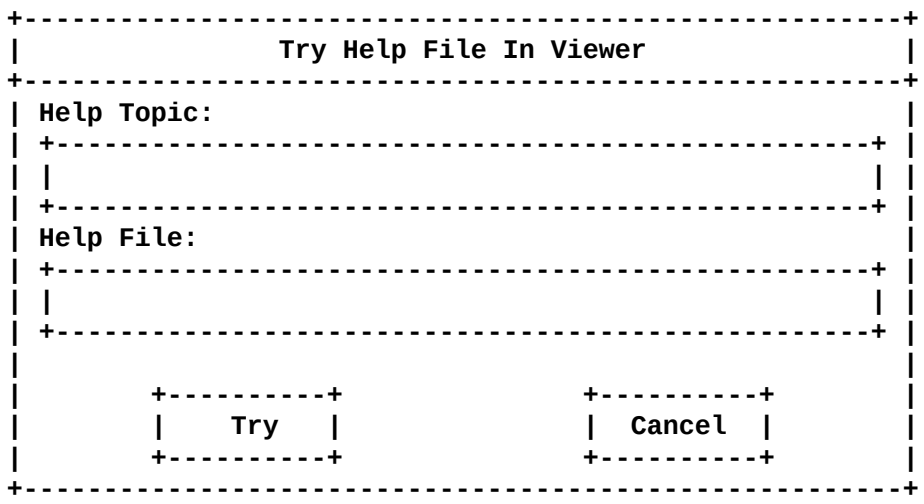


You simply give the Help Compiler the name of a source Help file -- which is a text file written using HELPX Help-language format codes -- and the name of a target Help file, and then click on the "Compile" button; the "Compile Status" shows the results.

The "Help" button brings up the Help utility with a Help file that provides a tutorial on the HELPX language. The "Options..." button allows you to select several compiler options:



-- that don't make too much sense unless you know about HELPX. The "Try..." button allows you to bring up the Help utility with a given Help file and topic:

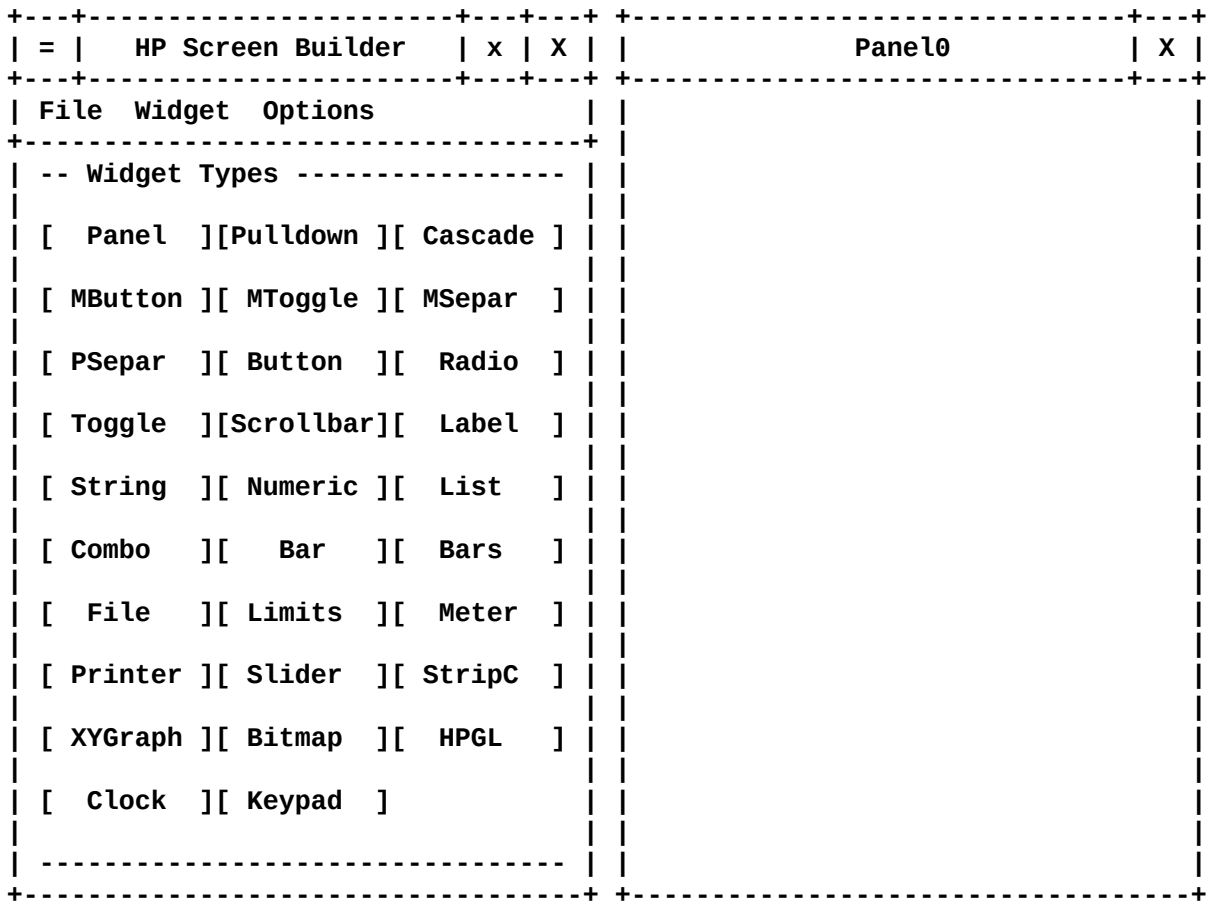


The "Stop" button halts a compile in progress, and of course "Quit" bails out of the utility. The "File" menu on the main panel simply duplicates the "Source...", "Destination...", and "Quit" buttons.

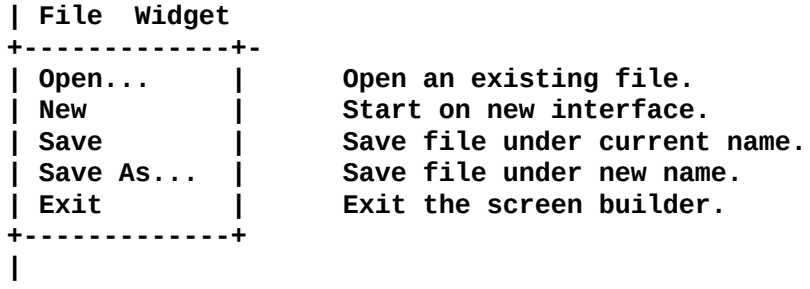
[6] SCREEN BUILDER

* The Screen Builder application is an interactive tool for placing and sizing widgets within a panel; it can save the layout it defines for editing or running later, or for incorporating into a program using the SYSTEM WIDGET. As noted, unlike the other apps, it is supported under HP BASIC for Windows.

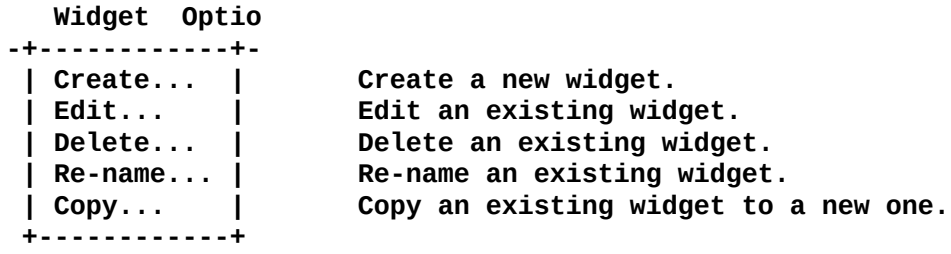
If you bring up the Screen Builder, you get the panels:



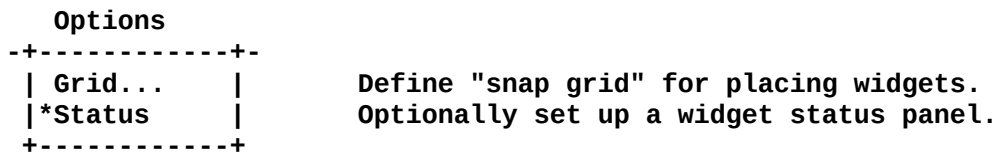
The Screen Builder allows you to set up a user interface and store information that describes it into a file. The "File" menu allows you to perform the appropriate operations on such files:



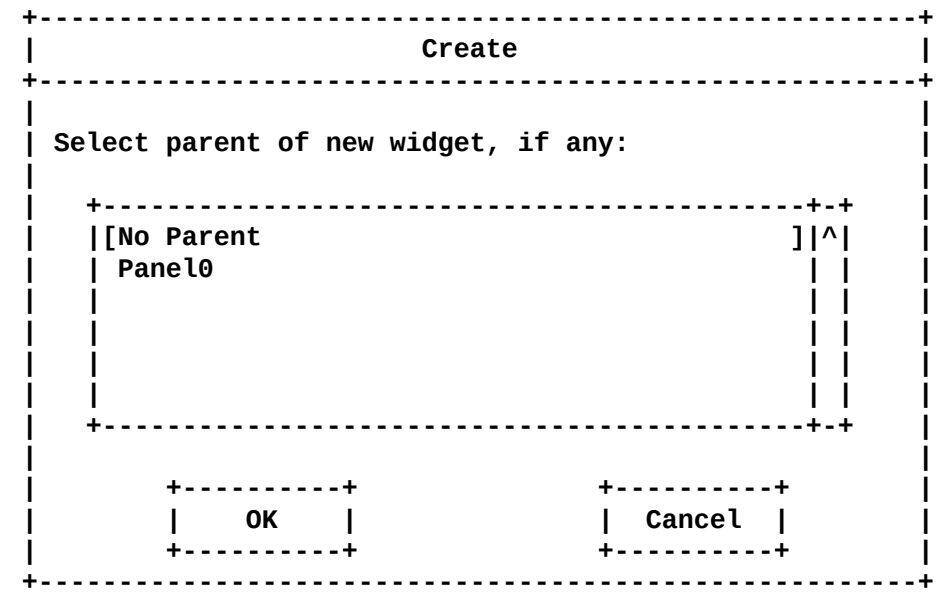
The "Widget" menu allows you to create and perform other operations on widgets:



The "Options" menu allows you to select two options:



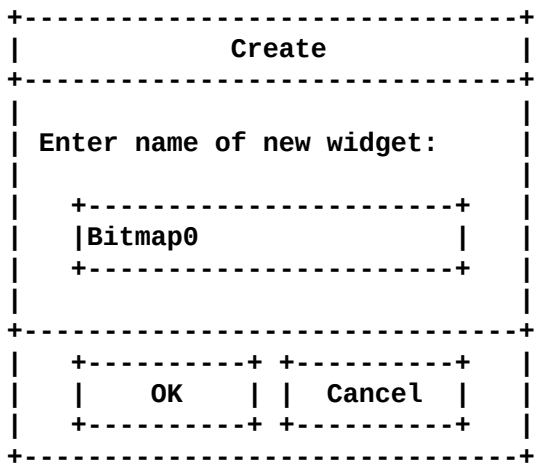
* Using the Screen Builder is simple. You are, by default, given an initial PANEL to lay out widgets; you can then create a new widget by selecting one from the Screen Builder panel (or, if you feel contrary, from the "Create" menu entry). Say you select a BITMAP widget from the Screen Builder panel; you first get a dialog to ask you what to use for a parent, if anything:



Note how the intial PANEL is identified as a string, "Panel0"; this is analogous to a widget handle and is used in the target program to identify a widget.

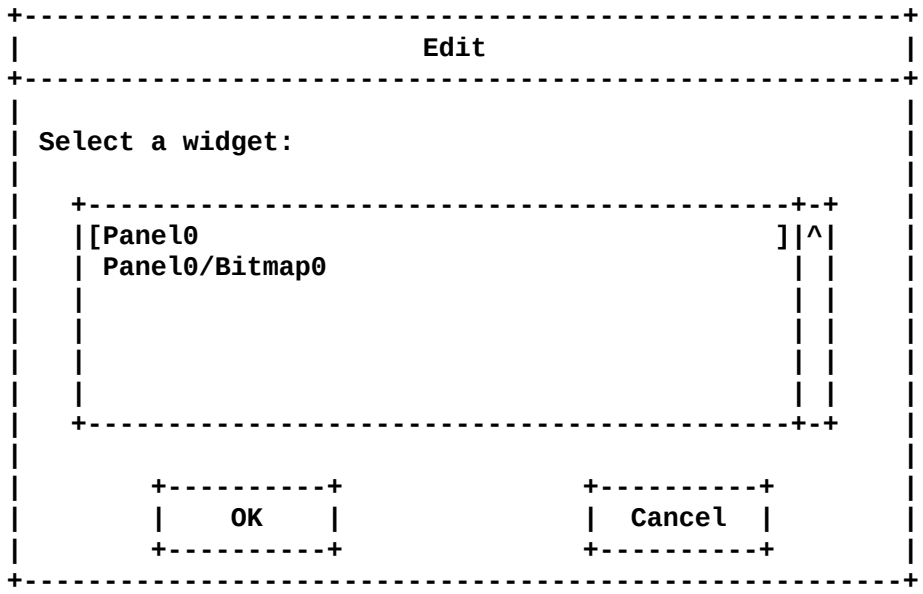
Once you select a parent -- of course we select "Panel0" in this case -- you get a dialog to define the

identifier string of the new BITMAP widget:



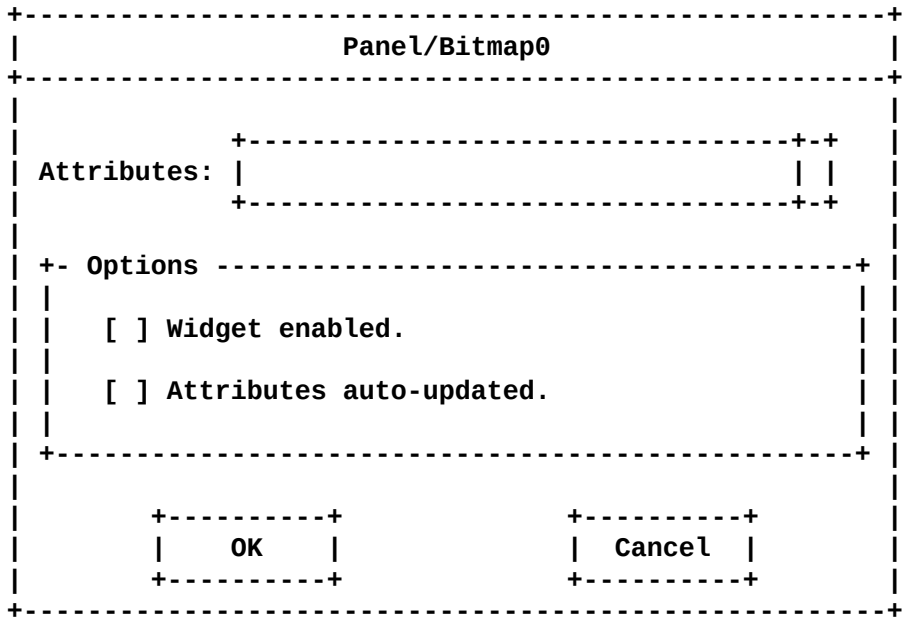
We'll leave the name at the default, "Bitmap0", for now ... anyway, click on OK and the BITMAP widget appears on the default PANEL, and you can move or size it as you please. You can continue to add widgets in this way for about as long as you like.

* Once you establish widgets, though, you will want to set the appropriate attributes on them. You do this with the "Edit" menu entry, which gives you the dialog:

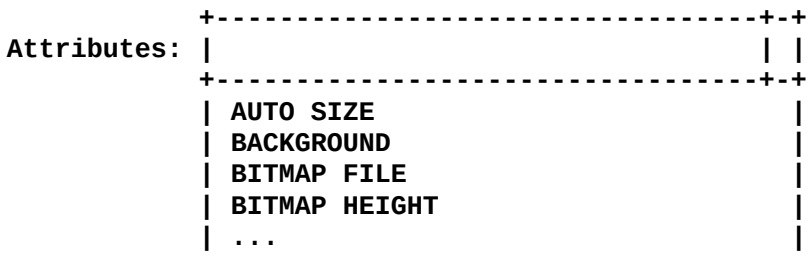


This is lists all the widgets in the user interface being constructed -- we only have two, the default PANEL and the BITMAP widget. Note how the name of the PANEL and the BITMAP form a "path" -- "Panel0/Bitmap0". This path defines the parent-child relationship between the two.

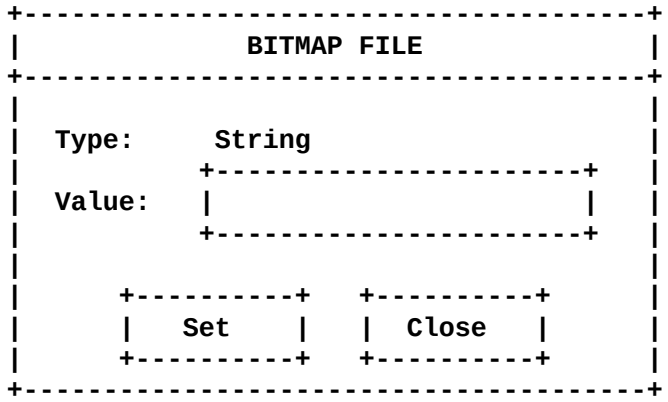
Anyway, select the BITMAP widget and you get an editing dialog:



You can also get this edit dialog simply by double-clicking on a widget ... but to continue, the "Attributes" field controls a pull-down menu:



Select an attribute, say "BITMAP FILE", and you get a dialog to set its value:



Enter a file name, then click on "Set", and you'll set that attribute on the widget -- but you won't go back to the editor dialog until you click on "Close". (This operation can seem a little frustrating at first.)

* Referring back to the widget-editing dialog, there are two flags you can set -- "Widget enabled" and "Attributes auto-updated". The two work together.

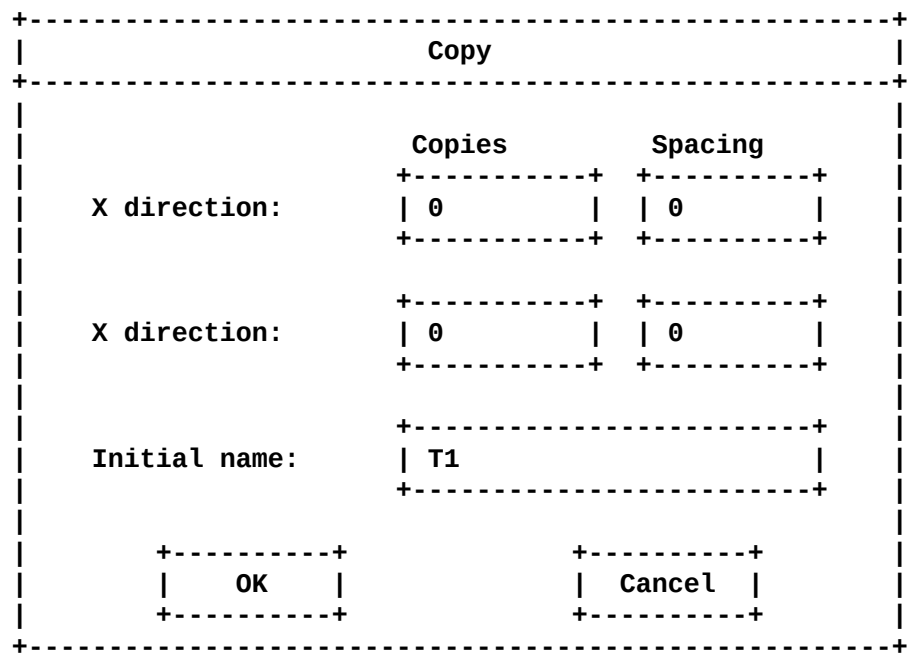
Normally, you can't interact with a widget while you are manipulating it with the Screen Builder; you can't enter text in a STRING widget, or move the slider in a SLIDER widget. Setting "Widget enabled" allows you to interact with the widget, though the value won't be saved in the definition file.

That might seem pretty useless, except for the second flag, "Attributes auto-updated". When you have set "Widget enabled" so you can interact with a widget, and then set "Attributes auto-updated", then when you use the editing dialog and try to modify an attribute in a dialog, that attribute will change as you change the corresponding component on the widget.

That means that if you have the flags set, and attempt to edit the VALUE attribute on a SLIDER widget, the value in the setting dialog will change as you move the slider on the widget -- and will be saved as the appropriate value when you click on the "Save:" button.

* The "Delete", "Re-name", and "Copy" menu entries simply bring up a dialog with a list of the existing widgets; you can select a widget from the list, and following dialogs, much like those already seen, will step you through the operation.

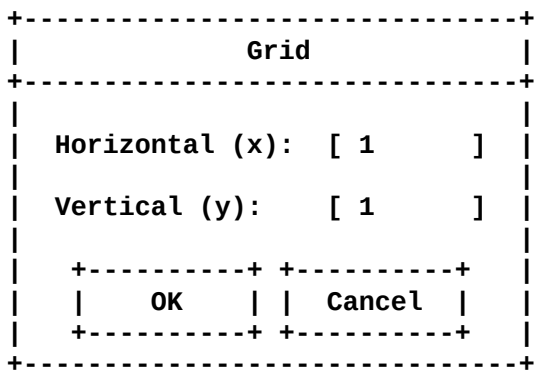
The "Copy" operation does have the additional nice feature of allowing you to create as many copies as you like at once, and place them in a row or column. When you select a widget to copy, you get the dialog:



If you select 5 copies in the X direction with a spacing of 0, you get 5 copies of the widget in a row going to the right, nested together. Selecting copies in the Y direction gives you a column of copies of the widget that grows downward. You can adjust the spacing between the widgets in pixels by setting the appropriate spacing field.

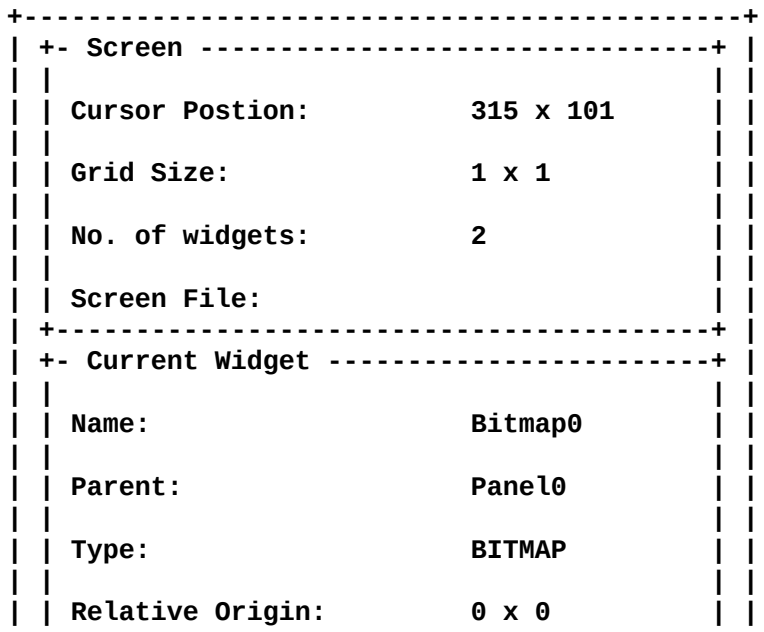
Note that the copy operation gives names that are an incremented count from the initial widget; if the first widget is "T1" as shown above, the copies will be "T2", "T3", and so on.

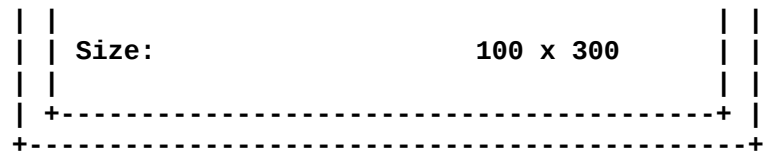
* The "Grid" option brings up a dialog to allow you to set a "snap grid" for placing widgets on the user interface:



If you set the grid coordinates to, say, "4,4", then you can only place the widgets on pixels that are a multiple of 4.

The "Status" option allows you to bring up another Screen Builder panel that gives interactive status on the widgets in the user interface:





Note that this panel may come up over the main Screen Builder panel and hide it ... just move it out of the way with the mouse if that happens. Anyway, the Status Panel displays information on widgets as you pass the mouse over them.

* Once you have set up a user interface and then saved it, you can incorporate it into a program. However, this requires use of the SYSTEM widget, and has to be delayed to the chapter on that subject.

[<>]