NOTE 228 – AIPS++ Assay Manifesto

J. McMullin

January 7, 1999

A postscript version of this note is available (193kB).

Contents

1 Motivation

This note is an effort to consolidate the existing assay tests being performed by the chief tester, site managers and others to establish the state of an AIPS++ installation or release.

We also chart the improvements necessary to further improve these tests as AIPS++ moves into a broader public use.

2 Mission

Integration and unit testing are vital to the success of AIPS++.

3 Implementation

3.1 assay()

The assay class is used to track the results of running the test and demonstration functions. It is useful for verifying an installation. It can also assay any list of functions. The functions must not require arguments.

The functions to be tested should use the Glish fail mechanism for assay to be meaningful.

The standard tests and demonstrations are those that are expected to not fail in a working installation.

```
- include "assay.g"
T
```

```
- assay().trytests()
```

There are several implement functions under assay:

1. **try** Try each of a list of functions.

   For each function in the list, run the function (*without arguments*) and trap any fails. At the end, the failures and corresponding messages are logged, and the number of failures is returned.

   Note that the argument is a list of function *names*, not the functions themselves; it returns the integer number of functions which failed.

   ```
   - include "assay.g"
   T
   - assay().try('foo bar tabletest')
   foo failed:  argument must be defined
   bar failed:  argument must be defined
   2
   ```

2. **tests** Returns names of the standard tests.

3. **demos** Returns names of the standard demonstrations.

4. **trytests** Tries the standard tests. The arguments are the string names of the tests; it returns the integer number that failed.

5. **trydemos** Tries the standard demonstrations. The arguments are the names of the demos; it returns the number that failed.

3.2 Daily Testing – Site Managers

The philosophy is a build and smoke test. AIPS++ needs to be the primary authority on the current status of the system.

For AIPS++ site managers, examine each installation sneeze.log for 'fail' and 'error' and do the following:

1. a) assay().trytests().

2. b) Basic GUI testing (e.g., toolmanager).

3. c) Alert users and developers to failures.

4. d) Consider dedicated machine for assay tests.

5. e) Periodic deeper assay tests (as implemented).

3.3 Daily Testing – Chief Tester

The chief tester needs to perform more fundamental and deeper tests of the system. In particular, sneeze/inhale logs, unit testing reports and assay() results should all be logged.

3.3.1 sneeze/inhale logs

The **Chief Tester** should examine the sneeze.log on at least one machine of every supported architecture/OS:

1. (1) linux - muse in CV (check daily and weekly).
2. (2) sgi - argus in Socorro
3. (3) alpha - kaputar in Narrabri
4. (4) sun4sol - duw01 in Dwingeloo
5. (5) linux - sneffels in socorro (check daily and weekly)

Automatic filtering of the aips2-inhale exploder can help streamline the time spent in this review for different machines. The process is as follows:

1. create/modify .procmailrc file to filter the appropriate information. The following is an example:

```
# this is my new .procmailrc 11/16/99
PATH=/bin:/usr/bin:/usr/local/bin # use a minimal path.
MAILDIR=$HOME/mail # my mail directory
LOGFILE=$MAILDIR/procmail.log
TESTDIR=/home/charybdis_2/jmcmulli/computing/chief_tester/test_results
INHALEDIR=/home/charybdis_2/jmcmulli/computing/chief_tester/inhale_logs

# message is to aips2-inhale?
:0:
* ^To.*aips2-inhale
{

# get sneeze files from muse
:0:
* ^From.*aips2mgr@muse
```

```
$INHALEDIR/muse_linux

# get sneeze files from argus (SGI)
:0:
* ^From.*aips2mgr@zia
$INHALEDIR/argus_sgi

# get sneeze files from duw01 (sun4sol_gnu)
:0:
* ^From.*aips2mgr@nfra
$INHALEDIR/duw01_sun4sol

# get sneeze files from kaputar (Alpha)
:0:
* ^From.*aips2mgr@kaputar
$INHALEDIR/kaputar_alpha

# get test results summary of aips - linux
:0:
* ^Subject.*linux.*test.*results.*aips
$TESTDIR/aips_linux

        # get test results summary of trial - linux
        :0:
        * ^Subject.*linux.*test.*results.*trial
        $TESTDIR/trial_linux

        # get test results summary of aips - sun4sol
        :0:
        * ^Subject.*sun4sol.*test.*results.*aips
        $TESTDIR/aips_sun4sol

        # get test results summary of trial - sun4sol
        :0:
        * ^Subject.*sun4sol.*test.*results.*trial
        $TESTDIR/trial_sun4sol

# dump all the rest
:0:
/dev/null
```

```
}
```

2. add .forward file with the following contents:

```
"|/usr/local/bin/procmail -f-"
```

3. add user name to aips2-inhale.list in /aips++/daily/code/admin/system.

The generated files are then examined (grep -i fail/error) each morning and the Assay Checklist sections are filled in appropriately (see Section x).

3.4 Weekly Unit Testing

C++ unit tests are performed every week in both Socorro and NFRA on both the aips and trial modules. Tests are run over the weekend and results are mailed to either aips2-modulebosses (Socorro:aips results, NFRA:aips and trial) or aips2-inhale (Socorro:aips and trial results).

The script which performs this task is testsuite (/aips++/daily/$ARCH/bin/testsuite) which runs runtests (also /aips++/daily/$ARCH/bin/runtests). runtests writes a series of reports in /aips++/daily/$ARCH/bintest/ for each module and then a summary file called runtests.report which is parsed by testsum.sh to write the report which is mailed to aips2-lib, etc.

```
#
# Thursday's daily cumulative inhale is managed by testsuite.
# -----------------------------------------------------------
30 19 * * 4 (. /aips++/daily/.profile ; gmake -C /aips++/daily/code megaunlink
; /aips++/daily/sun4sol_egcs/bin/testsuite -c ; gmake -C /aips++/daily/code
megalink) 2>&1 | tee /aips++/daily/testsuite.log | mailx -s "tarzan
testsuite.log" aips2-inhale@nrao.edu aips2mgr; mailx -s "sun4sol_egcs test
results summary - aips" aips2-modulebosses aips2-inhale <
/aips++/daily/sun4sol_egcs/bintest/runtests_aips.summary; mailx -s
"sun4sol_egcs test results summary - trial" aips2-inhale <
/aips++/daily/sun4sol_egcs/bintest/runtests_trial.summary
#
```

These reports are also dumped to files on disk and reviewed each morning if available.

3.4.1 assay()

assay().trytests() should be run daily. A simple crontab automates this procedure:

```
#!/opt/local/bin/bash
#
# set up display to prevent tests from failing
export DISPLAY=0:0
# get rid of procmail.log
rm $HOME/mail/procmail.log
#
cd /home/charybdis_2/jmcmulli/computing/chief_tester/assay
# get rid of old files
rm -r *.plot image* 3C*
source /aips++/daily/aipsinit.sh
aips++ -l ct.g 2>&1 | tee ct.log
```

where, ct.g is:

```
#
#
#
#include 'assay.g'
numfail:=assay().trytests()
print 'Number of fails is: ',numfail;
memoryassay();
mymemoryassay := function(verbose=F) {
  names := symbol_names();
  memory := [];
  j:=0;
  for (i in 1:length(names)) {
    name:=names[i];
    list[i] := name;
    memory[i] := eval(spaste('sizeof(',name,')'));
  }
  list:=sort_pair(memory, list);
  memory:=sort(memory);
  if(verbose) {
    for (i in 1:length(list)) {
      print list[i], memory[i];
```

```
    }
  }
  return sum(memory);
}

fp:=open(">> /home/charybdis_2/jmcmulli/computing/chief_tester/assay/memory.hist");
a:=shell('date');
b:=mymemoryassay();
mem_string:=spaste(a,' -- ',as_string(b));
ok:=fprintf(fp,'%s\n',mem_string);

exit
```

Time for startup, total memory of glish variables and any fails/errors should be logged as required and mailed to aips2-lib. The total memory in glish variables is also automatically logged to memory.hist.

3.5 Stable Release Requirements

Requirements for stable release:

1. a) cumulative inhale occurs without error
2. b) all aips package test programs pass and all of the trial package test programs which passed for the previous stable contiue to pass (or they no longer exist).
3. c) the documentation builds
4. d) assay().trytests() passes
5. e) the help system works (insure that the browser will be driven to the appropriate page from several Help menus).

   This can be complicated by a recent problem encountered in NRAO-CV in which a linux machine would neither spawn a browser when necessary nor drive an existing browser. The problem results from the fact that different systems in that location use either the standard netscape script for the netscape command or a binary executable. The script accepts the 'remote' flag which is what is used by the help menus while the executable happily ignores it. How do we deal with this?
6. f) dish works

Both e) and f) require a checklist of user interactions in the absence of an automated cron job.

In addition, there are a number of check-in protocols which should be adhered to. Maintaining the stability of daily is a high priority for everyone in the project. Breaking daily affects the work of a large number of people. Maintaining the stability of weekly is essential. Breaking weekly is disastrous for the work of a large number of people. Our goal should be that every weekly can be marked as stable. Checkins of changes to existing code should be made with the above two goals in mind.

1. a) Changes to code that is used by others must be well-tested on the project compiler (egcs 1.1.2) before checkin. Checkin of purely developmental code is less crucial but still good habits should be followed.
2. b) All checkins except simple ones, and bug fixes should be done early in the week. Early means with two more of your inhales to get it right before the final weekly.
3. c) If you have a *really* substantial change, alert others in the project via your weekly report beforehand.
4. d) All development sites should be running a daily using the project compiler.

3.6 Release Testing Checklist

Tasks for all supported platforms/OS revisions.

1. a) Install on pristine machine.
2. b) Run from CDROM (including GUI tests, e.g., toolmanager).
3. c) Run from disk installation.
4. d) assay().trytests() at a high level.
5. e) Test documentation for missing links using htdig.
6. f) Standardized walk-through of key documentation links.
7. g) Review release notes and readme documents for accuracy.
8. h) Review CDROM label and cover information.
9. i) Verify advertised patches work correctly.
10. j) Log release testing results to a mail exploder for future reference.

3.6.1 Patch Testing

1. a) Detailed checklist of patch contents.

2. b) Verified all items on list.

4 Future Imperatives – Nov 99 – May 00

4.1 Recently completed

1. • List of required hardware and third party software.

   Completed: 11/99

   Hardware requirements:

   (a) • Sparcstation IPX-class or better (preferably at least Sparc 20 or a Pentium class machine.

   (b) •SunOS 5, ideally SunOS 5.5 (a.k.a. Solaris 2.5) or higher; Linux 2.0 or 2.1. We expect distributions for HPUX, SGI and Dec Alpha (OSF1) to follow).

   (c) • 64 Megabytes of memory (32 MB is workable but the system swaps most of the time).

   (d) • At least 250 Megabytes of disk space (not counting space for data).

   (e) • At least 100 Megabytes of swap space.

   3rd Party Software requirements (README file from pub/import on aips2.nrao.edu):

```
bison       Compiler compiler
            When:    1998/06/16 JAU
            Version: 1.25
            Source:  prep.ai.mit.edu:/pub/gnu
            Files:   bison-1.25.tar.gz
            Usage:   Needed for compiling the Tables module.

BLAS        See LAPACK

dvips       Program to convert documents from dvi format to PostScript
```

```
             When:    1992/06/03
             Version: 5.490
             Source:  labrea.stanford.edu:/pub
             Files:   dvips-5.490.tar.gz
             Usage:   Used for optional production of AIPS++ PostScript
                      documentation.

flex         Lexical analyzer generator
             When:    1998/06/16 JAU
             Version: 2.5.4a
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   flex-2.5.4a.tar.gz
             Usage:   Needed for compiling the Tables module.

ghostscript  GNU "PostScript" interpreter
             When:    1998/06/16 JAU
             Version: 4.03
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   ghostscript-4.03.tar.gz
ghostscript-fonts-std-4.03.tar.gz
             Usage:   Used by 'ghostview'.

ghostview    GNU "PostScript" previewer
             When:    1998/06/16 JAU
             Version: 1.5
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   ghostview-1.5.tar.gz
             Notes:   Requires 'ghostscript'.
             Usage:   Optionally used to preview/print selected pages from
                      PostScript files.

gmake        GNU make.
             When:    1998/06/16 JAU
             Version: 3.76.1
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   make-3.76.1.tar.gz
             Notes:   Installed as 'gmake' to distinguish it from the version
                      of 'make' supplied with unix operating system.
             Usage:   Required to rebuild AIPS++ from its sources.
Installed automatically if necessary by 'configure'
```

```
when AIPS++ is first installed.

gzip         GNU file compression and expansion utility
             When:    1994/04/15
             Version: 1.2.4
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   gzip-1.2.4.tar
             Usage:   Required for installing AIPS++ from compressed tar
                      files.  Installed automatically if necessary by
                      'configure' when AIPS++ is first installed.

LAPACK       FORTRAN library of commonly used numerical algorithms
                      (encompasses LINPACK and EISPACK).
             When:    1998/06/16 JAU
             Version: 2.0
             Source:  netlib.org:/lapack
             Files:   lapack.tar.gz
             Notes:   Now includes the BLAS library.
             Usage:   Required.

LaTeX        LaTeX document preparation system.
             When:    1995/03/06
             Version: 2e
             Source:  ftp.Uni-Mainz.DE:/pub/tex/macros/latex
             Files:   latex/{base,packages,contrib}.tar.gz
             Usage:   Optionally used for compilation of AIPS++ printable
                      documentation.

latex2html   LaTeX to HTML document converter.
             When:     1998/08/08
             Version:  98.2 beta 5
             Source:   Released:
       ftp://www-dsed.llnl.gov/files/programs/unix/latex2html/
             Source:   Beta:
                  http://cdc-server.cdc.informatik.th-darmstadt.de/~latex2html/
             Files:    l2h98_2beta5.tar.gz
             Notes:    Make sure all the utilities are up-to-date that are
                       needed by latex2html.
             Usage:    Optionally used to generate AIPS++ web documentation.
```

```
aips2l2h.patch  Patch for latex2html
             When     1998/09/23
             Version  1.
             Source   /home/sneffels2/wyoung/latex2html
             Files    aips2l2h.patch
             Notes    This is an aips++ specific kludge for generating the
                      User's reference manual.
             Usage    typically patch -p0 <aips2l2h.patch

make         See gmake

perl         The perl shell programming language
             When:    1998/06/16 JAU
             Version: 5.004_04
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   perl-5.004_04.tar.gz
             Usage:   Required for the document extractor, 'cxx2html'.

PGPLOT       Plotting package written by Tim Pearson
             When:    1998/06/16 JAU
             Version: 5.2
             Source:  astro.caltech.edu:/pub/pgplot
             Files:   pgplot5.2.tar.gz
             Usage:   Required.  Currently used for AipsView, Glish, and
                      related applications.

rcs          Revision Control System for source code management
             When:    1995/08/21
             Version: 5.7
             Source:  prep.ai.mit.edu:/pub/gnu
             Files:   rcs-5.7.tar.gz
             Usage:   Required for code management at AIPS++ development
                      sites.

tcl/Tk       Tool Control Language
             When:    1998/06/22
             Version: 8.0 (patch level 2)
             Source:  http://www.scriptics.com/
             Files:   tcl8.0p2.tar.gz   tk8.0p2.tar.gz
             Usage:   Required by Glish and Aipsview
```

```
tex          The TeX document formating system
             When:    -
             Version: 3.14
             Source:  byron.u.washington.edu:/pub/unixtex
             Files:   unixtex subdirectory
             Usage:   Optionally used for compilation of AIPS++ printable
                      documentation.
             Notes:   All AIPS++ documents are available in PostScript form
                      from aips2.cv.nrao.edu:/pub/aips++/docs.
```

2. • guidelines for the writing of a glish test script.

   The guidelines for testing glish scripts are broken into two parts. The first part can be run automatically and performs tests on all public functions (we assuming that private functions will be effectively tested through the exercise of the public functions which utilize them); the procedures for incorporating a generic set of tests is outlined below (and based on the model used in deconvolver). The second part of testing is a checklist of items for testing the GUI elements of a given task; this list is under development (1/5/00).

   (a) a) Each glish script will require a test function, e.g. task.g will have a function 'tasktest'. Within this function all public methods should be tested. The first line of this function will be 'stime := time();' so that the test procedures may be timed for performance.

   (b) b) If data is being manipulated, a test directory should be setup for putting test data and manipulating it. Note that this directory must also be cleaned up before running each time.

   (c) c) A 'checkresult' function will be included within 'tasktest'. This function will look as follows:

```
  checkresult := function(ok, ntest, nametest, ref results) {
   if(is_fail(ok)) {
   results[ntest] := paste("Test", ntest, " on ", nametest,
"failed ", ok::message);
    }
     else if(is_boolean(ok)) {
       if(ok) {
```

```
          results[ntest] := paste("Test", ntest, " on ",
nametest, "succeeded");
        }
        else {
          results[ntest] := paste("Test", ntest, " on ",
nametest, "failed ", ok::message);
        }
      }
      else {
        results[ntest] := paste("Test", ntest, " on ", nametest,
"returned", ok);
      }
  }
```

(d) d) Each public function will then be tested as:

```
nametest := 'taskmethod';
note('Test task.method');
ntest+:=1
ok:=task.method(arg1,arg2,etc);
checkresult(ok,ntest,nametest,results);
```

After all tests have been run:

```
etime:=time();
```

With the final results posted as:

```
for (result in results) {
note(result);
}
note('##  task.tasktest()');
note('## Finished in run time = ',(etime-stime),' seconds');
```

Completed: Initial version 1/5/00

3. • successful completion of all glish test scripts in aips.

   Completed: Generally at 100% for aips test scripts.

4. • test script for 50% of the glish functions in trial.

   Completed: Currently we achieve ∼80%

4.2 To do:

5. • test script for every glish function in aips.

   Completed:

6. • DISH test scripts.

   Completed:

7. • test programs for all classes in the aips package.

   Completed:

8. • assay level specification (0,...n). Highest level includes largest tests (e.g., WF, MF).

   Completed:

9. • assay test areas (e.g., imaging, calibration, ...) as second axis against test level (0,...n), including an 'all' option.

   Completed:

10. • standardized GUI test scripts requiring specific GUI interactions.

    Completed:

A Modules tested by assay.g

Test Script dir	Functions Tools Constructors	Whatis Missing
tcomponentlist.g:componentlisttest		
trial/implement/ComponentModels/test	emptycomponentlist(C)	
	is_componentlist	Functions
	type	sort
	length	sample
	simulate	edit
	setshape	select
	shapetype	deselect
	convertshape	selected
	getshape	getlabel
	rename	setlabel
	close	getfluxvalue
	done	getfluxunit
	componentlist(T)	getfluxpol
	indices	setflux
	remove	convertfluxunit
	recover	convertfluxpol
	purge	getrefdir
		getrefdirra
		getrefdirdec
		getrefdirframe
		convertrefdir
		spectrumtype
		getspectrum
		setspectrum
		getfreq
		getfreqvalue
		getfrequnit
		getfreqframe
		setfreq
		setfreqframe
		convertfrequnit
		Tools
		componenteditor
		Constructors
		asciitocomponentlist
deconvolver.g:deconvolverlongtest		

trial/apps/deconvolver	deconvolver	
	boxmask	
	clarkclean	
	setscales	
	clean	
	ft	
	make	
	smooth	
	makeprior	
	mem	
	mem-with-prior	
	mem-with-mask	
	residual	
	restore	
	summary	
	clipimage	
	open	
	done	
fftserver.g:fftservertest		
trial/implement/Mathematics	complexfft	real test
	realtocomplexfft	
	convolve	
	crosscorr	
	autocorr	
	shift	
	mfft	
	done	
guientry.g:guientrytest		
trial/implement/Widgets	guientry()	
	range	
	boolean	
	booleanarray	
	array	
	scalar	
	choice	
	check	
	file	
	string	
	quantity	

	record	
	region	
	epoch	
	direction	
	position	
imager.g:imageralltests		
imager.g:imagertest		
trial/apps/imager/	imager(C)	imagertester(C)
		open
		setjy
	setimage	
	setoptions	
	setdata	
	summary	
	make	
	weight	
	filter	
	image	
	fitpsf	
	close	
	done	
	boxmask	
	clean	
	ft	
	plotvis	
	restore	
	smooth	
	mask	
imager.g:imagerlongtest	setimage	
	advise	
	setdata	
	setoptions	
	plotsummary	
	plotuv	
	plotvis	
	plotweights	
	fitpsf	
	setbeam	
	weight	
	uvrange	

	filter
	make
	boxmask
	regionmask
	smooth
	exprmask
	image
	setscales
	clean
	n
	restore
	residual
	clipimage
	sensitivity
	ft
	correct
	clipvis (in doc)
	done
imager.g:imagermftest	setimage
	make
	clean
	plotvis
	close
	done
imager.g:imagerpbtest	setimage
	setdata
	setvp
	make
	weight
	filter
	setbeam
	clean
	close
	done
imager.g:imagerselfcaltest	setimage
	setdata
	clean
	clipimage
	selfcal
	close

	done
imager.g:imagercomponenttest	setimage
	setdata
	summary
	make
	ft
	plotvis
	clean
	close
	done
imageservertest.g:imageservertest	
trial/implement/Images/ (tests 1-26)	imagefromarray(C)
	flush (not in doc)
	close
	open
	rename
	delete
	done
	image(C)
	shape
	name
	boundingbox
	summary
	stats
	histo
	coord
	coordinates
	getregion
	putregion
	subimage
	moments
	hanning
	padimage
	getchunk
	putchunk
	setmiscinfo
	miscinfo
	tofits
	imagefromfits(C)

	imagefromshape(C)	
	subimage(C)	
	imageconcat(C)	
	imagecalc(C)	
	statistics	
	is_image	
	imagetools	
	type	
	id	
	lock	
	unlock	
	set	
	replacemaskedpixels	
	coordmeasures	
	setfrequencysystem	
	maskhandler	
	history	
	arrconvolve	
	sepconvolve	
	histograms	
	fitsky	
	gausstopixel	
	view	
	maskhandlergui	
	momentsgui	
	sepconvolvegui	
interpolate1d.g:interpolate1dtest		
trial/implement/Mathematics	initialize	(a real test)
	interpolate	
	setmethod	
pgplotter.g:pgplottertest	pgplotter(C)	(a real test)
trial/implement/Guiutils	busy	commandgui
	demo	editgui
	done	addtool
		tool
		(not in doc...)
		gui
		screen

		plotfile
		restore
		type
		(implicit calls)
		postscript(W)
		clear(W)
		record(W)
		displaylist(W)
		refresh(W)
		lastchange(W)
		message(W)
		shortnametofullname
polyfitter.g:polyfittertest	fit	(a real test)
trial/implement/Mathematics	multifit	
	eval	
randomnumbers.g:randomnumberstest	normal	(a real test)
trial/implement/Mathematics	reseed	
ttable.g:tabletest	tablecreatescalarcoldesc	
trial/apps/table	tablecreatearraycoldesc	
	tablecreatedesc	
	table(C)	tablefromfits(C)
	getdesc	iswritable
	flush	lock
	putinfo	unlock
	info(not in doc)	
	addreadmeline	hassdatachanged
	tableinfo	haslock
	colnames	lockoptions
	getcoldesc	ismultiused
	addcols	browse
	name	selectrows
	tableopentables	summary
	nrows	rownumbers
	addrows	setmaxcachesize
	putcol	isscalarcol
	getcol	coldatatype
	getcell	colarraytype

putcell	ncols
putkeyword	removerows
getkeyword	getcellslice
getkeywords	getcolslice
putkeywords	putcellslice
keywordnames	(not in doc)
removekeyword	putcolslice
fieldnames	(not in doc)
putcolkeyword	getcolshapestring
getcolkeyword	removecolkeyword
getcolkeywords	ok
putcolkeywords	
colkeywordnames	(not in doc)
tableiterator(C)	
next	reset
table	
terminate	
tablerow(C)	
get	set
put	
close	
query	
close	
tableexists	is_table
tablecommand	tabledemo
tableiswritable	tablerename
tablecopy	newtb
tabledelete	
closeallTables	(not in doc)
tablefromascii(C)	
toascii	

B assay.g

```
# assay.g: a glish convenience script for all tests and demos
# Copyright (C) 1996,1997,1998,1999,2000,2001
# Associated Universities, Inc. Washington DC, USA.
#
# This library is free software; you can redistribute it and/or modify it
```

```
# under the terms of the GNU Library General Public License as published by
# the Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.
#
# This library is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
# FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Library General Public
# License for more details.
#
# You should have received a copy of the GNU Library General Public License
# along with this library; if not, write to the Free Software Foundation,
# Inc., 675 Massachusetts Ave, Cambridge, MA 02139, USA.
#
# Correspondence concerning AIPS++ should be addressed as follows:
#        Internet email: aips2-request@nrao.edu.
#        Postal address: AIPS++ Project Office
#                        National Radio Astronomy Observatory
#                        520 Edgemont Road
#                        Charlottesville, VA 22903-2475 USA
# $Id: assay.latex,v 19.0 2003/07/16 03:50:40 aips2adm Exp $
#

pragma include once

include "general.g";
include "utility.g";
include "synthesis.g";
include "note.g";

# Runs each function in the list and checks for a fail.
# Tracks the number of fails and reports at the end.
# Returns the number of fails.

  const assay:=function() {

    self.tests:=['display_multi_column_list_test',
 'display_multi_column_text_test',
 'fftservertest',
 'imageralltests',
 'imagetest',
```

```
 'interpolate1dtest',
                 'pgplottertest',
 'polyfittertest',
 'randomnumberstest',
 'tabletest'];

    self.demos:=['display_multi_column_list_demo',
 'display_multi_column_text_demo',
 'fftserverdemo',
 'imagedemo',
 'inputsdemo',
 'interpolate1ddemo',
                 'pgplottertest',
 'polyfitterdemo',
 'randomnumbersdemo',
 'tabledemo'];

    const self.dontfail:=function(f) {
      if(!is_defined(f)) return 'argument must be defined';
      if(!is_string(f)) return 'argument must be a string';
      if(!is_function(eval(f))) return paste(f, 'is not a function');
      result:=eval(spaste(f,'()'));
      if(is_fail(result)) {
print f, 'fails: ', result::message;
        return result::message;
      }
      return '';
    }

    const public.try:=function(functionlist) {
      if(!is_string(functionlist)) fail "Need list of functions";
      if(functionlist=='') fail "Need list of functions";
      funclist:=split(functionlist);
      messages:=array('', len(funclist));
      for (i in 1:len(funclist)) {
messages[i]:=self.dontfail(funclist[i]);
      }
      failed:='';
      numberfailed:=0;
      for (i in 1:len(funclist)) {
```

```
if(messages[i]!='') {
  note(paste(funclist[i], 'failed: ', messages[i]));
  numberfailed+:=1;
}
      }
      return numberfailed;
    }

# Print tests
    const public.tests:=function() {return self.tests;}

# Print demos
    const public.demos:=function() {return self.demos;}

# Assay all tests
    const public.trytests:=function(tests=F){
      if(is_string(tests)&&strlen(tests)) {
return public.try(tests);
      }
      else {
return public.try(self.tests);
      }
    }

# Assay all demos
    const public.trydemos:=function(demos=F){
      if(is_string(demos)&&strlen(demos)) {
return public.try(demos);
      }
      else {
return public.try(self.demos);
      }
    }
    const public.all:=function() {return trytests()+trydemos();};

    const public.type:=function() {return "assay";};

    return ref public;

  }
```

C assay_meta.g

```
# assay_meta.g: Standard meta information for assay
#
#   Copyright (C) 1996,1997,1998,1999,2000,2001
#   Associated Universities, Inc. Washington DC, USA.
#
#   This program is free software; you can redistribute it and/or modify it
#   under the terms of the GNU General Public License as published by the Free
#   Software Foundation; either version 2 of the License, or (at your option)
#   any later version.
#
#   This program is distributed in the hope that it will be useful, but WITHOUT
#   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
#   FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for
#   more details.
#
#   You should have received a copy of the GNU General Public License along
#   with this program; if not, write to the Free Software Foundation, Inc.,
#   675 Massachusetts Ave, Cambridge, MA 02139, USA.
#
#   Correspondence concerning AIPS++ should be addressed as follows:
#          Internet email: aips2-request@nrao.edu.
#          Postal address: AIPS++ Project Office
#                          National Radio Astronomy Observatory
#                          520 Edgemont Road
#                          Charlottesville, VA 22903-2475 USA
#
#   $Id: assay.latex,v 19.0 2003/07/16 03:50:40 aips2adm Exp $
#

pragma include once;

include 'types.g';

types.class('assay').includefile('assay.g');

types.method('ctor_assay');
```

```
types.method('try').string('functionlist');

types.method('tests').string('return');
types.method('demos').string('return');
types.method('trytests').string('tests').integer('return');
types.method('trydemos').string('demos').integer('return');
types.method('all').integer('return');
```

D checker.g

```
# checker.g: check the environment of AIPS++
#
#   Copyright (C) 1996,1997,1998,1999,2000,2001
#   Associated Universities, Inc. Washington DC, USA.
#
#   This program is free software; you can redistribute it and/or modify it
#   under the terms of the GNU General Public License as published by the Free
#   Software Foundation; either version 2 of the License, or (at your option)
#   any later version.
#
#   This program is distributed in the hope that it will be useful, but WITHOUT
#   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
#   FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for
#   more details.
#
#   You should have received a copy of the GNU General Public License along
#   with this program; if not, write to the Free Software Foundation, Inc.,
#   675 Massachusetts Ave, Cambridge, MA 02139, USA.
#
#   Correspondence concerning AIPS++ should be addressed as follows:
#          Internet email: aips2-request@nrao.edu.
#          Postal address: AIPS++ Project Office
#                          National Radio Astronomy Observatory
#                          520 Edgemont Road
#                          Charlottesville, VA 22903-2475 USA
#
#   $Id: assay.latex,v 19.0 2003/07/16 03:50:40 aips2adm Exp $
```

```
#

pragma include once

include 'note.g';

const checker := function() {

  private := [=];

  public.gui := function() {
    wider private;
    warnings := 0;
    if(!have_gui()) {
      warnings +:= 1;
      note('Glish cannot display a GUI: no GUIs will be available',
   priority='WARN');
      note('Recommend checking value of environment variable DISPLAY',
   priority='WARN');
      note('Also check if \'xhost +\' needs to be run',
   priority='WARN');
    }
    return warnings;
  }

  public.display := function() {

    wider private;
    warnings := 0;
    if(strlen(shell('xwininfo -root | grep TrueColor'))) {
      warnings +:= 1;
      note('X server is running in TrueColor: display via aipsview or viewer will not work',
   priority='WARN');
      note('Recommend restarting X server with e.g. \'startx -- -bpp 8\'',
   priority='WARN');
    }
    return warnings;
  }

  public.plotter := function() {
```

```
    wider private;
    warnings := 0;
    if(0) {
      if(!(is_defined('pgplot')||!is_function(pgplot))) {
warnings +:= 1;
note('pgplot is not available: plotting will not work', priority='WARN');
      }
    }
    found := F;
    include 'os.g';
    for (field in "PGPLOT_FONT PGPLOT_DIR") {
      if(has_field(environ, field)&&dos.fileexists(environ[field])) {
        found := T;
if(field=='PGPLOT_DIR') {
  if(!dos.fileexists(spaste(environ.PGPLOT_DIR, '/grfont.dat'))) {
    warnings +:= 1;
    note('The directory pointed to by the pgplot environment PGPLOT_DIR does not contain grfont.dat: labelling on plots will not be available', priority='WARN');
    note('Recommend setting PGPLOT_DIR to location of pgplot grfont.dat file',
 priority='WARN');
  }
  break;
}
      }
    }
    if(!found) {
      warnings +:= 1;
      note('Neither pgplot environment variables PGPLOT_FONT, PGPLOT_DIR are not defined: labelling on plots will not be available', priority='WARN');
      note('Recommend setting PGPLOT_FONT to location of pgplot grfont.dat file',
   priority='WARN');
    }
    return warnings;
  }

  public.memory := function() {
    wider private;
    include 'sysinfo.g';
    warnings := 0;
    if(sysinfo().memory()<63) {
      note('The physical memory on your system is too low for optimum performance of AIPS++',
```

```
   priority='WARN');
      warnings +:= 1;
    }
    return warnings;
  }

  public.perl := function() {
    warnings := 0;
    if(strlen(which_client('perl')) == 0){
      note('Perl is not in your path.  The AIPS++ help will not be available to you.',
           priority='WARN')
      warnings +:= 1;
    }
    return warnings;
  }

  public.all := function() {
    wider public;
    warnings := public.gui() + public.display() + public.memory() + public.plotter()
                + public.perl();
    if(warnings) {
      note('Environment is not optimum for AIPS++', priority='WARN');
    }
    else {
      note('Environment is suitable for AIPS++');
    }
    return warnings;
  }

  public.type := function() {
    return 'checker';
  }

  return public;
}

const defaultchecker := const checker();
const dch := const defaultchecker;
note('defaultchecker (dch) ready', priority='NORMAL', origin='checker.g');
```

E memoryassay.g

```
# memoryassay.g: Assay of memory
#
#   Copyright (C) 1996,1997,1998,1999,2000,2001
#   Associated Universities, Inc. Washington DC, USA.
#
#   This program is free software; you can redistribute it and/or modify it
#   under the terms of the GNU General Public License as published by the Free
#   Software Foundation; either version 2 of the License, or (at your option)
#   any later version.
#
#   This program is distributed in the hope that it will be useful, but WITHOUT
#   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
#   FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for
#   more details.
#
#   You should have received a copy of the GNU General Public License along
#   with this program; if not, write to the Free Software Foundation, Inc.,
#   675 Massachusetts Ave, Cambridge, MA 02139, USA.
#
#   Correspondence concerning AIPS++ should be addressed as follows:
#          Internet email: aips2-request@nrao.edu.
#          Postal address: AIPS++/ Project Office
#                          National Radio Astronomy Observatory
#                          520 Edgemont Road
#                          Charlottesville, VA 22903-2475 USA
#
#   $Id: assay.latex,v 19.0 2003/07/16 03:50:40 aips2adm Exp $
#

pragma include once;
include 'note.g'

memoryassay := function(verbose=F) {
  names := symbol_names();
  memory := [];
  j:=0;
  for (i in 1:length(names)) {
```

```
    name:=names[i];
    list[i] := name;
    memory[i] := eval(spaste('sizeof(',name,')'));
  }
  list:=sort_pair(memory, list);
  memory:=sort(memory);
  if(verbose) {
    for (i in 1:length(list)) {
      print list[i], memory[i];
    }
  }
  note('Total memory used in Glish variables = ', sum(memory), ' bytes');
  return T;
}
```

F testsuite

```
#!/bin/sh
#-----------------------------------------------------------------------------
# testsuite: Run tests on the latest AIPS++ update
#-----------------------------------------------------------------------------
#
#   Copyright (C) 1992-1997,1998,2000,2001
#   Associated Universities, Inc. Washington DC, USA.
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

```
#
#   Correspondence concerning AIPS++ should be addressed as follows:
#          Internet email: aips2-request@nrao.edu.
#          Postal address: AIPS++ Project Office
#                          National Radio Astronomy Observatory
#                          520 Edgemont Road
#                          Charlottesville, VA 22903-2475 USA
#
#-----------------------------------------------------------------------------
# Usage: testsuite [-i] [-m module ] [-p package]
#-----------------------------------------------------------------------------
# testsuite may optionally run an inhale and then do a gmake runtests.
# It is invoked regularly as a cron job, on aips2.nrao.edu.
#
# Options:
#   -i     do a gmake runtests after doing an inhale
#   -m     do a gmake runtests for a specfic module
#   -p     do a gmake runtests for a specfic package
#          the -m and -p options maybe combined
#
# Status return values correspond to the various phases of the operation
#    standard testsuite status
#
# Notes:
#=============================================================================
# Initialize
#-----------------------------------------------------------------------------
# Guard against being overwritten.
  case $0 in
  *-)
     ARGS=$*
     ;;
  *)
     cp $0 $0-
     exec $0- $*
     ;;
  esac

# Identify ourselves.
  HOST=`uname -n | sed -e 's/\..*//'`
```

```
  LOGNAME=${LOGNAME-$USER}

  echo ""
  echo "testsuite: Execution on $HOST by $LOGNAME commenced at"
  echo "testsuite: `adate`"
  echo "testsuite: `adate -l`"

  echo ""
  echo "testsuite: Command line arguments \"$*\""

# Check that AIPSPATH is defined.
  if [ "$AIPSPATH" = "" ] ; then
     exec 1>&2
     echo ""
     echo "testsuite: AIPSPATH is undefined, abort!"
     exit 1
  fi

# Get the site name.
  SITE=`echo $AIPSPATH | awk '{print $3}'`

# Produce a status report.
  echo ""
  echo "testsuite: Current status of the $SITE installation"
  avers -l

# Determine the AIPS++ root directory.
  AIPSROOT=`echo $AIPSPATH | awk '{print $1}'`

# Check the accessibility of the AIPSROOT area.
  if [ ! -d "$AIPSROOT" ]
  then
     exec 1>&2
     echo ""
     echo "testsuite: The AIPS++ root directory does not exist or is"
     echo "           not accessible, $AIPSROOT"
     exit 1
  fi

# Get account information.
```

```
  AIPS2MGR=`getrc -i account.manager    2> /dev/null`
  AIPS2PRG=`getrc -i account.programmer 2> /dev/null`
  MGRACCNT=`echo $AIPS2MGR | sed -e 's/\..*//'`
  MGRGROUP=`echo $AIPS2MGR | sed -e 's/.*\.//'`
  PRGGROUP=`echo $AIPS2PRG | sed -e 's/.*\.//'`

# Check that this script is being run from the right account.
#  if [ "$MGRACCNT" != "$LOGNAME" -a "$MGRACCNT" != "" ]
#  then
#     exec 1>&2
#     echo ""
#     echo "testsuite: This script can only be run by $MGRACCNT, not" \
#          "$LOGNAME, abort!"
#     exit 1
#  fi

# Set default file creation mode.
  umask 002

# Parse command input.
  MODE=incremental
  HYBRID=
  DINKUM=
  DOCSYS=docsys
  TARGETS=runtests
  LATCHKEY=
  INHALE=0
  DEFMODULES=0
  DEFPACKAGES=0
  PACKAGES=""
  MODULES=""
  HASMODULES=0
  HASPACKAGES=0
  INHALE_FLAG="-m"

  while [ "$#" -gt 0 ]
  do
     case $1 in
     -c)
        INHALE_FLAG="-c"
```

```
    INHALE=1
    ;;
-i)
    INHALE=1
    ;;
-m*)
    DEFMODULES=1
    DEFPACKAGES=0
    HASMODULES=1;
    case $1 in
    -m)
       shift
       MODULES="$1"
       ;;
    *)
       MODULES=`echo $1 | sed -e 's/^-m//'`
       ;;
    esac
    ;;
-p*)
    DEFMODULES=0
    DEFPACKAGES=1
    HASPACKAGES=1;
    case $1 in
    -p)
       shift
       PACKAGES="$1"
       ;;
    *)
       PACKAGES=`echo $1 | sed -e 's/^-p//'`
       ;;
    esac
    ;;
*)
    if [ $DEFPACKAGES -gt 0 ] ; then
       PACKAGES="$PACKAGES $1"
    elif [ $DEFMODULES -gt 0 ] ; then
       MODULES="$MODULES $1"
    else
       exec 1>&2
```

```
        echo ""
        echo "Usage: testsuite [-i] [-m module names] [-p package names]"
        exit 1
     fi
     ;;
  esac

  [ "$2" = "" ] && break

  shift
done


# Report the amount of free space available for the update.
  echo ""
  echo "testsuite: Available space in the /tmp and $AIPSROOT partitions"
  df /tmp $AIPSROOT

# Define major AIPS++ directories.
  AIPSLAVE="$AIPSROOT/slave"
  AIPSCODE="$AIPSROOT/code"


# Get the user name.
  USERNAME=$LOGNAME@

  echo "modules=$MODULES  packages=$PACKAGES "

  if [ $INHALE -gt 0 ] ; then
    echo ""
    echo "testsuite: inhale started"
    echo ""
    inhale $INHALE_FLAG > $AIPSROOT/inhale.log 2>&1
    echo ""
    echo "testsuite: runtests started"
    echo ""
    runtests
  else
     if [ $HASPACKAGES -gt 0 -o $HASMODULES -gt 0 ] ; then
       if [ $HASPACKAGES -eq 0 ] ; then
```

```
        PACKAGES="aips"
      fi
      for PACK in $PACKAGES
      do
        if [ $HASMODULES -gt 0 ] ; then
           for MODULE in $MODULES
           do
              if [ -d $AIPSCODE/$PACK/implement/$MODULE ] ; then
                 cd $AIPSCODE/$PACK/implement/$MODULE
                 gmake runtests
              else
                 echo "testsuite: package $PACK has no module name $MODULE."
              fi
           done
        else
          if [ -d $AIPSCODE/$PACK ] ; then
              cd $AIPSCODE/$PACK
              gmake runtests
          else
              echo "testsuite: no package named $PACK"
          fi
        fi
      done
    else
       runtests
    fi
fi

echo ""
echo "testsuite: Execution terminated at"
echo "testsuite: `adate`"
echo "testsuite: `adate -l`"
exit 0
```

G runtest

```
#!/bin/sh
```

```
#-----------------------------------------------------------------------------
# Usage: runtests
#-----------------------------------------------------------------------------
# runtests runs the AIPS++ test programs and mails the results to the AIPS++
# email exploders.
#
# Options:
#   none
#
# Status returns:
#    0:  success
#    1:  initialization error
#
# Notes:
#    1) Failures are only counted for the "aips" package.
#
# Original: 1997/01/29 by Mark Calabretta, ATNF
# $Id: assay.latex,v 19.0 2003/07/16 03:50:40 aips2adm Exp $
#-----------------------------------------------------------------------------
# Fundamentals.
  if [ "$AIPSPATH" = "" ] ; then
     echo "runtests: AIPSPATH is undefined, abort!" 1>&2
     exit 1
  fi

  AIPSROOT=`echo $AIPSPATH | awk '{ print $1 }'`
  AIPSARCH=`echo $AIPSPATH | awk '{printf("%s/%s",$1,$2)}'`
  BINTESTD=`echo $AIPSPATH | awk '{printf("%s/%s/bintest",$1,$2)}'`

# Get the current version number.
  VERSION=`avers | awk '{printf("%.6s",$1)}'`

# HP-UX needs to use mailx (for the -s switch). Other systems use mail.
  if [ `uname` = "HP-UX" ] ; then
    MAILCMD=mailx
  else
    MAILCMD=mail
  fi

# Do it, sending the results to aips2-inhale.
```

```
  {  exec 2>&1

     echo ""
     echo "runtests: $AIPSPATH"
     echo ""
     echo "runtests: Validating $VERSION."

#    Initiate the test suite, preserving the log.
     cd $AIPSROOT
     gmake -C code TESTOPT=opt NODEP=1 runtests > $BINTESTD/runtests.log 2>&1

#    Check for failures in the aips package only (nobody relies on trial).
     cd $BINTESTD
     FAILED=`grep '^aips-.*FAIL (' runtests.report | wc -l`
     FAILED=`echo $FAILED`
     echo ""
     echo "runtests: Version $VERSION finished with $FAILED failures."
     echo ""
     echo "runtests: runtests summary and complete log are appended."
     echo ""
     cat $BINTESTD/runtests.report
     echo ""
     cat $BINTESTD/runtests.log
     echo $PATH
     testsum.sh $AIPSARCH aips > $BINTESTD/runtests_aips.summary 2>&1
     testsum.sh $AIPSARCH trial > $BINTESTD/runtests_trial.summary 2>&1
  } > $BINTESTD/runtests.rptNlog

  exit 0
```

H testsum.sh

```
#!/bin/sh
#-----------------------------------------------------------------------------
# testsum.sh: Formats the runtests.report file
#-----------------------------------------------------------------------------
#
#   Copyright (C) 1992-1997,1998,1999,2000,2001
```

```
#   Associated Universities, Inc. Washington DC, USA.
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   Correspondence concerning AIPS++ should be addressed as follows:
#          Internet email: aips2-request@nrao.edu.
#          Postal address: AIPS++ Project Office
#                          National Radio Astronomy Observatory
#                          520 Edgemont Road
#                          Charlottesville, VA 22903-2475 USA
#
#-----------------------------------------------------------------------------
# Usage: testsum.sh $AIPSARCH package
#-----------------------------------------------------------------------------
# testsum.sh -- Makes a nicely formated listing out of the runtests.report
#              file found in $AIPSARCH/bintest
#
# Options: None
#
# Notes:
#    Needs gawk or nawk to work
#=============================================================================
AWK=awk
if [ -x /usr/bin/nawk ] ; then
   AWK=nawk
 elif [ -x /usr/local/bin/gawk ] ; then
   AWK=gawk
 elif [ -x /opt/local/gnu/bin/gawk ] ; then
```

```
   AWK=gawk
 elif [ -x /usr/bin/gawk ] ; then
   AWK=gawk
fi
PACK=$2
VERSION=`avers | awk '{printf("%.6s",$1)}'`
TPASS=`grep "^$2.*PASS" $1/bintest/runtests.report | wc -l`
TUNTESTED=`grep "^$2.*UNTESTED" $1/bintest/runtests.report | wc -l`
TFAIL=`grep "^$2.*FAIL" $1/bintest/runtests.report | wc -l`
echo
echo "Summary of $AIPSPATH runtests $VERSION"
echo
echo "******************************************************************************"
echo
echo $PACK | awk '{printf "Test results for %s package\n", $1}'
echo $TPASS $TFAIL $TUNTESTED| awk '{printf "\t%5.1f%% Passed %d of %d (%d skipped)\n", 100*$1/($1+$2), $1, $1+$2, $3}'
echo "******************************************************************************"
echo
echo "Tests that failed to compile"
echo
grep "^$2.*FAIL.*compile" $1/bintest/runtests.report
echo
echo "Tests that failed to execute"
echo
grep "^$2.*FAIL.*execute" $1/bintest/runtests.report
echo
echo "Tests that failed to verify"
echo
grep "^$2.*FAIL.*verify" $1/bintest/runtests.report
grep "^$2.*FAIL.*execute" $1/bintest/runtests.report > /tmp/aips2tests.noexecute
echo
echo "Tests that were skipped"
echo
grep "^$2.*UNTESTED" $1/bintest/runtests.report
if [ -s /tmp/aips2tests.noexecute ]
then
   echo
   echo "******************************************************************"
   echo "Details tests that had execution failures"
   echo "******************************************************************"
```

```
   echo

   NOVERIFY=`awk '{print $2}' /tmp/aips2tests.noexecute`
   for testrpt in $NOVERIFY
   do
      echo
      echo "*****************************************************************"
      echo "$testrpt -- report"
      echo "*****************************************************************"
      echo
      $AWK -v testrpt=$testrpt '{printf "%s-execute> %s\n", testrpt, $0}' $1/bintest/$testrpt.report
   done
fi
rm /tmp/aips2tests.noexecute
echo
grep "^$2.*FAIL.*verify" $1/bintest/runtests.report > /tmp/aips2tests.noverify
if [ -s /tmp/aips2tests.noverify ]
then
   echo
   echo "*****************************************************************"
   echo "Details tests that had verify failures"
   echo "*****************************************************************"
   echo

   NOVERIFY=`awk '{print $2}' /tmp/aips2tests.noverify`
   for testrpt in $NOVERIFY
   do
      echo
      echo "*****************************************************************"
      echo "$testrpt -- report"
      echo "*****************************************************************"
      echo
      $AWK -v testrpt=$testrpt '{printf "%s-verify> %s\n", testrpt, $0}' $1/bintest/$testrpt.report
   done
fi
rm /tmp/aips2tests.noverify
echo
echo "Tests that passed"
echo
grep "^$2.*PASS" $1/bintest/runtests.report
```

```
echo
echo "*****************************************************************"
echo "End tests report for package $PACK"
echo "*****************************************************************"
```

I Example Daily Checklist

```
Daily checklist testing:

Documentation spot-check: OK
htdig spot-check : OK
Sneeze logs:

argus_sgi: OK
duw01_sun4sol: OK
muse_linux: OK
kaputar_alpha: No sneeze

Assay:

linux: --
        machine name, location, type:  muse, CV
Time to initialize AIPS++ = 82
Total memory used in Glish variables = 2651068 bytes

        assay results:

Problems with componentlist_meta.g:  -- Notify tcornwel

"componentlist_meta.g", line 73: warning, .scalarmeasure is not a field in [exis
"componentlist_meta.g", line 73: error, F is not a function value
"componentlist_meta.g", line 73: warning, operand to .vector_double is not a rec
ord
"componentlist_meta.g", line 200: warning, .scalarmeasure is not a field in [exi
"componentlist_meta.g", line 200: error, F is not a function value

Problems with imageservertest: -- Notify nkilleen
```

```
Test 18 - hanning
Creating image 'imagetest_temp/shapetoimage.image' of shape [10, 20]
Given region object is invalid
Starting image::hanning
Specified axis of 20is greater than input image dimension of 2
Caught an exception! Event type=run exception=Mon Nov 22 11:07:46 1999    SEVERE
 image::hanning(...) (file /aips++/daily/code/trial/implement/Images/DOimage2.cc
, line 1707) ObjectID=[sequence=871 host=charybdis pid=5787 time=943268865]:
Specified axis of 20is greater than input image dimension of 2
Method hanning fails!
Parameter drop cannot convert between numbers and strings
Method hanning fails!
...

Test 27 - GUIS: momentsgui, sepconvolvegui, maskhandlergui, view
Converting pixels array to float type
Caught an exception! Event type=create exception=Invalid Table operation: SetupN
ewTable imagetest_temp/arraytoimage1.image is already opened (is in the table ca
che)
arraytoimage constructor 1 failed
"assay.g", line 69: warning, .message is not a field in [file=imageservertest.g,
 line=4519, stack=. . . eval imagetest imageservertest]
imagetest fails:  F
"assay.g", line 70: warning, .message is not a field in [file=imageservertest.g,
 line=4519, stack=. . . eval imagetest imageservertest]
"assay.g", line 81: error, types are incompatible

assay() still returns 0 fails despite problems

Unit Testing:

Test/Trial results summary

linux_egcs  test  results -- socorro sneffels

OK

sun4sol_egcs test results -- socorro tarzan

OK
```

```
sun4sol_egcs trial results -- socorro tarzan

Problems: Notify aips2-modulebosses, tcornwel, nkilleen

Summary of /aips++/daily sun4sol_egcs socorro tarzan runtests 13.172

******************************************************************************

Test results for trial package
         77.6% Passed 121 of 156 (2 skipped)
******************************************************************************

Tests that failed to compile

trial-Coordinates-test  tTabularCoordina  13.172  1999/11/19 14:31:00  FAIL (compile)
trial-Display-test      dPSLatticeAsRast  13.172  1999/11/19 14:31:18  FAIL (compile)
trial-DisplayDatas-test dLatticeAsRaster  13.172  1999/11/19 14:32:32  FAIL (compile)
trial-DisplayDatas-test tArrayAsRaster    13.172  1999/11/19 14:32:43  FAIL (compile)
trial-Fitting-test      tLinearFitSVD     13.172  1999/11/19 14:35:18  FAIL (compile)
trial-Graphics-test     tArrayBrowser     13.172  1999/11/19 14:35:59  FAIL (compile)
trial-IO-test           tTapeIO           13.172  1999/11/19 14:36:03  FAIL (compile)
trial-Lattices-test     dLatticeStatisti  13.172  1999/11/19 14:43:59  FAIL (compile)
trial-MeasurementCompon KnownAlgorithms   13.172  1999/11/19 15:06:59  FAIL (compile)
trial-MeasurementCompon tAlgoPClark       13.172  1999/11/19 15:07:10  FAIL (compile)
trial-MeasurementCompon tBeamSquint       13.172  1999/11/19 15:07:14  FAIL (compile)
trial-MeasurementEquati dCEMemModel       13.172  1999/11/19 15:08:47  FAIL (compile)
trial-MeasurementEquati dClarkCleanModel  13.172  1999/11/19 15:08:51  FAIL (compile)
trial-MeasurementEquati tCCList           13.172  1999/11/19 15:08:54  FAIL (compile)
trial-MeasurementEquati tCEMemModel       13.172  1999/11/19 15:09:00  FAIL (compile)
trial-MeasurementEquati tClarkCleanLatMo  13.172  1999/11/19 15:09:05  FAIL (compile)
trial-MeasurementEquati tClarkCleanModel  13.172  1999/11/19 15:09:11  FAIL (compile)
trial-MeasurementEquati tConvEqnMask      13.172  1999/11/19 15:09:15  FAIL (compile)
trial-MeasurementEquati tConvEqnStokes    13.172  1999/11/19 15:09:20  FAIL (compile)
trial-MeasurementEquati tConvolutionEqua  13.172  1999/11/19 15:09:24  FAIL (compile)
trial-MeasurementEquati tHogbomCleanMask  13.172  1999/11/19 15:09:28  FAIL (compile)
trial-MeasurementEquati tHogbomCleanMode  13.172  1999/11/19 15:09:32  FAIL (compile)
trial-MeasurementEquati tHogbomCleanStok  13.172  1999/11/19 15:09:37  FAIL (compile)
trial-MeasurementEquati tLatConvEquation  13.172  1999/11/19 15:09:44  FAIL (compile)
trial-MeasurementEquati tStokesUtil       13.172  1999/11/19 15:09:51  FAIL (compile)
```

```
trial-MeasurementEquati tVisibilityItera  13.172  1999/11/19 15:10:01  FAIL (compile)
trial-MeasurementSets-t tMSMainBuffer     13.172  1999/11/19 15:10:17  FAIL (compile)
trial-MeasurementSets-t tMSRange          13.172  1999/11/19 15:10:22  FAIL (compile)
trial-MeasurementSets-t tMSSelector       13.172  1999/11/19 15:10:35  FAIL (compile)
trial-MeasurementSets-t tMSSimulator      13.172  1999/11/19 15:10:45  FAIL (compile)
trial-Tasking-test      testServer        13.172  1999/11/19 15:13:15  FAIL (compile)

Tests that failed to execute

trial-Display-test      dRGBWCRaster      13.172  1999/11/19 14:31:41  FAIL (execute)
trial-Lattices-test     tLatticeConvolve  13.172  1999/11/19 14:56:27  FAIL (execute)

Tests that failed to verify

trial-Coordinates-test  dCoordinates      13.172  1999/11/19 14:28:11  FAIL (verify)
trial-Lattices-test     tLEL              13.172  1999/11/19 14:54:07  FAIL (verify)

Tests that were skipped

trial-Fitting-test      tLinearFitConstr  13.172  1999/11/19 14:35:13     UNTESTED
trial-Images-test       dPagedImage       13.172  1999/11/19 14:38:12     UNTESTED

*****************************************************************
Details tests that had execution failures
*****************************************************************


*****************************************************************
dRGBWCRaster -- report
*****************************************************************

dRGBWCRaster-execute> Error: Can't open display:
dRGBWCRaster-execute> FAIL (execution failure): dRGBWCRaster

*****************************************************************
tLatticeConvolve -- report
*****************************************************************

nawk: can't open file /aips++/daily/sun4sol_egcs/bintest/tLatticeConvolve.report
 source line number 1
```

```
*****************************************************************
Details tests that had verify failures
*****************************************************************


*****************************************************************
dCoordinates -- report
*****************************************************************

dCoordinates-verify> 4c4
dCoordinates-verify> < Expected error: Stokes value XX is not found on any axis position
dCoordinates-verify> ---
dCoordinates-verify> > Expected error: Stokes value 9 is not found on any axis position
dCoordinates-verify> FAIL (output not verified): dCoordinates

*****************************************************************
tLEL -- report
*****************************************************************

tLEL-verify> 382c382
tLEL-verify> <       Caught expected exception; message is: LELFunctionBool::eval - unknown function
tLEL-verify> ---
tLEL-verify> >       Caught expected exception; message is: LELFunctionBool::eval - cannot be used
tLEL-verify> 384c384
tLEL-verify> <       Caught expected exception; message is: LELFunctionBool::eval - unknown function
tLEL-verify> ---
tLEL-verify> >       Caught expected exception; message is: LELFunctionBool::eval - cannot be used
tLEL-verify> FAIL (output not verified): tLEL
```

J Example Release checklist

```
Release testing log for RedHat 6.0
----------------------------------

a) Use pristine machines
------------------------
```

Used:
bonzo.aoc.nrao.edu, recently upgraded to RH6.0 (somewhat pristine).
voyager.aoc.nrao.edu, used for build (not pristine at all).

b) Run from CDROM (including GUI tests, eg. toolmanager)

Problems:

```
1) On bonzo, reports:
    "<fail>: tab tk_load event: couldn't load file ../gPgplot.so.5
    undefined symbol getenv_
    later: : "glish.eval" warning: uninitialized global var.
             pgplot used.
    pgplotter() is not functional.
    Note: bonzo has /usr/lib/libpgplot.so.5 from rpm pgplot-5.2.0-3u,
    "ldd imager" indicates it is linked against /usr/lib/libpgplot.so.5.
    Not seen on voyager, which has pgplot-5.2.0-8u.

2) Global function windows in the toolmanager do not dismiss or disappear
   on termination.

3) tablebrowser reported tableserver.g (locking) errors on startup
   from the catalog, when examining 3C273XC1.ms. Not fully functional
   after that. Not seen on voyager - could be a simple locking error
   in previous application that was run.

4) assay().trytests() completes with zero, but there are imageserver
   errors.
   In /home/maser2/aips++/system/release-1.2.3/assay.cdrom.{voyager|bonzo}.
```

c) Run from disk installation

Problems:

```
1) $a_root is not unset. Startup aipsinit.csh retain CDROM $AIPSPATH
   in error.

2) Same pgplot errors as when running from the CDROM directly.
```

```
3) assay().trytests() completes with zero, but there are imageserver
   problems.
   In /home/maser2/aips++/system/release-1.2.3/assay.disk.{voyager|bonzo}.

d) Assay()
---------

See (a) and (b).

e) htdig check
--------------

Not done.

f) Standardized walk-through of documentation links
---------------------------------------------------

Problems:

1) recipes links yield directory listings.
2) programmer documentation in aips has wrong links (not found);
   eg. Array or Vector.
3) "Return to AIPS++ Home page" link at the bottom of each
   User Ref. Man page returns the user to a directory listing.

g) Release notes
----------------

No problems found with readme.html.

h) CDROM label and cover information
------------------------------------

Not included with this CDROM.

i) Verify advertised patches
----------------------------

Checked: calibrater_meta.g       ok
```

```
        cookbook                ok
        calibrater::fluscale()  ok

--------------------------------------------------------------------------
```

K Module Bosses

```
----------------------------------------------------------------------
AIPS PACKAGE (Jun 99)
----------------------------------------------------------------------
LIBRARY:
Arrays:                 Ger             gvd@nfra.nl
Containers:             Darrell         dschieb@nrao.edu
Deconvolution:          Ralph           rmarson@nrao.edu
Exceptions:             Darrell
FITS:                   Bob             bgarwood@nrao.edu
Functionals:            Ralph
Glish:                  Darrell
Graphics:               David           dbarnes@atnf.csiro.au
Guiutils:               Bob
IO:                     Ger
Images:                 Neil
Inputs:                 Ger
Lattices:               Ger
Logging:                Ger
Mathematics:            Ralph
MeasurementSets:        MarkW           mwiering@atnf.csiro.au
Measures:               Wim             Wim.Brouw@csiro.au
OS:                     Ger
Quanta:                 Wim
RTTI:                   Darrell
Tables:                 Ger
Tasking:                Darrell
Utilities:              Ger

APPS:
gtable                  Ger

----------------------------------------------------------------------
```

```
TRIAL PACKAGE (Jun 99)

Module bosses are NOT allocated for the trial package. Instead the
authors of the code are responsible for the code. If the author cannot
support the code eg., because they have left the project, then the
code is marked as unsupported. Unsupported code may be deleted if it
is not widely used. Otherwise a foster parent will have to be found.
----------------------------------------------------------------------
LIBRARY:
Arrays:
  MatrixSolver          Tim Cornwell
  NNLSMatrixSolver      Tim Cornwell
  RigidVector           Mark Wieringa
  SquareMatrix          Mark Wieringa
ComponentModels:
  All classes           Ralph Marson
Containers:
  HashMap*              Darrell Schiebel
  RecordField*Writer    Bob Garwood
Coordinates:
  All classes           Neil Killeen
Datalinetool:
  All classes           Ger van Diepen
Display:
  All classes           David Barnes
DisplayCanvas:
  All classes           David Barnes
DisplayDatas:
  All classes           David Barnes
DisplayEvents:
  All classes           David Barnes
FITS
  All classes           David Barnes
Fitting:
  All classes           Wim Brouw
Functionals:
  Algorithm             Wes Young
  Applicator            Wes Young
  All other classes     Ralph Marson
Graphics:
```

```
  All classes           David Barnes
Guiutils:
  aipsbrowser           Wes Young
  bug                   Wes Young
  choice                David Barnes
  displaylist*          Joe McMullin
  displaytext           Ger van Diepen
  filterlistbox         Jow McMullin
  fontchooser           Wes Young
  guicomponents         Wes Young
  guiframework          Wes Young
  guiutils              Wes Young
  inputbox              Tim Cornwell
  inputframe            Wes Young
  oldcombobox           Bob Garwood
  pgplotter*            Jow McMullin
  popuphelp             Wim Brouw
  recordbrowser         Wes Young

IO
  TapeIO                Ralph Marson

Images:
  All classes           Ger van Diepen

Lattices:
  All classes           Ger van Diepen

Mathematics:
  AutoDiff*             Ralph Marson
  ConvolveGridder       Tim Cornwell
  DO*                   Ralph Marson
  Gridder               Tim Cornwell
  HistAcc               Ralph Marson
  Interpolate1DArray    Mark Wieringa
  NNGridder             Tim Cornwell
  SCSL                  Doug Roberts
  StatAcc               Ralph Marson
  VectorKernel          Neil Killeen
  VectorPool            Ralph Marson
```

```
  fftserver            Ralph Marson
  gaussian             Ralph Marson
  interpolat_1d        Ralph Marson
  lsfit                Jan Noordam
  matrix               Jan Noordam
  polyfitter           Ralph Marson
  randomnumbers        Ralph Marson
  ranges               Bob Garwood
  sinusoidfitter       Ralph Marson
  statistics           Ralph Marson

MeasurementComponents:
  All classes          Tim Cornwell

MeasurementEquations:
  All classes          Tim Cornwell

MeasurementSets:
  MS*Buffer* classes   Ralph Marson
  MSSummary            Peter Barnes
  MSLister             Peter Barnes
  All other classes    Mark Wieringa

Measures:
  All classes          Wim Brouw

OS:
  Modcomp*             Ralph Marson

Parallel:
  All classes          Wes Young

Quanta:
  All classes          Wim Brouw

SDCalibration:
  All classes          Bob Garwood

SDIterators:
  All classes          Bob Garwood
```

```
Simulators:
  All classes           Tim Cornwell

TableMeasures:
  All classes           Ger van Diepen

Tasking:
  All classes           Darrell Schiebel

Utilities:
  aips++                Tim Cornwell
  aips++init.g          Tim Cornwell
  assay                 Tim Cornwell
  defaultattributes     Ralph Marson
  demonstration         Tim Cornwell
  edit                  Tim Cornwell
  finclude              Tim Cornwell
  gmisc                 Bob Garwood
  itemcontainer         Neil Killeen
  note                  Bob Garwood
  plugins               Tim Cornwell
  serverexists          Neil Killeen
  sh                    Tim Cornwell
  stopwatch             Tim Cornwell
  substitute            Ger van Diepen

Widgets:
  All classes           David Barnes
APPS
  aips2help             Wes Young
  aipstask              Athol Kemball
  aipsview              Harold Ravlin
  apputil               Empty
  aserver4mpi           Wes Young
  calibrator            Athol Kemball
  catalog               Tim Cornwell
  componentlist         Ralph Marson
  datalinetool          Jan Noordam
  deconvolver           Mark Holdaway
```

dish	Bob Garwood
dishplot	Joe McMullin
duplicates	Wim Brouw
fits2ms	Bob Garwood
fits2table	Bob Garwood
gDisplay	David Barnes
gcatalog	Tim Cornwell
gfitgauss	Bob Garwood
gplot1d	Joe McMullin
gtablebrowser	Wes Young
guimisc	Wes Young
image	Neil Killeen
imager	Tim Cornwell
measures	Wim Brouw
misc	Wes Young
ms	Mark Wieringa
ms2fits	Bob Garwood
ms2sdfits	Bob Garwood
numerics	Ralph Marson
pimager	Wes Young
plotter	Joe McMullin
quanta	Wim Brouw
reident	Wim Brouw
sdfits2ms	Bob Garwood
sditerator	Bob Garwood
simpleimage	Tim Cornwell
simulator	Tim Cornwell
table	Ger van Diepen
unused	Wim Brouw
used	Wim Brouw
viewer	David Barnes
visplot	Tim Cornwell