```
FLAGVER       0
DOCALIB       1
GAINUSE       0
IN2NAME      ' '
IN2CLASS     ' '
IN2SEQ        0
IN2DISK       0
INVERS        0
NCOMP         0
FLUX          0
NMAPS         0
PMODEL        0
SOLINT        0
SOLTYPE      'APPR'
PRTLEV        0
REFANT        0
BPARM         0
CPARM         0
BADDISK       0

GO PCAL
```

Importing visibility data from FITS files
This benchmark has the following code descriptor:

FUV-RD-*data*

where *data* is the input data description, further broken down as *data*=[*instrument*][*size*][*compression*][*obsmode*][*nchan*]. The parameters are specified as follows: i) *instrument* - telescope abbreviation ii); *size* - dataset size on an integer scale of 1 to 10 (4 $\sim$ 630k UVFITS visibilities); iii) *compression* - MS compression (U=uncompressed; C=compressed); iv) *obsmode* - observing mode (C=continuum, L=line); v) *nchan* - number of frequency channels.
The individual FITS Filler benchmarks implemented in the default run of the *fuvrd*() method at present are enumerated in Table 1.867.

Tools

atmosphere	Tool for calculating atmospheric effects in the mm regime
benchmark	tool for standard AIPS++performance benchmarks

1.9.1 benchmark - Tool

Package utility
Module benchmark

Table 1.867: Table for FITS UV read benchmark (import visibility data into AIPS++ from FITS files). Common part of the code is 'FUV-RD-VLA'.

Benchmark code	Dataset	Mode	N_{chans}	Compressed?	Data size
C-U125K-C1	vlac125k.fits	Continuum	1	False	125K
C-U1M-C1	vlac1m.fits	Continuum	1	False	1M
L-U125K-C64	vlal125K64Chan.fits	Line	64	False	125K

tool for standard AIPS++performance benchmarks

Description
The `benchmark` tool provides access to the standard AIPS++ performance benchmarks. A `benchmark` tool is created without any user-specified parameters.

Constructors

benchmark	Construct an benchmark tool

Functions

all	Run all defined benchmarks
almafuvrd	Run all ALMA-TI almati2ms FITS UV read benchmarks
almatifuvrd	Run the almati2ms FITS Filler, IRAM data, ALMA-TI format, uncompressed benchmarks
calvla	Run all default VLA calibrator benchmarks
calvlau27s10	Run the VLA calibrator benchmarks for G and D Jones
ccsf	Run all Clark Clean, single-field benchmarks
ccsfbi4nl	Run the Clark Clean, single-field, BIMA, 139M vis, uncompressed, continuum benchmarks
ccsfvlacu125k	Run the Clark Clean, single-field, VLA, 125k vis, uncompressed, continuum benchmarks
ccsfvlacu1m	Run the Clark Clean, single-field, VLA, 1M vis, uncompressed, continuum benchmarks
ccsfvlalu125k	Run the Clark Clean, single-field, VLA, 125K vis, 64 frequency channel, uncompressed, spectral line benchmarks
done	Terminate the benchmark process
fuvrd	Run all FITS UV read benchmarks
fuvrdvlacu125k	Run the FITS Filler, VLA, 125k vis, uncompressed, continuum benchmarks
fuvrdvlacu1m	Run FITS Filler, VLA, 1M vis, uncompressed, continuum benchmarks
fuvrdvlalu125k	Run the FITS Filler VLA, 125K vis, 64 frequency channel, uncompressed, spectral line benchmarks
sdall	Run the single dish benchmark tests
sdbench	Run the single dish benchmark tests for a particular data set

benchmark - Constructor

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Construct an benchmark tool

Synopsis
benchmark(datarepospath)

Description

This is used to construct a `benchmark` tool. This tool can then be used to run individual benchmarks or all defined benchmarks.
It can be used as `benchmark()` to construct the benchmark tool with the data repository set to the standard AIPS++ directory.
This returns a Glish variable containing the tool functions of `benchmark`.

Returns
benchmark tool

Arguments

datarepospath	Path for the test data repository Allowed:String Default: '$AIPSROOT/data/demo/benchmark'

Example

```
bench:=benchmark()
bench.all()
bench.done()
```

---

benchmark.all - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run all defined benchmarks

Synopsis
all()

Description
Run all benchmarks currently defined.

Arguments

Returns
Bool

Example

```
bench:=benchmark()
bench.all()
bench.done()
```

---

benchmark.done - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Terminate the benchmark process

Synopsis
done()

Description
This is used to totally stop the `benchmark` process. It is a good idea to conserve memory use on your machine by stopping the process once you no longer need it.

Returns
Bool

Example

```
bench:=benchmark()
bench.all()
bench.done()
```

---

benchmark.ccsf - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run all Clark Clean, single-field benchmarks

Synopsis
ccsf()

Description
This function will run all Clark Clean, single-field imaging benchmarks which are defined.

Returns
Bool

Example

```
bench:=benchmark()
bench.ccsf()
bench.done()
```

---

benchmark.ccsfvlacu125k - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the Clark Clean, single-field, VLA, 125k vis, uncompressed, continuum benchmarks

Synopsis
ccsfvlacu125k(imsizes, stokes, weight, spwid, niter, mode, nchan, start, step, fieldid, facets, tile, cache)

Description
Run the Clark Clean, single-field imaging benchmarks defined for the VLA dataset of 125k uncompressed continuum visibilities.

Returns
Bool

Arguments

imsizes	List of image sizes to be used for benchmarking Allowed:Vector of ints Default: [512,1024,2048]
stokes	List of stokes parameters to be used for benchmarking Allowed:Vector of 'I' and/or 'IQUV' Default: ['I','IQUV']
weight	List of weighting schemes to be used for benchmarking Allowed:Vector of 'uniform' and/or 'natural' Default: ['natural','uniform']
spwid	List of spectral windows to be used for benchmarking Allowed:Vector of ints Default: [1]
niter	The number of CLEAN iterations to be used Allowed:Integer Default: 1000
mode	Imaging mode Allowed:'mfs' or 'channel' Default: 'mfs'
nchan	Number of frequency channels to use Allowed:Integer Default: 1
start	Start channel number Allowed:Integer Default: 1
step	Step to use for stepping through channels Allowed:Integer Default: 1
fieldid	Field ID parameter of imager Allowed:Integer Default: 1
facets	Number of facets to be used for imaging Allowed:Integer Default: 1
tile	Tile size used internally by imager for PagedArrays Allowed:Integer Default: 16
cache	The cache size internally by imager for PagedArrays Allowed:Integer Default: 4194304 (4MB)

Example

```
bench:=benchmark()
bench.ccsfvlacu125k()
bench.done()
```

---

benchmark.ccsfvlacu1m - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the Clark Clean, single-field, VLA, 1M vis, uncompressed, continuum benchmarks

Synopsis
ccsfvlacu1m(imsizes, stokes, weight, spwid, niter, mode, nchan, start, step, fieldid, facets, tile, cache)

Description
Run the Clark Clean, single-field imaging benchmarks defined for the VLA dataset of 1M uncompressed continuum visibilities.

Returns
Bool

Arguments

imsizes	List of image sizes to be used for benchmarking Allowed:Vector of ints Default: [512,1024,2048]
stokes	List of stokes parameters to be used for benchmarking Allowed:Vector of 'I' and/or 'IQUV' Default: ['I','IQUV']
weight	List of weighting schemes to be used for benchmarking Allowed:Vector of 'uniform' and/or 'natural' Default: ['natural','uniform']
spwid	List of spectral windows to be used for benchmarking Allowed:Vector of ints Default: [1]
niter	The number of CLEAN iterations to be used Allowed:Integer Default: 1000
mode	Imaging mode Allowed:'mfs' or 'channel' Default: 'mfs'
nchan	Number of frequency channels to use Allowed:Integer Default: 1
start	Start channel number Allowed:Integer Default: 1
step	Step to use for stepping through channels Allowed:Integer Default: 1
fieldid	Field ID parameter of imager Allowed:Integer Default: 1
facets	Number of facets to be used for imaging Allowed:Integer Default: 1
tile	Tile size used internally by imager for PagedArrays Allowed:Integer Default: 16
cache	The cache size internally by imager for PagedArrays Allowed:Integer Default: 4194304 (4MB)

Example

```
bench:=benchmark()
bench.ccsfvlacu1m()
```

```
bench.done()
```

---

benchmark.ccsfvlalu125k - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the Clark Clean, single-field, VLA, 125K vis, 64 frequency channel, uncompressed, spectral line benchmarks

Synopsis
ccsfvlalu125k(imsizes, stokes, weight, spwid, niter, mode, nchan, start, step, fieldid, facets, tile, cache)

Description
Run the Clark Clean, single-field imaging benchmarks defined for the VLA dataset of 125K uncompressed spectral line visibilities.

Returns
Bool

Arguments

imsizes	List of image sizes to be used for benchmarking Allowed:Vector of ints Default: [512]
stokes	List of stokes parameters to be used for benchmarking Allowed:Vector of 'I' and/or 'IQUV' Default: ['I']
weight	List of weighting schemes to be used for benchmarking Allowed:Vector of 'uniform' and/or 'natural' Default: ['natural','uniform']
spwid	List of spectral windows to be used for benchmarking Allowed:Vector of ints Default: [1]
niter	The number of CLEAN iterations to be used Allowed:Integer Default: 1000
mode	Imaging mode Allowed:'mfs' or 'channel' Default: 'channel'
nchan	Number of frequency channels to use Allowed:Integer Default: 64
start	Start channel number Allowed:Integer Default: 1
step	Step to use for stepping through channels Allowed:Integer Default: 1
fieldid	Field ID parameter of imager Allowed:Integer Default: 1
facets	Number of facets to be used for imaging Allowed:Integer Default: 1
tile	Tile size used internally by imager for PagedArrays Allowed:Integer Default: 16
cache	The cache size internally by imager for PagedArrays Allowed:Integer Default: 4194304 (4MB)

Example

```
bench:=benchmark()
bench.ccsfvlalu125k()
```

```
bench.done()
```

---

benchmark.ccsfbi4nl - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the Clark Clean, single-field, BIMA, 139M vis, uncompressed, continuum benchmarks

Synopsis
ccsfbi4nl(imsizes, stokes, weight, spwid, niter, mode, nchan, start, step, fieldid, facets, tile, cache)

Description
Run the Clark Clean, single-field imaging benchmarks defined for the BIMA dataset of 139M uncompressed continuum visibilities.

Returns
Bool

Arguments

imsizes	List of image sizes to be used for benchmarking Allowed:Vector of ints Default: [256]
stokes	List of stokes parameters to be used for benchmarking Allowed:Vector of 'I' and/or 'IQUV' Default: ['I']
weight	List of weighting schemes to be used for benchmarking Allowed:Vector of 'uniform' and/or 'natural' Default: ['natural']
spwid	List of spectral windows to be used for benchmarking Allowed:Vector of ints Default: [1]
niter	The number of CLEAN iterations to be used Allowed:Integer Default: 1000
mode	Imaging mode Allowed:'mfs' or 'channel' Default: 'channel'
nchan	Number of frequency channels to use Allowed:Integer Default: 20
start	Start channel number Allowed:Integer Default: 40
step	Step to use for stepping through channels Allowed:Integer Default: 1
fieldid	Field ID parameter of imager Allowed:Integer Default: 1
facets	Number of facets to be used for imaging Allowed:Integer Default: 1
tile	Tile size used internally by imager for PagedArrays Allowed:Integer Default: 16
cache	The cache size internally by imager for PagedArrays Allowed:Integer Default: 4194304 (4MB)

Example

```
bench:=benchmark()
bench.ccsfbi4nL()
```

```
bench.done()
```

---

benchmark.fuvrd - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run all FITS UV read benchmarks

Synopsis
fuvrd()

Description
This function will run all FITS UV read benchmarks which are defined.

Returns
Bool

Example

```
bench:=benchmark()
bench.fuvrd()
bench.done()
```

---

benchmark.fuvrdvlacu125k - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the FITS Filler, VLA, 125k vis, uncompressed, continuum benchmarks

Synopsis
fuvrdvlacu125k(fitsin, compress)

Description
Run the FITS Filler benchmarks defined for the VLA dataset of 125k uncompressed continuum visibilities.

Returns
Bool

Arguments

fitsin	FITS file name Allowed:string Default: vlac125k.fits
compress	MS compression mode Allowed:Bool Default: F

Example

```
bench:=benchmark()
bench.fuvrdvlacu125k()
bench.done()
```

---

benchmark.fuvrdvlacu1m - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run FITS Filler, VLA, 1M vis, uncompressed, continuum benchmarks

Synopsis
fuvrdvlacu1m(fitsin, compress)

Description
Run FITS Filler benchmarks defined for the VLA dataset of 1M uncompressed continuum visibilities.

Returns

Bool

Arguments

fitsin	FITS file name Allowed:string Default: vlac1M.fits
compress	MS compression mode Allowed:Bool Default: F

Example

```
bench:=benchmark()
bench.fuvrdvlacu1m()
bench.done()
```

---

benchmark.fuvrdvlalu125k - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the FITS Filler VLA, 125K vis, 64 frequency channel, uncompressed, spectral line benchmarks

Synopsis

fuvrdvlalu125k(fitsin, compress)

Description

Run the FITS Filler benchmarks defined for the VLA dataset of 125K uncompressed spectral line visibilities with 65 channels.

Returns

Bool

Arguments

fitsin	FITS file name Allowed:string Default: vlal125K64Chan.fits
compress	MS compression mode Allowed:Bool Default: F

Example

```
bench:=benchmark()
bench.fuvrdvlalu125k()
bench.done()
```

---

benchmark.almafuvrd - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run all ALMA-TI almati2ms FITS UV read benchmarks

Synopsis
almafuvrd()

Description
This function will run all almati2ms ALMA-TI FITS UV read benchmarks which are defined.

Returns
Bool

Example

```
bench:=benchmark()
bench.almafuvrd()
bench.done()
```

---

benchmark.almatifuvrd - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the almati2ms FITS Filler, IRAM data, ALMA-TI format, uncompressed benchmarks

Synopsis
almatifuvrd(fitsin, compress)

Description
Run the almati2ms FITS Filler benchmarks defined for an IRAM dataset presented in ALMA-TI format. Expects to find dataset in /aips++/data/alma/test.

Returns
Bool

Arguments

fitsin	FITS file name Allowed:string Default: 07-feb-1997-g067-04.fits
compress	MS compression mode Allowed:Bool Default: F

Example

```
bench:=benchmark()
bench.almatifuvrd()
bench.done()
```

---

benchmark.calvla - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run all default VLA calibrator benchmarks

Synopsis
calvla()

Description
This function will run all the default VLA calibrator benchmarks.

Returns
Bool

Example

```
bench:=benchmark()
bench.calvla()
bench.done()
```

---

benchmark.calvlau27s10 - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the VLA calibrator benchmarks for G and D Jones

Synopsis
calvlau27s10(datafile, myms, jones, nsolint, nchan, start, step, tablename)

Description
Run the calibrator benchmarks defined for VLA uncompressed dataset with 27 antennas and 128 frequency channels.

Returns
Bool

Arguments

datafile	FITS file name Allowed:string Default: calvlac27s10.fits
myms	Name of the MS. Will bypass the fitstoms() operation Allowed:string Default: ' '
jones	List of Jones matrices to be solved for Allowed:array of strings Default: ['G','D']
nsolint	No. of solution intervals Allowed:integer Default: 100
nchan	No. of frequency channels to be used Allowed:integer Default: 1
start	The starting frequency channel number Allowed:integer Default: 1
step	Increment for iterating through frequency channels Allowed:integer Default: 1
tablename	Name of a calibration table to be applied before the solver is invoked Allowed:string Default: "

Example

```
bench:=benchmark()
bench.calvlac27s10()
bench.done()
```

---

benchmark.sdall - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the single dish benchmark tests

Synopsis
sdall()

Description
Run the single dish benchmark tests. This uses GBT data to test the timings of the import, scan access, scan plotting, calibration, averaging and baselining.

Returns
Bool

Example

```
bench:=benchmark()
bench.sdall()
bench.done()
```

---

benchmark.sdbench - Function

Package	**utility**
Module	**benchmark**
Tool	**benchmark**

Run the single dish benchmark tests for a particular data set

Synopsis
sdbench(code)

Description
Run the single dish benchmark tests for a specific data set. The available codes and matching data are:

- 42256 - Spectral Processor data - [17,18] - OffOn PSWITCH
- 121k - Spectral Processor data - [36,37] - Track FSWITCH
- 114k - Spectrometer data - [20,21] - OffOn PSWITCH
- 224k - Spectrometer data - [43,44] - OffOn PSWITCH
- 128k - Spectrometer data - [48,48] - Track FSWITCH

- 1216k - Spectrometer data - [22,23] - OffOn PSWITCH
- 12125k- Spectrometer data - [51,52] - OffOn PSWITCH

The code delineates:

- first number: number of IFs (1, 2 or 4)
- second number: number of polarizations (1 or 2)
- third number: number of channels (256, 1k, 4k, 8k, 16k, 125k)

Returns
Bool

Arguments

code	code for test Allowed:string Default: none (see above)

Example

```
bench:=benchmark()
bench.sdbench('42256')
bench.done()
```