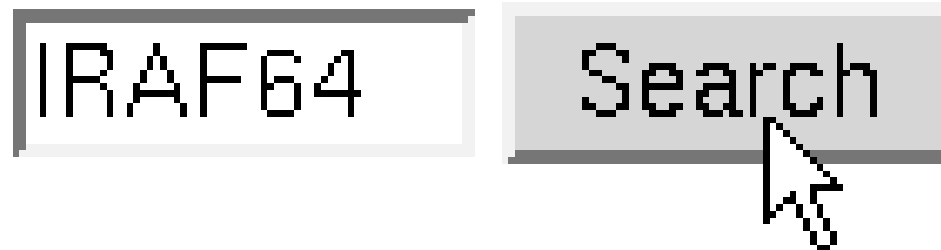


# Current status of 64-bit IRAF

Chisato Yamauchi

Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency  
(ISAS/JAXA)



# Overview

- Importance of 64-bit IRAF
- Difficulty of porting IRAF to 64-bit
- SPP data models
- Our project - The IRAF64 Project
- Our results
- Screenshots
- Our Plan

# Importance of 64-bit IRAF

- ➔ IRAF is the de fact standard software for analysis of optical/infrared data.
- ➔ Modern large telescopes and satellites still utilize the IRAF.
- ➔ IRAF will keep being used in the future.
- ➔ Current 32-bit IRAF has the 4GB barrier, which should be removed to handle incredibly high-resolved/wide-field data in the future.  
(e.g. SUBARU Hyper Suprime-Cam will output a 2.8GB of FITS)

next: Difficulty of porting IRAF to 64-bit

# Difficulty of porting IRAF to 64-bit

- ➔ IRAF requires a small assembler code for each architecture.

```
zsvjump_:
    # %rsi ... &status   %rdi ... &jmpbuf
    movq   %rsi, (%rdi)   # store &status in jmpbuf[0]
    movl   $0, (%rsi)     # zero the value of status
    addq   $8, %rdi       # change point to &jmpbuf[1]
    movl   $0, %esi       # change arg2 to zero
    jmp    __sigsetjmp    # let sigsetjmp do the rest
```

This is zsvjump.s for x86\_64 OS. The arguments of a function are received by registers (not by stack).

# Difficulty of porting IRAF to 64-bit

- ➔ Intrinsic problem is the tacit assumption of size of integer and pointer types in IRAF code.

i.e., `sizeof(int) == sizeof(long) ==  
sizeof(pointer) == sizeof(real)`.

## Examples of actual problems:

- Incompatible pointer args of functions.
- There is 'Memi' (for memory access for integer type). However, 'Memp' (that for pointer) does not exist.
- P2R() and P2I() macros do not exist .
- There are tricks around 'struct' constructions in some codes.

next: Data model of SPP

# Data model of SPP

- ➔ Which data model is suitable for 64-bit SPP?

	short	int	long	pointer	
ILP32	16	32	32	32	(32-bit gcc)
LP64	16	32	64	64	(64-bit gcc)
ILP64	16	64	64	64	

- If we select ILP64, a number of tools will work without code modification. However, IRAF contains many codes dependent on 32-bit integer.
- Genuine approach is to renounce the tacit assumption of integer/pointer sizes.

We select LP64 model and revise SPP specifications, and remove the root problem.

next: [Our project](#)

# Our project - The IRAF64 Project

- ➔ Porting the IRAF software to the x86\_64 Linux OS with NATIVE x86\_64 (Opteron, EM64T) executable.
- ➔ This IRAF is developed as a candidate of IRAF Revision 3.0.
- ➔ Our project uses the SourceForge.Net (<https://sourceforge.net/>).
- ➔ The source code is maintained using the Subversion, and the repository is public now.
- ➔ We review and update all C/SPP codes.

# Our project - The IRAF64 Project

## ➔ Project members

### - Project Admins/Developer:

Chisato Yamauchi (Japan Aerospace Exploration Agency)

### - Developer/Tester:

Keith Rosema (University of Washington)

Sergio Pascual (Complutense University of Madrid)

Yasushi Nakajima (National Astronomical Observatory of Japan)

next: Our results



# Our results: general improvements

- ➔ Improvements of C code to meet the present standard.
  - Enabled gcc's '-Wall' option and minimized gcc's warnings.
  - Replaced ancient style (K&R -> ANSI).
  - Appended complete prototype declarations.
  - Improved a lot of security issues.
  - gcc-4.3 is also OK!
- ➔ Rearranged IRAF directory structure.
- ➔ Redefined 'MACH' environment variable.  
e.g., MACH=x86\_64-linux-redhat, MACH=i386-linux-generic, etc.

# Our results: general improvements

- ➔ Appended an mechanism of auto-detection of OS and architecture when building IRAF.
- ➔ Applied 'GNU make' to build IRAF easily.

```
$ make boot
```

```
$ make iraf
```

```
$ make reboot
```

```
$ make tables
```

```
$ make noao
```

```
$ make install (DESTDIR=...)
```

OS and architecture are automatically detected on 'make boot' stage.

- ➔ Replaced csh-scripts with sh-scripts.

# Our results: updates for x86\_64

- ➔ Wrote an assembler code for Linux x86\_64 architecture.
- ➔ Prepared the environment for 'LP64-SPP' development.
  - Some modifications for SPP preprocessor/convertor and f77 configurations.
  - xc, rpp, xpp and mkpkg are OK!
- ➔ Reviewed and updated C and SPP codes for libsys.a, libvops.a, libcur.a, libstg.a, libds.a, libex.a and libc.a.
  - Supported 64-bit memory allocation and file I/O.
  - Supported large data (> 4GB) in the majority of functions.

# Our results: updates for x86\_64

- ➔ 'cl' is running on CentOS 4.x/Fedora 8,9.
  - A few tasks of core system are OK.
  - However, the majority of tasks cause PANIC error.

next: Screenshots

# Screenshots (cl)

```
xterm - izumo.plain.isas.jaxa.jp:[~/iraf_test]
=====
A_A  IRAF64 pre-alpha based on NOAO/IRAFNET PC-IRAF Revision 2.14
('v') This is NOT the released version of IRAF64.
=====

This 64-bit version of IRAF is ported by the IRAF64 project hosted by
Chisato Yamauchi (ISAS/JAXA).
Visit http://www.ir.isas.jaxa.jp/~cyamauch/iraf64/ to report problems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type `help <command>'. To run a
command or load a package, type its name. Type `bye' to exit a
package, or `logout' to get out of the CL. Type `news' to find out
what is new in the version of the system you are using.

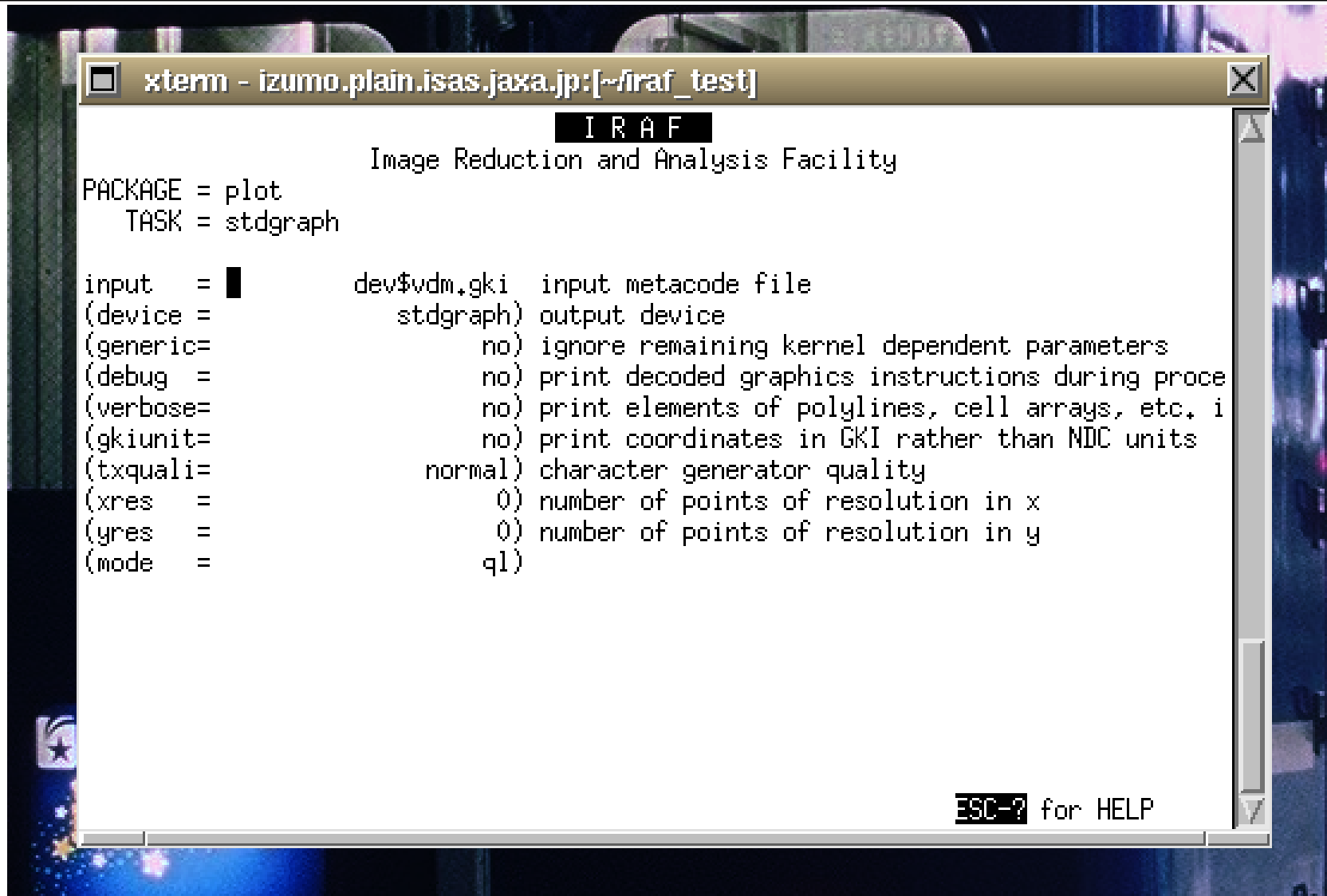
The following commands or packages are currently defined:

    dataio.    language.  obsolete.   softtools.  utilities.
    dbms.      lists.      plot.       system.
    images.   noao.      proto.      tables.

cl> !ldd /opt/local/iraf/iraf/bin/cl.e
    libreadline.so.4 => /usr/lib64/libreadline.so.4 (0x0000003049c00000)
    libncurses.so.5 => /usr/lib64/libncurses.so.5 (0x0000003049a00000)
    libm.so.6 => /lib64/tls/libm.so.6 (0x0000003049200000)
    libc.so.6 => /lib64/tls/libc.so.6 (0x0000003048f00000)
    /lib64/ld-linux-x86-64.so.2 (0x0000003048d00000)

cl> █
```

# Screenshots (epar)



The screenshot shows an xterm window titled "xterm - izumo.plain.isas.jaxa.jp:[~/iraf\_test]". The window content displays the IRAF logo and the text "Image Reduction and Analysis Facility". Below this, it shows the current package and task: "PACKAGE = plot" and "TASK = stdgraph". A list of command-line options follows, including "input", "(device)", "(generic)", "(debug)", "(verbose)", "(gkiunit)", "(txqual)", "(xres)", "(yres)", and "(mode)". Each option is followed by its default value and a brief description. At the bottom right of the window, it says "ESC-? for HELP".

```
xterm - izumo.plain.isas.jaxa.jp:[~/iraf_test]
      I R A F
      Image Reduction and Analysis Facility

PACKAGE = plot
      TASK = stdgraph

input  = dev$vdm.gki  input metacode file
(device = stdgraph) output device
(generic= no) ignore remaining kernel dependent parameters
(debug = no) print decoded graphics instructions during proce
(verbose= no) print elements of polylines, cell arrays, etc. i
(gkiunit= no) print coordinates in GKI rather than NDC units
(txqual= normal) character generator quality
(xres = 0) number of points of resolution in x
(yres = 0) number of points of resolution in y
(mode = ql)

ESC-? for HELP
```

# Screenshots (showcap)

```
xterm - izumo.plain.isas.jaxa.jp:[~/sraf_test]
cl> showcap
cmd : `set' device
    | `*' (to dump full graphcap entry
    | cc [arg1 [arg2 [arg3]])
    ;

cc :  a two character capcode (e.g., 'cm')
    |  an encoder program (non alpha first char)
    ;

* set vt640
:RC=(1$2)^X\E[24;65H\E[7mLIGHT PEN READY\E[Om($$)^]\E"(1$2)5($D)4($$)g:\
WC=^]%t\E/f:Ow=150^]^\_ :CW=^X\E[24;0H\E[K:GE=150^]^\_ :GD=^X\E[24;0H\E[K:\
lt#5;nc#2;se;CL=50^]\E^L:xr#640;yr#480;ar#,.57;xs#,.23;ys#,.13;ar#,.70;\
ch#,.0294;co#80;cw#,.0125;in;k1#1;k2#127;kf=cl;li#35;lt#5;nc#1;\
nk#127;pl;pm;th#4;t1#1;t2#2;t3#3;t4#4;tx;wc;xr#1024;yr#780;xs#,.20;\
ys#,.14;CD=^M:CN#6;LT=^]\E/(1$0)1d\E`($1-5)0d\E(1_+.$D)0d\E`($$:\
MS=\034;PL;RC=\E^Z:SC=(,!3, & *, &+!1, & *, &+!2;TH=\E(1#47+.\_:\
TS=^]%t^\_ :VS=^];DE=^];X1#0;X2#1023;XY=%t;Y1#0;Y2#779;Ow=^]^\_ :\
CW=(#682!2#0!1)^]%t^\_ :GE=^]^\_ :CL=1000(#32!9)\E^L;LR=(#32!9;GD=(9#1-!99$0#31!9$9
#22*!2#0!1)^]%t^\_ :\

* TH 2
program: ^[(1#47+.\_
encoding: ^[1
        status = 0
* █
```

# Screenshots (stdgraph)

The screenshot displays the IRAF software interface. On the left, a terminal window titled 'xterm - izumo.plain.isas.jaxa.jp:~/iraf\_t' shows the following text:

```
=====
A__A IRAF64 pre-alpha based on NOAO/IRAF
('v') This is NOT the released version
=====

This 64-bit version of IRAF is ported by
Chisato Yamauchi (ISAS/JAXA).
Visit http://www.ir.isas.jaxa.jp/~cyamauchi/

Welcome to IRAF. To list the available commands, type 'help'.
For detailed information about a command, type 'help command'.
To load a package, type 'load package', or 'load package, version'.
To unload a package, type 'unload package'.
To see what is new in the version of the system you are using, type 'news'.

The following commands or packages are currently installed:

  dataio.    language.  obsolete.  software.
  dbms.     lists.    plot.      su.
  images.   noao.    proto.     tables.

cl> stdgraph dev$vdm.gki
cl> 
```

On the right, two plots are shown. The top plot, titled 'irafterm', displays a graph with 'Y - AXIS' on the vertical axis (ranging from -0.2 to 1.0) and 'X - AXIS' on the horizontal axis (ranging from -15 to 15). It features a prominent central peak at x=0 and several smaller peaks on either side, with a complex, multi-lobed structure on the right side.

The bottom plot, titled 'The SINC Function', displays a graph with 'Y - AXIS' on the vertical axis (ranging from -0.2 to 1.0) and 'X - AXIS' on the horizontal axis (ranging from -30 to 30). It shows a central peak at x=0 and a series of smaller, symmetric peaks on either side, characteristic of a sinc function.

Below the plots, a plot titled 'Line 250 of dev\$pix[200:300,\*]' is shown. The vertical axis ranges from 0 to 500, and the horizontal axis ranges from 0 to 100. The plot displays a series of peaks, with the most prominent ones at approximately x=10, x=35, x=65, and x=85. The plot is labeled 'ngc 4147 b 1800'.



# Screenshots (display)

```
xterm - izumo.plain.isas.jaxa.jp:[~/iraf_test]
-----
A__A  IRAF64 pre-alpha based on NOAO/IRAFNET PC-IRAF Revision 2.14
('v') This is NOT the released version of IRAF64.
-----

This 64-bit version of IRAF is ported by the IRAF64 project hosted by
Chisato Yamauchi (ISAS/JAXA).
Visit http://www.ir.isas.jaxa.jp/~cyamauch/iraf64/ to report problems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type 'help <command>'. To run a
command or load a package, type its name. Type 'bye' to exit a
package, or 'logout' to get out of the CL. Type 'news' to find out
what is new in the version of the system you are using.

The following commands or packages are currently defined:

    dataio.    language.  obsolete.  softtools.  utilities.
    dbms.      lists.       plot.      system.
    images.    noao.       proto.    tables.

c1> !ds9%
c1> display dev$pix 1
z1=35, z2=346,0218
c1> !!dd /opt/local/iraf/iraf/bin/cl.e
libreadline.so.4 => /usr/lib64/libreadline.so.4 (0x0000003049c00000)
libncurses.so.5 => /usr/lib64/libncurses.so.5 (0x0000003049a00000)
libm.so.6 => /lib64/tls/libm.so.6 (0x0000003049200000)
libc.so.6 => /lib64/tls/libc.so.6 (0x0000003048f00000)
/lib64/ld-linux-x86-64.so.2 (0x0000003048d00000)
c1> !printenv SPP_DATA_MODEL
lp64
c1>
```

The screenshot shows the IRAF64 graphical user interface. At the top, there is a menu bar with options: File, Bin, Zoom, Scale, Color, Region, WCS, Analysis, and Help. Below the menu bar is a control panel with several input fields and buttons. A small thumbnail image of a galaxy is visible in the upper right corner of the control panel. The main window displays a large, detailed grayscale image of a spiral galaxy. At the bottom of the window, there is a horizontal axis with numerical labels: 20, 40, 60, 80, 100, 120, 140, 160, 180, and 200.

# Screenshots (FITS file)

```
xterm - izumo.plain.isas.jaxa.jp:[~/iraf_test]
-----
A_A  IRAF64 pre-alpha based on NOAO/IRAFNET PC-IRAF Revision 2.14
('v') This is NOT the released version of IRAF64.
-----

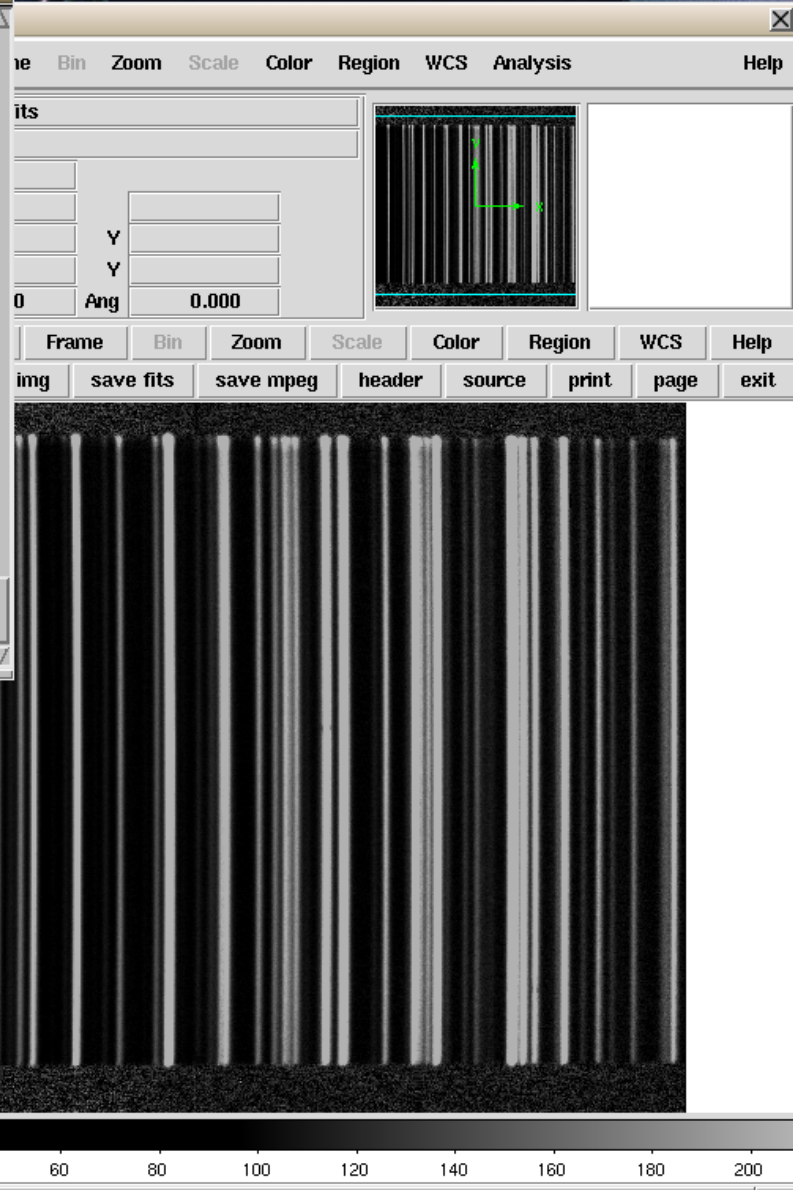
This 64-bit version of IRAF is ported by the IRAF64 project hosted by
Chisato Yamauchi (ISAS/JAXA).
Visit http://www.ir.isas.jaxa.jp/~cyamauch/iraf64/ to report problems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type 'help <command>'. To run a
command or load a package, type its name. Type 'bye' to exit a
package, or 'logout' to get out of the CL. Type 'news' to find out
what is new in the version of the system you are using.

The following commands or packages are currently defined:

    dataio,  language,  obsolete,  softtools,  utilities,
    dbms,    lists,     plot,      system,
    images,  noao,     proto,    tables.

cl> displ 1_comp.fits
frame to be written into (1:16) (1):
z1=-783,5954 z2=976,688
cl> ls
1_comp.fits clhistory.txt gf20030912_56_mos.fit login.cl uparm xg xt
cl> !uname -a
Linux izumo.plain.isas.jaxa.jp 2.6.9-67.0.15.ELsmp #1 SMP Thu May 8 10:50:20 EDT
2008 x86_64 x86_64 x86_64 GNU/Linux
cl> █
```



# Our plan

- ➔ First, we intend to provide 64-bit support of core system.  
(i.e., libraries and basic tasks in sys/ pkg/ and math/ directories).
- ➔ Tables library and NOAO standard utilities will be updated after support of core system.
- ➔ Collaboration with STScI/NOAO may be wanted for updates of Tables library and NOAO standard utilities.