1.2 catalog - Module

Module for file operations

Description

The catalog module displays a list of disk files on your command line interface or in a GUI, depending upon how you set up your *.aipsrc* file.

Display the list of files in a File Catalog GUI by adding the following line to your *.aipsrc* file:
`catalog.default:  gui`
or
Display the list of files on your screen by adding the following line to your *.aipsrc* file:
`catalog.default:  screen`
File operations possible are: copying, renaming, deletion, viewing, selecting, *etc.*.

A catalog `tool` is created by the global constructor function `catalog`. A default catalog tool called **defaultcatalog** is automatically created by typing `include catalog.g` in your glish command line window. You can use **dc** as a shortcut name for **defaultcatalog**.

Double-clicking on a directory will cause a show to that directory or will cause a selected file name to be sent back (if the select(anddismiss) option is enabled).

Aipsrc Variables

catalog.default	'gui' starts GUI when catalog is started Allowed:String Default: ''
catalog.confirm	Confirm file deletion/copy/move? Allowed:yes, directory, no Default: yes
catalog.tablesizeoption	How to show table sizes Allowed:no, shape, bytes Default: no
catalog.alwaysshowdir	Always show directories in file listing? Allowed:Bool Default: T
catalog.sortbytype	Sort file listing by file type? Allowed:Bool Default: F
catalog.vscrollbar	Where to place vertical scrollbar Allowed:left, right Default: right
catalog.view.PostScript	Viewer for PostScript files Allowed:String Default: ghostview
catalog.view.Graphics	Viewer for Graphics files Allowed:String Default: xv
catalog.view.image	Viewer for image files Allowed:imageview, aipsview, defaultviewer Default: imageview
catalog.edit.ascii	Editor for ascii files ('' = default editor) Allowed:String Default: ''
catalog.edit.Glish	Editor for Glish files ('' = default editor) Allowed:String Default: catalog.edit.ascii
catalog.view.ascii	Viewer for ascii files ('' = default editor) Allowed:String ('text' is in glish window) Default: catalog.edit.ascii
catalog.view.Glish	Viewer for Glish files ('' = default editor) Allowed:String ('text' is in glish window) Default: catalog.edit.Glish

Tools

catalog	File operations
checker	Tool to check the AIPS++ environment

Functions

cat	Show all files
ccat	Show calibration tables only
dcat	Show directories only
fcat	Show FITS files only
gcat	Show Glish files only
icat	Show images only
mscat	Show measurement sets only
tcat	Show all tables only

1.2.1 catalog - Tool

Package	**utility**
Module	**catalog**

File operations

Constructors

catalog	File operations

Functions

availabletypes	Return vector of possible file types
canonicalize	Return a file name in canonical form
cli	Use the screen for catalog messages
copy	Copy
create	Create
delete	Delete
dismiss	Dismiss the gui
done	Stop this catalog tool
edit	Edit
execute	Execute a glish file
getalwaysshowdir	Get the alwaysshowdir status
getconfirm	Get the default confirmation status
getmask	Get the mask
getsortbytype	Get the sortbytype status
gettablesizeoption	Get the tablesize option
gui	Use the GUI
lastdirectory	Get the last directory used
lastshowtypes	Get the last shown filetypes used
list	return contents of directories
refresh	Refresh the listing
rename	Rename
screen	Use the screen for catalog messages
selectcallback	Get the selectcallback function

setalwaysshowdir	Set the alwaysshowdir flag
setconfirm	Set the confirmation option
setmask	Set the mask
setselectcallback	Set the selectcallback function
setsortbytype	Set the sortbytype flag
settablesizeoption	Set the tablesize option
show	Summarize the directory on the screen or on the GUI
summary	Show a summary
tool	Tool
type	Return the type of this tool
view	View
whatis	Determine what a particular file is
whatisfull	Determine full characteristics of a file

catalog - Constructor

Package	**utility**
Module	**catalog**
Tool	**catalog**

File operations

Synopsis
catalog()

Returns
tool

---

catalog.gui - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Use the GUI

Synopsis
gui(refresh, show_types, vscrollbarright)

Description

An alternate method of specifying the display of catalog messages is via the aipsrc mechanism: add the following to your .aipsrc file

```
catalog.default: gui
```

The default side where the vertical scrollbar is placed is determined by an aipsrc variable as shown below. Default is 'right'.

```
catalog.vscrollbar: left
```

Arguments

refresh	in	Refresh the contents (in case GUI is already active) Allowed:Bool Default: F
show_types	in	File types to show Allowed:Vector of strings Default: unset
vscrollbarright	in	Vertical scrollbar on right side? (otherwise on left) Allowed:Bool Default: unset

Returns
Bool

---

catalog.screen - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Use the screen for catalog messages

Synopsis
screen()

Description
An alternate method of specifying the display of catalog messages is via the aipsrc mechanism: add the following to your .aipsrc file

```
catalog.default: screen
```

Actually, anything besides "gui" will send log messages to the screen.

Returns
Bool

---

catalog.cli - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Use the screen for catalog messages

Synopsis
cli()

Description
Synonym for function `screen`.

Returns
Bool

---

catalog.availabletypes - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Return vector of possible file types

Synopsis
availabletypes()

Description
Return a vector of string containing the possible files types. Some types are given below. More types can be added in the future. These are:

```
Image                  a table containing an image
Measurement Set        a table holding a MeasurementSet
Calibration            a calibration table
```

```
Log message            a log table
Other Table            a table with an unknown type
FITS                   a FITS file
Postscript
Graphics               gif, tif, jpg, ppm, xpm, bmp, xbm
Glish                  an ascii file holding a glish script
ascii                  an ascii file
Regular File           any other regular file
Directory
```

Note that symlinks are normally resolved. If the file linked to does not exist, the type 'not existing' is shown.

Returns
Vector of strings

---

catalog.whatis - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Determine what a particular file is

Synopsis
whatis(file, dir)

Description
Analyzes the file characteristics to determine what it contains:

- Tables are easily recognized and the type (*e.g.* Measurement Set, Image, Calibration) determined,
- Directories, Postscript, Graphics, Glish, and FITS files are recognized,
- Other files may give mixed results, either showing up as Regular File or ascii.

Note that symlinks are normally resolved, so only if the file linked to does not exist, the type will be 'symbolic link'..

Arguments

file	in	Name of file to be classified Allowed:Any string Default: '.'
dir	in	Directory of the file Allowed:Any string Default: '.'

Returns
Record containing classification information

Example

```
  include "catalog.g";
  dc.whatis("3C273XC1.MS")
# [type=Measurement Set, istable=T, size=0]
```

---

catalog.whatisfull - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Determine full characteristics of a file

Synopsis
whatisfull(file, dir)

Description
This function is similar to function whatis, but it also determines the date and if needed the size of tables. The table and image size can be retrieved in bytes or as the shape depending on the setting of the tablesizeoption variable.

Arguments

file	in	Name of the file Allowed:Any string Default: '.'
dir	in	Directory of the file Allowed:Any string Default: '.'

Returns
Record containing characteristics

Example

```
  include "catalog.g";
  dc.whatisfull("3C273XC1.MS")
# [type=Measurement Set, istable=T, date=Jun 3 16:28, size=[2300,20]]
```

---

catalog.list - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

return contents of directories

Synopsis
list(files, listtypes, strippath)

Description
A vector of strings is returned containing the names of the files in the given directory and matching the types. The possible types can be seen with the function availabletypes. The given path names can contain wildcards in the base name. Optionally the full path is returned. The default is that the full path is returned if the files argument contains more than 1 name.
The resulting vector is made unique, so possible duplicate names are removed.

Arguments

files	in	Path name(s) to be processed Allowed:Vector of strings Default: '.'
listtypes	in	Types to be listed Allowed:Vector of strings Default: 'All'
strippath	in	Strip directory from resulting names Allowed:Bool Default: unset

Returns
Vector of strings

Example

```
  include "catalog.g";
  dc.list("*.g c*")
# catalog.g catalog_meta.g catalog.help synclistboces.g
```

---

catalog.show - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Summarize the directory on the screen or on the GUI

Synopsis
show(dir, show_types, writestatus)

Description
The number of files selected, the names, the types, and the subtypes are all displayed, either on the screen or on the GUI.
The GUI can be used for various file operations such as copying, renaming, deletion, browsing, *etc.*

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'
show_types	in	Types to be shown Allowed:String Default: F
writestatus	in	Write messages to status line? Allowed:Bool Default: T

Returns
Bool

Example

```
include "catalog.g";
dc.show(,'Image Calibration')
```

---

catalog.setmask - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the mask

Synopsis
setmask(mask)

Description
Files can be masked by a string obeying the usual shell syntax. This string can be set either from the GUI or using this command from the command line.

Arguments

mask	in	Mask string Allowed:String Default: ''

Returns
Bool

Example

```
include "catalog.g";
dc.mask('di*')
```

---

catalog.getmask - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the mask

Synopsis
getmask()

Returns
Mask String

---

catalog.setconfirm - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the confirmation option

Synopsis
setconfirm(flag)

Description
Before some operations (copy, rename, delete), the user can be prompted for confirmation. This function sets the default for that option. It can also be set from the GUI or from the aipsrc file like

```
catalog.confirm:  yes
```

Possible confirm options are:

- yes means that confirmation is asked for all files.
- directory means that confirmation is only asked for directories.
- no means that confirmation is never asked.

For backward compatibility the boolean values T (=yes) and F (=no) are also possible.
Note that when using `dc.setconfirm()` in a script it should always be done like:

```
cursts := dc.getconfirm();
dc.setconfirm(F);
dc.delete (...);
dc.setconfirm(cursts);
```

otherwise the default confirmation status is changed unexpectedly in the GUI.

Arguments

flag	in	Flag Allowed:yes, directory ,no Default: yes

Returns
Bool

Example

```
  include "catalog.g";
  dc.setconfirm(T)
```

---

catalog.getconfirm - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the default confirmation status

Synopsis
getconfirm()

Description
The functions delete, copy, and rename can ask for confirmation. By default they do so if the default confirmation status (as returned by this function) is true. It can be set (or unset) by one of the following:

- Pressing the confirm button under the options menu of the GUI
- The following appears in your .aipsrc file:

```
catalog.confirm:  yes
catalog.confirm:  directory
catalog.confirm:  no
```

- You did e.g. the following at the command line:

```
dc.setconfirm('yes')
```

Returns
String

---

catalog.settablesizeoption - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the tablesize option

Synopsis
settablesizeoption(tablesizeoption)

Description
When listing a table, it can be told if and how its size is shown. The size can be shown in bytes (which is the total of the table and all its subtables). The size can also be shown as the shape (number of columns and rows) of the table. For a table containing an image, the image shape is shown. Showing the sizes means that it takes a bit longer to show all the tables in the listing.

This function sets this option (valid options are no, bytes and shape). If an invalid option is given, it is set to no. It can also be set from the GUI.
The initial state can be set from the .aipsrc file:

```
catalog.tablesizeoption:  no
```

Arguments

tablesizeoption	in	option Allowed:String Default: no

Returns
Bool

---

catalog.gettablesizeoption - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the tablesize option

Synopsis
gettablesizeoption()

Returns
String

---

catalog.setalwaysshowdir - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the alwaysshowdir flag

Synopsis
setalwaysshowdir(flag)

Description

When a specific file type is shown, it is often nice to show subdirectories (which may also contain such files) as well. This function sets this option. It can also be set from the GUI.
The initial state can be set from the .aipsrc file:

```
catalog.alwaysshowdir:  T
```

Arguments

flag	in	Flag Allowed:Bool Default: T

Returns

Bool

---

catalog.getalwaysshowdir - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the alwaysshowdir status

Synopsis

getalwaysshowdir()

Returns

Bool

---

catalog.setsortbytype - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the sortbytype flag

Synopsis
setsortbytype(flag)

Description
A list of files can optionally be ordered on file type. This function sets that option. It can also be set from the GUI.
The initial state can be set from the .aipsrc file:

```
catalog.sortbytype: F
```

Arguments

flag	in	Flag Allowed:Bool Default: F

Returns
Bool

---

catalog.getsortbytype - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the sortbytype status

Synopsis
getsortbytype()

Returns
Bool

---

catalog.refresh - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Refresh the listing

Synopsis
refresh()

Returns
Bool

---

catalog.lastdirectory - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the last directory used

Synopsis
lastdirectory()

Returns
String

---

catalog.lastshowtypes - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the last shown filetypes used

Synopsis
lastshowtypes()

Returns
String

---

catalog.delete - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Delete

Synopsis
delete(files, refreshgui, confirm)

Description
Delete one or more files. The argument is a list of files to be deleted. Wildcards may be used (except in the directory path). It returns the number of files deleted (or a fail in case of errors). Note that confirmation is asked for if the `confirm` argument is 'yes' (for any file) or 'directory' (for a directory). By default the `confirm` argument is unset meaning that the default confirmation option (as returned by getconfirm()) is used.

Arguments

files	in	Files to be deleted Allowed:String
refreshgui	in	Refresh the GUI? Allowed:Bool Default: T
confirm	in	Confirmation? Allowed:yes, directory, no Default: unset

Returns
Integer

Example

```
dc.delete("3C273.*model")
```

---

catalog.copy - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Copy

Synopsis
copy(file, newfile, confirm)

Description
Copy a file.
Note that if the new file name already exists confirmation is asked for if the `confirm` argument is 'yes' (for any file) or 'directory' (for a directory). By default the `confirm` argument is unset meaning that the default confirmation option (as returned by getconfirm()) is used.

Arguments

file	in	File to be copied Allowed:String
newfile	in	Name of new file Allowed:String
confirm	in	Confirmation? Allowed:yes, directory, no Default: unset

Returns
Bool

Example

```
dc.copy("3C273.restored", "3C273new/restored");
```

---

catalog.rename - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Rename

Synopsis
rename(file, newfile, confirm)

Description
Rename a file.
Note that if the new file name already exists confirmation is asked for if the `confirm` argument is 'yes' (for any file) or 'directory' (for a directory). By default the `confirm` argument is unset meaning that the default confirmation option (as returned by getconfirm()) is used.

Arguments

file	in	File to be renamed Allowed:String
newfile	in	New name of file Allowed:String
confirm	in	Confirmation? Allowed:yes, directory, no Default: unset

Returns
Bool

Example

```
dc.rename("3C273.restored", "3C273new.restored");
```

---

catalog.view - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

View

Synopsis
view(file)

Description
View a file using an appropriate viewer:

- The Viewer for Images (normally the image.view function) unless told differently via the aipsrc variable `catalog.view.image`.
- ghostview for PostScript files.
- xv for Graphics files.
- the tablebrowser for non-image tables.
- The Glish pager for Glish and ascii files.

You can change the viewer for PostScript, Graphics, image Glish, and ascii files via the aipsrc mechanism: *e.g.* you can add the following to your .aipsrc file

```
catalog.view.image: aipsview
catalog.view.PostScript: pageview
catalog.view.Graphics: /usr/local/bin/xv
catalog.view.Glish: xterm -e less
catalog.view.ascii: emacs -r
```

It is also possible to define these variables as an empty string, which indicates that there is no viewer available for that file type. E.g. it might be possible that a site does not have xv available.

Arguments

file	in	File to be viewed Allowed:String

Returns
Bool

Example

```
dc.view("3C273.restored")
```

---

catalog.tool - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Tool

Synopsis
tool(file)

Description
Construct a Tool from a file using toolmanager.show.

Arguments

file	in	File to be processed Allowed:String

Returns
Bool

Example

```
dc.tool("3C273.restored")
```

---

catalog.edit - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Edit

Synopsis
edit(file)

Description
Edit a file using an appropriate editor. For tables, the table browser is used. For ASCII and Glish files, the editor used is first that defined in the environment variable EDITOR. If that is not defined, then emacs is used. This can be overridden by putting the appropriate string in your .aipsrc file:

```
catalog.edit.Glish: xterm -e vi
catalog.edit.ascii: emacs
```

Arguments

file	in	File to be edited Allowed:String

Returns
Bool

Example

```
dc.edit("3C273.g")
```

---

catalog.create - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Create

Synopsis
create(file, type, refreshgui)

Description
Create a file using the appropriate method:

- the default editor for ascii and Glish files
- the unix command mkdir for Directories

You can change the editor Glish, and ascii files via the aipsrc mechanism: *e.g.* add the following to your .aipsrc file

```
catalog.edit.Glish: xterm -e vi
catalog.edit.ascii: emacs
```

Arguments

file	in	File to be created Allowed:String
type	in	Type of file to be created Allowed:ascii—Glish—Directory
refreshgui	in	Refresh the GUI? Allowed:Bool Default: T

Returns
Bool

Example

```
dc.create("3C273.g")
```

---

catalog.summary - Function

Package **utility**
Module **catalog**
Tool **catalog**

Show a summary

Synopsis
summary(file)

Description
Summarize a file using the summary function of the appropriate tool:

- For AIPS++ Tables: summary.
- For AIPS++ Images: summary.
- For MeasurementSets: summary.

The summary information is written to the log file.

Arguments

file	in	File to be summarized Allowed:String

Returns
Bool

Example

```
dc.summary("3C273.restored")
```

---

catalog.execute - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Execute a glish file

Synopsis
execute(file)

Description
Execute a glish script. Other file types cannot be executed.

Arguments

file	in	File to be executed Allowed:String

Returns
Bool

Example

```
dc.execute("test.g")
```

---

catalog.canonicalize - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Return a file name in canonical form

Synopsis
canonicalize(file)

Description
It returns the name in canonical form by removing possible . and .. parts in the string and by adding the working directory if a relative name is given. If an illegal file name is given, a . is returned as file name.

Arguments

file	in	Name of file to be canonicalized Allowed:Any string Default: '.'

Returns
String

Example

```
  include "catalog.g";
  dc.canonicalize("3C273XC1.MS/..")
# /aips++2/gvandiep/data
```

---

catalog.type - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Return the type of this tool

Synopsis
type()

Description
It returns the type of this tool (catalog).

Returns
String

Example

```
  include "catalog.g";
  dc.type()
# catalog
```

---

catalog.dismiss - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Dismiss the gui

Synopsis
dismiss()

Description
Dismiss the gui (if used).

Returns
Bool

---

catalog.done - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Stop this catalog tool

Synopsis
done()

Description
The catalog tool is stopped after its possible gui is dismissed and possible outstanding viewers are dismissed. Note that it is not a good idea to stop the default catalog tool, because it can be used by other scripts.

Returns
Bool

---

catalog.setselectcallback - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Set the selectcallback function

Synopsis
setselectcallback()

Description
Set the callback function used when the send or sendandddismiss button is pressed in the gui. It also activates those buttons. The callback function gets one argument which is a string containing the full name of the selected file. If no file is selected, the directory is returned.
When the break button is pressed in the gui, the selectcallback function is called with the argument unset(). Thereafter the the selectcallback is unset. If the argument given to setselectcallback is not a function, the selectcallback is unset.

Returns
Bool

Example

```
include "catalog.g";
aa := function (f) { if (!is_unset(f)) print f}
dc.setselectcallback(aa)
```

---

catalog.selectcallback - Function

Package	**utility**
Module	**catalog**
Tool	**catalog**

Get the selectcallback function

Synopsis
selectcallback()

Description
Get the callback function set by setselectcallback.

Returns
String

1.2.2 cat - Function

Package	**utility**
Module	**catalog**

Show all files

Synopsis
cat(dir)

Description
cat(dir) is a convenient shorthand for:

```
dc.show(dir)
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.3 icat - Function
Package utility
Module catalog

Show images only

Synopsis
icat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'Image')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.4 ccat - Function
Package utility
Module catalog

Show calibration tables only

Synopsis
ccat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'Calibration')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.5 mscat - Function
Package utility
Module catalog

Show measurement sets only

Synopsis
mscat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'Measurement Set')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.6 fcat - Function

Package utility
Module catalog

Show FITS files only

Synopsis
fcat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'FITS')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.7 gcat - Function

Package utility
Module catalog

Show Glish files only

Synopsis
gcat(dir)

Description

This is a convenient shorthand for:

```
dc.show(dir,'Glish')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.8 tcat - Function
Package utility
Module catalog

Show all tables only

Synopsis
tcat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'<Any Table>')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.9 dcat - Function
Package utility
Module catalog

Show directories only

Synopsis
dcat(dir)

Description
This is a convenient shorthand for:

```
dc.show(dir,'Directory')
```

Arguments

dir	in	Directory to be processed Allowed:String Default: '.'

Returns
Bool

1.2.10 checker - Tool

Package	**utility**
Module	**catalog**

Tool to check the AIPS++ environment
include "checker.g"

Constructors

checker	Construct an checker

Functions

all	Check all properties
display	Check display properties
gui	Check gui properties
memory	Check memory properties
perl	Check to see if perl is available
plotter	Check plotter properties
type	Return the type of this tool

Description

This tool checks the AIPS++ environment for suitability for running.
Unsuitable settings are noted and recommendations are made.
There is a defaultchecker with abbreviation dch. Normally only one should be needed.

checker - Constructor

Package	**utility**
Module	**catalog**
Tool	**checker**

Construct an checker

Synopsis
checker()

Returns
checker tool

Description

This is the only `checker` constructor. Since only one checker should ever be needed, the use of this constructor should be unnecessary.

---

checker.type - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Return the type of this tool

Synopsis
type()

Returns
String

Description

This tool function returns the string "checker". If you have a tool, but don't know what kind it is, this function usually exists to tell you who the tool is.

---

checker.all - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check all properties

Synopsis
all(verbose)

Returns
Number of warnings

Description
All the checks are performed.

Arguments

verbose	Be verbose? Allowed:Boolean Default: F

---

checker.display - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check display properties

Synopsis
display(verbose)

Returns
Number of warnings

Description
The root visual is checked to see if it is not in TrueColor mode.

Arguments

verbose	Be verbose? Allowed:Boolean Default: T

---

checker.perl - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check to see if perl is available

Synopsis
perl(verbose)

Returns
Number of warnings

Description
perl is used for some operations in the help system. This checks to see if perl is available.

Arguments

verbose	Be verbose? Allowed:Boolean Default: T

---

checker.plotter - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check plotter properties

Synopsis
plotter(verbose)

Returns
Number of warnings

Description
The environment variable PGPLOT_FONT is checked to verify that it points to an existing file. Also the Glish variable pgplot is checked for existence and that it is a function.

Arguments

verbose	Be verbose? Allowed:Boolean Default: T

---

checker.gui - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check gui properties

Synopsis
gui(verbose)

Returns
Number of warnings

Description
have_gui() is run to see if Glish has a GUI.

Arguments

verbose	Be verbose? Allowed:Boolean Default: T

---

checker.memory - Function

Package	**utility**
Module	**catalog**
Tool	**checker**

Check memory properties

Synopsis