

[Next](#) [Up](#) [Previous](#) [Contents](#)

Next: [How parameter files are](#) **Up:** [pil](#) **Previous:** [Sample programs](#) [Contents](#)

IRAF parameter files

The purpose of PIL library is to enable ISDC applications access IRAF compatible parameter files. IRAF parameter file is a text file. Every line describes single parameter. Format is as follows :

```
name,type,mode,default,min,max,prompt
```

- name : name of parameter
- type : type of parameter. Allowable values:
 - b : means parameter of boolean type
 - i : means parameter of integer type
 - r : means parameter of real type
 - s : means parameter of string type
 - f : for parameter of filename type. f may be followed by any combination of r/w/e/n meaning test for read access, write access, presence of file, absence of file. Thus fw means test whether file given as a value of parameter is writable.
- mode : mode of parameter. Allowable value is any reasonable (it is: a/h/q/hl/ql) combination of :
 - a/auto : effective mode equals to the *value* of parameter named mode in parameter file. If that parameter is not found in parameter file or is found invalid then the effective mode is 'hidden'.
 - h/hidden : No questions are asked, unless default value is invalid for given data type. for example "qwerty" is specified as a value for

integer parameter. Note that is `PILOverrideQueryMode` function was called with argument set to `PIL_QUERY_OVERRIDE` then no questions are asked even for parameters with invalid values.

- `l/learn` : If application changed parameter value, new value will be written to parameter file when application terminates. Actually this takes places when application calls either `PILClose()` or `PILFlushParameters()` (or `CommonExit()` when writing ISDC's CTS compliant software). If this flag is not specified then any changes to the parameter (via `PILGetXXX` or `PILPutXXX`) are lost and are not written to disk.
- `q/query` : Always ask for parameter. The format of the prompt is : prompt field from parameter file (parameter name if prompt field is empty) followed by allowable range (if any) in `<>`, followed by default value (if any) in `[]`, followed by colon. Pressing RETURN key alone accepts default value. If newly entered value (or default value in the case RETURN key alone is pressed) is unacceptable library prompts user to reenter value. If the value is overridden by command line argument, then the PIL library does not prompt for that parameter (if value is valid and within boundaries if any) Note that is `PILOverrideQueryMode` function was called with argument set to `PIL_QUERY_OVERRIDE` then no questions are asked ever (even for parameters with invalid values).
- `default` : default value for parameter [this field is optional]. This can be: yes/no/y/n for booleans, integer/real literals (ie. 123, -34567 for integers, 1.23, 1234, -45.3e-5 for reals) for integers/reals, and string literals for strings and filenames. String literals can be: `abcdef`, `"abcdef"`, `'abcdef'`.
- `min` : minimum value allowable for parameter [this field is optional]. Range checking is enabled only if both `min` and `max` fields are nonempty. See also `max`. When `max` field is empty then `min` field can also contain pipe symbol (vertical bar) separated list of allowable values for given parameter. This works for integer, real, string and filename types. For instance :

```
OutFile,f,q1,"copy.o",/dev/null|copy.o|dest.o,, "Enter output file name"
```

specifies that `OutFile` parameter can take only 3 distinct values, it is `/dev/null`, `copy.o` or `dest.o`. Furthermore, for string (and `_ONLY_` for string, this does `_NOT_` apply for filename parameter types) parameter types, case does not matter, and string returned by `PILGetString` is converted to uppercase. Note, that automatic conversion to uppercase is only done when enum list is specified for given parameter.

- `max` : maximum value allowable for parameter [this field is optional]. Works for integer, real string and filename data types. Both `min` and `max` must be specified for range checking to be active. Also `min` cannot be larger than `max`. In case of any format error in `min` or `max` PIL assumes that no range checking is requested. See also `min` for a description of enumerated value list.
- `prompt` : short description of parameter to be displayed whenever library asks for value. If none is given library will display parameters name.

As an example :

```
pressure,r,ql,1013,,,"Enter atmospheric pressure in hPa"
```

describes parameter named pressure of type real, mode = query + learn, with default value = 1013, no range checking and prompt

```
"Enter atmospheric pressure in hPa"
```

Empty lines and lines beginning with '#' are considered to be comment lines.

[Next](#) [Up](#) [Previous](#) [Contents](#)

Next: [How parameter files are](#) **Up:** [pil](#) **Previous:** [Sample programs](#) [Contents](#)

Bryan Irby 2004-10-05