

Notebooks as Subwindows in Control Panels

You can create a notebook subwindow in a control panel using the `NewNotebook` operation. A notebook subwindow might be used to present status information to the user or to permit the user to enter multi-line text. Here is an example:

```
NewPanel /W=(150, 50, 654, 684)
NewNotebook /F=1 /N=nb0 /HOST=# /W=(36, 36, 393, 306)
Notebook # text="Hello World!\r"
```

The notebook subwindow can be plain text (`/F=0`) or formatted text (`/F=1`).

By default, the notebook ruler is hidden when a notebook subwindow is created. You can change this using the `Notebook` operation.

The status bar is never shown in a notebook subwindow and there is no way to show it.

To make it easier to use for text input or display, when a formatted text notebook subwindow is first created, and when you resize the width of the subwindow, Igor automatically adjusts the Normal ruler's right indent so that all of the text governed by the Normal ruler fits in the subwindow. This adjustment is done for the Normal ruler only. Other rulers, including Normal+ (variations of Normal) rulers, are not adjusted.

You can programmatically insert text in the notebook using the **Notebook** operation.

If you create a window recreation macro for the control panel, by default the contents of the notebook subwindow are saved in the recreation macro. If you later run the macro to recreate the control panel, the notebook subwindow's contents are restored. This also applies to experiment recreation which automatically uses window recreation macros.

If you do not want the contents of the notebook subwindow to be preserved in the recreation macro, you must disable the autosave property, like this:

```
Notebook Panel0#nb0, autosave=0
```

When you create a window recreation macro while autosave is on, it will contain commands that look something like this:

```
Notebook kwTopWin, zdata="GaqDU%ejN7!Z)ts!+J\\\.F^>EB"
Notebook kwTopWin, zdata= "jmRiCVsF?/]21,HG<k,\"@i1,&\\\.F^>EB"
Notebook kwTopWin, zdataEnd=1
```

The `Notebook zdata` command sends to the notebook encoded binary data in an Igor-private format that represents the contents of the notebook when the recreation macro was created. In real life, there would be a number of `zdata` commands, one after the other, which cumulatively define the contents of the notebook. The notebook accumulates all of the `zdata` text. The `zdataEnd` command causes the notebook to decode the binary data and use it to restore the notebook's contents.

When you save an experiment containing a control panel, a window recreation macro is created for you by Igor. When you open the experiment, Igor runs the recreation macro to recreate the control panel. If autosave is off, after saving and reopening the experiment, the notebook will be empty. If autosave is on, the window recreation macro will include `zdata` and `zdataEnd` commands that restore the contents of the notebook subwindow.

The encoded binary data includes a checksum. If the `Notebook zdata` commands have been altered, the checksum will fail and you will receive an error when the `Notebook zdataEnd` command executes.

For a demonstration of notebook subwindows, choose **File**→**Example Experiments**→**Feature Demos**→**Notebook in Panel**.