URL: http://cxc.harvard.edu/ciao3.4/paramio_pquery.html
Last modified: December 2006

AHELP for CIAO 3.4

pquery

Context: paramio

Jump to: Description Examples See Also

Synopsis

Query a parameter value from S−Lang.

Syntax

```
String_Type pquery( paramfile, param )
```

Description

This function is essentially the same as the command−line version (see "ahelp tools pquery"), although only one parameter − the param value above − can be queried at a time. The parameter value is always returned as a string by this function. To use pquery() the paramio module must have been loaded.

paramfile is a string

If called with the name of the parameter file then the function will, if it is not a hidden parameter, prompt the user and then return the selected value.

paramfile is a Param_File_Type variable

The behaviour is the same if the paramopen() call was made without supplying the optional third argument. If the third argument was given then, before querying the user, the routine searches for the parameter value among these values. If the parameter is found in this list then the user will not be queried for the value. This behaviour is useful when writing S−Lang scripts that use a parameter file; see the examples below and those in "ahelp paramio" for more information.

Error handling

As with all the paramio routines, the PF_Errno variable is set to 0 on success, or on error it is set to one of the error codes listed in the paramio documentation.

Example 1

```
chips> require("paramio")
chips> punlearn("dmcopy")
chips> pset( "dmcopy", "infile", "in.fits" )
chips> ifile = pquery( "dmcopy", "infile" )
Input dataset/block specification (in.fits):
chips> ifile
in.fits
chips> clval = pquery("dmcopy","clobber")
chips> clval
no
```

The pset() call sets the value for the infile parameter of dmcopy to "in.fits". We then call pquery() to find its value and − since it is not a hidden parameter − we are prompted for a value. In this example we accept the default value, and so the contents of ifile are "in.fits".

The query of the "clobber" parameter does not result in an interactive query since it is a hidden parameter.

Example 2

```
chips> pset( "dmcopy", "infile", "in.fits" )
chips> ifile = pquery( "dmcopy", "infile" )
Input dataset/block specification (in.fits): test.fits
chips> ifile
test.fits
chips> pget("dmcopy","infile")
in.fits
```

We repeat the first example but this time do not use the default value for infile, instead changing it to "test.fits".

The call to pget() is there to highlight that the "dmcopy" parameter file has not been updated by the pquery() call. This is because it was opened read−only by pquery(); to get the parameter value to actually change requires using paramopen(), as the following example shows.

Example 3

```
chips> punlearn("dmcopy")
chips> fp = paramopen("dmcopy","rw")
chips> pset( fp, "infile", "in.fits" )
chips> ifile = pquery( fp, "infile" )
Input dataset/block specification (in.fits): test.fits
chips> ifile
test.fits
chips> pget( fp, "infile" )
test.fits
chips> paramclose(fp)
chips> pget( "dmcopy", "infile" )
test.fits
```

In this example, since we explicitly open the parameter file as read−write, the pquery() call results in the parameter file being changed.

Example 4

```
chips> args = [ "dmcopy", "test.fits", "out=out.fits", "cl+" ]
chips> fp = paramopen("dmcopy","rw", args )
chips> ifile = pquery( fp, "infile" )
chips> ifile
test.fits
chips> ofile = pquery( fp, "outfile" )
chips> ofile
out.fits
chips> clval = pquery( fp, "clobber" )
chips> clval
yes
chips> paramclose( fp )
chips> pget( "dmcopy", "outfile" )
out.fits
chips> pget( "dmcopy", "clobber" )
no
```

Here we use the three−argument form of paramopen() to set parameter values as if the command had been run from the command−line. Since the parameter values for "infile" and "outfile" are contained in the parameter list (the args array here), the pquery() calls do not need to ask the user what the value is. And, as the parameter file was opened with a mode of "rw", these new parameter settings are written to the parameter file if the parameter mode is not "hidden".

Although this example was run within ChIPS, it is more likely to be useful when run from a S−Lang script executed by slsh; see "ahelp paramio" for an example of this.

See Also

concept
: parameter

modules
: paramio

paramio
: paccess, paramclose, paramopen, pget, pgets, plist_names, pset, punlearn

tools
: dmhistory, dmkeypar, dmmakepar, dmreadpar, paccess, pdump, pget, pline, plist, pquery, pset, punlearn

URL:
http://cxc.harvard.edu/ciao3.4/paramio_pquery.html
Last modified: December 2006