Notebook (*Selection*)

Notebook selection parameters

This section of Notebook relates to selecting a range of the content of the notebook.

findPicture={*graphicNameStr*, *flags*}

Searches for the picture containing the named graphic (*Macintosh only*) or the next picture if you pass `""`. Sets V_flag to 1 if the picture was found or to 0 if not found.

flags is a bitwise parameter interpreted as follows:

Bit 0: Show selection after the find.

All other bits are reserved for future use. Set bit 0 by setting *flags* = 1.

The search is always forward from the end of the current selection to the end of the document.

findSpecialCharacter={*specialCharacterNameStr*, *flags*}

Searches for the special character with the specified name or the next special character if you pass `""`. Selects the special character if it is found.

Sets V_flag to 1 if the special character was found or to 0 if not. Sets S_name to the name of the found special character or to `""` if it was not found.

flags is a bitwise parameter interpreted as follows:

Bit 0: Show selection after the find.

All other bits are reserved for future use. Set bit 0 by setting *flags* = 1.

If *specialCharacterNameStr* is empty (`""`), the search proceeds from the end of the current selection to the end of the document. Otherwise the search always covers the entire document.

findText={*textToFindStr*, *flags*}

Searches for the specified text. Sets V_flag to 1 if the text was found or to 0 if not found.

textToFindStr is a string expression for the text you want to find. If the text contains a carriage return, Igor considers only the part of the text before the carriage return.

flags is a bitwise parameter interpreted as follows:

Bit 0:	Show selection after the find.
Bit 1:	Do case-sensitive search.
Bit 2:	Search for whole words.
Bit 3:	Wrap around.
Bit 4:	Search backward.

All other bits are reserved and must be set to zero.

To set bit 0 and bit 3, use $2^0+2^3 = 9$ for *flags*. See **Setting Bit Parameters** on page IV-12 for details about bit settings.

If you are searching forward, the search starts from the end of the current selection. If you are searching backward, the search starts from the start of the current selection.

If you specify `""` as the text to search for, it “finds” the current selection. This displays the current selection using `findText={"", 1}`.

selection={*selStart*, *selEnd*}

selStart and *selEnd* are locations within the document. You can specify these document locations by using the following expressions:

(*paragraph*, *pos*) — *paragraph* and *pos* are numeric expressions.

paragraph is a paragraph number from 0 to *n*-1 where *n* is the number of paragraphs in the document.

pos is a byte position from 0 to *n* where *n* is the number of bytes in the paragraph. Position 0 is to the left of the first character in the paragraph. Position *n* is to the right of the last character in the paragraph.

startOfFile — Start of the document.

endOfFile — End of the document.

startOfParagraph — Start of current *selStart* paragraph.

endOfParagraph — End of current *selStart* paragraph.

startOfNextParagraph — Start of paragraph after current *selStart* paragraph.

endOfNextParagraph — End of paragraph after current *selStart* paragraph.

startOfPrevParagraph — Start of paragraph before current *selStart* paragraph.

endOfPrevParagraph — End of paragraph before current *selStart* paragraph.

endOfChars — Just before the carriage return of current *selStart* paragraph.

startOfPrevChar — Start of the character before the character at the current selection start or selection end. This moves the selection start or selection end like pressing the left arrow key.

Added in Igor Pro 7.00.

startOfNextChar — Start of the character after the character at the current selection start or selection end. This moves the selection like pressing the right arrow key.

Added in Igor Pro 7.00.

Igor clips the specified locations to legal values. It also sets the V_flag variable to 0 if the *selStart* location that you specified was valid, to 1 if the start paragraph was out of bounds and to 2 if the start position was out of bounds. You can use the startOfNextParagraph keyword to step through the document one paragraph at a time. When V_flag is nonzero, you are at the end of the document.

The terms next and prev are relative to the paragraph containing the start of the selected text before the selection keyword was invoked.

The selection keyword just sets the selection. If you also want to scroll the selected text into view you must also use the findText keyword as shown in the examples.

Selection Examples

Following are some examples of setting the selection:

```
// select all text in notebook
Notebook Notebook1 selection={startOfFile, endOfFile}

// move selection to the start of the notebook and display the selection
Notebook Notebook1 selection={startOfFile,startOfFile}, findText={"",1}

// move selection to the end of the notebook and display the selection
Notebook Notebook1 selection={endOfFile,endOfFile}, findText={"",1}

// select all of paragraph 3
Notebook Notebook1 selection={(3,0), (4,0)}

// select all of paragraph 3 and display the selection
Notebook Notebook1 selection={(3,0), (4,0)}, findText={"",1}

// select all of current paragraph except for trailing CR, if any
Notebook Notebook1 selection={startOfParagraph, endOfChars}
```

```
// select the first occurrence of "Hello" in the document and display the selection
Notebook Notebook1 selection={startOfFile,startOfFile}, findText={"Hello",1}

// select the first picture in the document
Notebook Notebook1 selection={startOfFile,startOfFile}, findPicture={"",1}
```

See Also

The **GetSelection** operation to “copy” the selection.