Functions

execute	execute a function periodically
remove	remove a callback function from the timer
wait	wait a specified amount of time

Description

Plot the lines of code in AIPS++ as a function of date. This only works at the AOC unless you wish to ftp the file from
ftp://aips2.nrao.edu/../linecount/source_lines.

Arguments

file	Name of text file containing source lines Allowed:String Default: /export/aips++/linecount/source_lines

Example

```
include 'aips2loc.g'
aips2loc();
```

1.6.29 ftp - Tool

Package utility
Module misc

Tool to ftp files
include "ftp.g"

Constructors

ftp	Construct an ftp tool

Functions

ascii	Use ascii mode for transfer
binary	Use binary mode for transfer
cd	Change directory

connect	Connect to the host
disconnect	Disconnect from the host
get	Get a file
send	Send a file
type	Return the type of this tool

Description

This tool gets and sends files using ftp. While it is not a particularly good interface to ftp, it is adequate for most simple applications.

ftp - Constructor

Package	**utility**
Module	**misc**
Tool	**ftp**

Construct an ftp tool

Synopsis
ftp(host, user, pass, command, prompt, verbose)

Returns
ftp tool

Description
This is the only `ftp` constructor.

Arguments

host	Host to connect to Allowed:String Default: aips2.nrao.edu
user	User name Allowed:String Default: anonymous
. pass	Password Allowed:String Default: e-mail address
command	ftp command string Allowed:String Default: ftp -n -v -i
. prompt	ftp command prompt Allowed:String Default: ftp¿
. verbose	Report lots of information? Allowed:Boolean Default: T—F

ftp.type - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Return the type of this tool

Synopsis
type()

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

Returns
String

---

ftp.connect - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Connect to the host

Synopsis
connect()

Description
This tool function connects to the host.

Returns
Boolean

---

ftp.disconnect - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Disconnect from the host

Synopsis
disconnect()

Description
This tool function disconnects from the host.

Returns
Boolean

---

ftp.binary - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Use binary mode for transfer

Synopsis
binary()

Description
Tell ftp to use binary mode when transferring files.

Returns
Boolean

---

ftp.ascii - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Use ascii mode for transfer

Synopsis
ascii()

Description
Tell ftp to use ascii mode when transferring files.

Returns
Boolean

---

ftp.get - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Get a file

Synopsis
get(file)

Description
This tool function gets a file.

Arguments

file	File name Allowed:String	

Returns
Boolean

---

ftp.send - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Send a file

Synopsis
send(file)

Description
This tool function sends a file.

Arguments

file	File name Allowed:String	

Returns
Boolean

---

ftp.cd - Function

Package	**utility**
Module	**misc**
Tool	**ftp**

Change directory

Synopsis
cd(dir)

Description
This tool function changes the current directory.

Arguments

dir	Directory name Allowed:String	

Returns
Boolean

1.6.30 ftptest - Function

Package utility
Module misc

Test the ftp tool

Synopsis
ftptest(verbose)

Description
This global function tests the ftp tool by anonymous connection to aips2.nrao.edu from where the file source_lines is retrieved.

Arguments

verbose	Report lots of information? Allowed:Boolean Default: T—F	

Returns
Boolean

1.6.31 make - Function

Package utility
Module misc

Function for a production system

Synopsis
make(target, args, makefile, Execute, script, debug)
include "make.g"

Description

`make`
`make` implements a *production system.* A production system makes a product or products according to some well-specified rules. It is of very widespread applicability but it's best suited to automating a set of steps that must be carried out in sequence but with some variation if, for example, a given product is sometimes available and sometimes not. The principle is familiar to users of the Unix make system, on which make is built. `make` keeps track of which files have changed and knows to remake dependent products.
`make` needs a *makefile* that specifies the production rules.

```
.SUFFIXES: .image .ms .fits

.fits.ms :
glish include 'ms.g';m:=fitstoms('$@', '$<');m.done();

.ms.image :
glish include 'imager.g';myimgr:=imager('$<');myimgr.advise(takeadvice=T,fieldofview='2arcmin');myimgr.makeimage('observed','$@');myimgr.done();

clean:
shell rm -rf *.fits *.ms *.image

data:
shell cp ${AIPSROOT}/data/demo/3C273XC1.fits .
```

By using this makefile, make knows how to make the image, 3C273XC1.image, from the MeasurementSet, 3C273XC1.ms, and it knows how to make the MeasurementSet from the original FITS file. The glish commands are:

```
include 'make.g';
make('clean');
make('data');
make('3C273XC1.image');
```

The arguments allowed are:

```
    -D <VARIABLE>
        define <VARIABLE> to be 1

    -I <DIR>
        Specify a directory in which to search for makefiles
        and included makefiles.

    -d flags
        Turn on debugging, and specify which portions of make are to
        print debugging information.  Flags is one or more of the following:

           A       Print all possible debugging information; equivalent to
                   specifying all of the debugging flags.

           d       Print debugging information about directory searching and
                   caching.

           g1      Print the input graph before making anything.

           g2      Print the input graph after making everything, or before
                   exiting on error.

           m       Print debugging information about making targets,
                   including modification dates.

           s       Print debugging information about suffix-transformation
                   rules.

           t       Print debugging information about target list
                   maintenance.

           v       Print debugging information about variable assignment.

    -e      Specify that environmental variables override macro assignments
            within makefiles.
```

If a script is specified, then `make` generates the commands and writes these to the script file. If the argument `execute` is set to T then the commands are also executed. If one of the command fails then the make is terminated at that point.