



AHELP for CIAO 3.4

plist_names

Context: [paramio](#)

Jump to: [Description](#) [Examples](#) [See Also](#)

Synopsis

List parameter names for a single tool from S-Lang.

Syntax

```
String_Type [] plist_names( paramfile )  
String_Type [] plist_mode( paramfile, mode )  
String_Type [] plist_type( paramfile, type )
```

Description

These functions return an array of strings containing the parameter names for the given parameter file:

- `plist_names()` returns the names of all the parameters in the file.
- `plist_mode()` returns the names of all parameters matching the supplied mode, which can be one of: `PF_MODE_AUTO`, `PF_MODE_QUERY`, `PF_MODE_LEARN`, or `PF_MODE_HIDDEN`.
- `plist_type()` returns the names of all parameters matching the supplied datatype, which can be one of: `PF_BOOL`, `PF_INT`, `PF_REAL`, `PF_STRING`, or `PF_FILENAME`.

If the parameter file exists but does not contain any entries that match the given mode or type then the return value will be an array with 0 elements; this is also the return value when the mode or type is not a recognised value. If the parameter file does not exist then the routines return the NULL value and `PF_Errno` will be set.

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> n = plist_names("dmextract")  
chips> n
```

```
String_Type[15]
chips> print(n[[0:3]])
infile
outfile
bkg
error
```

Here we use the `plist_names()` command to find the list of parameters for `dmextract`. The return value is an array of strings. For `dmextract` there are 16 parameters; we use the `print()` command to display the first four.

Note that the `require("paramio")` line only needs to be issued once per ChIPS/Sherpa session.

Example 2

```
chips> print( plist_mode("dmextract", PF_MODE_AUTO) )
infile
outfile
```

Here we use `plist_mode()` to find all the auto parameters for `dmextract`.

Example 3

```
chips> fp = paramopen( "dmextract" )
chips> print( plist_type( fp, PF_REAL ) )
bkgnorm
sys_err
```

Here we use `plist_type()` to find all the parameters for `dmextract` which accept a real number. As shown, the `plist` functions can also accept a `Param_File_Type` variable – as returned by `paramopen()` – to access a given parameter file.

See Also

concept

[parameter](#)

modules

[paramio](#)

paramio

[paccess](#), [paramclose](#), [paramopen](#), [pget](#), [pgets](#), [pquery](#), [pset](#), [punlearn](#)

tools

[dmhistory](#), [dmkeypar](#), [dmmakepar](#), [dmreadpar](#), [paccess](#), [pdump](#), [pget](#), [pline](#), [plist](#), [pquery](#), [pset](#), [punlearn](#)