Notebook (*Paragraph Properties*)

Notebook paragraph property parameters

This section of Notebook relates to setting the paragraph properties of the current selection in the notebook.

The margins, spacing, justification, tabs and rulerDefaults keywords provide control over paragraph properties which are governed by rulers. These keywords, in conjunction with the ruler and newRuler keywords, allow you to set paragraph properties. They are allowed for formatted text notebooks only, not for plain text notebooks.

The ruler keywords are described in detail below. Before we get to the detail, you should understand the different things you can do with rulers.

There are four things you can do with a ruler:

modify it	(analogous to manually adjusting a ruler).
redefine it	(analogous to the Redefine Ruler dialog).
create it	(analogous to the Define New Ruler dialog).
apply it	(analogous to selecting a ruler name from Ruler pop-up menu).

Igor's behavior in response to ruler keywords depends on the order in which the keywords appear.

To modify the ruler(s) for the selected paragraph(s), use the margins, spacing, justification, tabs and rulerDefaults keywords *without* using the newRuler or ruler keywords. For example:

```
Notebook Notebook0 tabs={36,144,288},justification=1
```

To redefine an existing ruler, invoke the ruler=*rulerName* keyword *before* any other keywords. For example:

```
Notebook Notebook0 ruler=Ruler1,tabs={36,144,288},justification=1
```

Unlike redefining the ruler manually, when you redefine an existing ruler using ruler=*rulerName*, it does not apply the ruler to the selected text. However, it does update any text governed by the redefined ruler.

To create a new ruler, invoke the newRuler=*rulerName* keyword *before* any other keywords. For example:

```
Notebook Notebook0 newRuler=Ruler1,tabs={36,144,288},justification=1
```

Unlike creating it manually, when you create a new ruler using newRuler=*rulerName*, it does not apply the new ruler to the selected text. If you do not set a particular ruler property when creating a new ruler, the property will be the same as for the Normal ruler. If the specified ruler already exists, newRuler=*rulerName* overwrites the existing ruler.

To apply an existing ruler to the selected text, invoke the ruler=*rulerName* keyword without any other keywords. For example:

```
Notebook Notebook0 ruler=ruler1
```

You and Igor will get confused if you mix ruler keywords with other types of keywords in the same command. It is alright, however to put a selection keyword at the start of the command. Mixing will not cause a crash or any drastic problem but it will likely produce results that you don't understand.

To keep things clear, follow these rules:

- If you use ruler=*rulerName* or newRuler=*rulerName*, put them before any other ruler keywords.
- Do not mix ruler keywords with other kinds, except that it is alright to use the selection keyword at the start of the command.

justification=*j* Sets text justification:

- *j*=0: Left aligned.
- *j*=1: Center aligned.
- *j*=2: Right aligned.
- *j*=3: Fully justified.

margins={*indent*,*left*,*right*}

indent sets the indentation of first line from left page margin.

left sets the paragraph's left margin in points measured from the left page margin.

right sets the paragraph's right margin in points measured from the left page margin.

newRuler=*rulerName*

Creates a new ruler with the specified name. If a ruler with this name already exists, it is overwritten.

ruler=*rulerName* Applies the named ruler to the selected text or to redefine the named ruler, as explained above.

rulerDefaults={*"fontName"*, *fSize*, *fStyle*, (*r*,*g*,*b*[,*a*])

}

"fontName" sets the ruler's text font, e.g., `"Helvetica"`.

fSize sets the ruler's text size.

fStyle sets the ruler's text style.

(*r*,*g*,*b*[,*a*]) sets the ruler's text color. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**. The default is opaque white.

You can use rulerDefaults only if you are redefining an existing ruler using ruler=*rulerName* or you are creating a new ruler using newRuler=*rulerName*.

spacing={*spaceBefore*,*spaceAfter*,*lineSpace*}

spaceBefore sets the extra space before paragraph in points.

spaceAfter sets the extra space after paragraph in points.

lineSpace sets the extra space between lines of a paragraph in points.

tabs={*tabSpec*} *tabSpec* is list of tab stops in points added to special values that change the tab stop type.

Tab stops have two parts: the tab stop position and the tab type. Each integer in the list of tabs encodes both of these parts as follows:

The low 11 bits contains the tab stop position in points.

The next two bits are reserved for future use and must be zero.

The high three bits are used to contain the tab type as follows:

left tab	0	
center tab	1	add 1*8192 to tab stop position.
right tab	2	add 2*8192 to tab stop position.
decimal tab	3	add 3*8192 to tab stop position.
comma tab	4	add 4*8192 to tab stop position.

Tabs Example

The following puts a left tab at 1 inch, a center tab at 3 inches and a decimal tab at 5 inches:

```
Notebook Notebook1 tabs={1*72, 3*72 + 8192, 5*72 + 3*8192}
```