IRAF AND BEYOND
Agenda

- Welcome
- IRAF Status and Plans – M. Fitzpatrick
- Compressed FITS Kernel – R. Seaman
- IRAF Pipelines – F. Valdes
- Future System Frameworks – D. Tody
- Discussion
IRAF Project Status

- IRAF v2.15 Alpha release, March 2010
  - Extended test period (intentional due to 64-bit changes)
  - Several interim updates
  - 64-bit Port Guide available
  - ~30 External Packages also ported

- Final v2.15 release in preparation

- Roadmap proposal for NOAO Director being developed

- Software development continues
  - Pipelines for ODI, MOSAIC/NEWFIRM
  - New FITS Kernel for tile compression
  - VO Integration
  - Ongoing maintenance

- X11IRAF Development on hold
iraf.net Status

- Entering 6\textsuperscript{th} Year of Operation
  - Almost 1600 registered \textit{(and real)} users
  - Avg \textgreater{}500 unique visits/day
  - Avg \textgreater{}1GB/day transfer (or \textasciitilde{}5-8 IRAF systems)
  - 73,618 visits from 145 countries (but lots of bots)
  - 2386 topics (> 1/day), 8533 posts (~4/day)

- Getting to be time for an update
IRAF v2.15 Highlights

- 64-bit platform support
- New MEMIO interface
  - Memory debugging features
- Architecture name changes
  - Single 'linux' arch for 32-bit systems
  - 'macintel' now 64-bit Intel, 'macosx' Universal 32-bit
  - Backwards compatible
- Automatic prototype checking, ANSI C code kernel/libs
- Cross-compilation support
  - Mac and Linux only
- Mixed architecture usage supported
- SVG Graphics Device for web presentation
- Simplified source builds
  - Support for toplevel make command
- Simplified single-user installs
- Now with TABLES goodness
64-bit IRAF Implications

- **For Users**
  - Just another supported platform
  - 10-15% speed increase over using 32-bit binaries
  - Large image/file/memory support

- **For Developers**
  - Some porting work *may* be required
    - **MUST** clean up prototype errors
    - **MUST** address use of TY_REAL in structs
    - **MAY** choose to provide 64-bit binaries
64-bit IRAF Implementation

- **We chose a systematic port over a proper one**
  - Minimize number of code changes required
  - Limit the types of code to be changed
  - Backwards interoperability a high priority
- **SPP Saves the Day**
  - Could use ILP64 model with LP64 GCC compilers
  - Automation hard, but tools can help

**Result**

Core system plus ~30 external packages ported in approx 3 man-months.
V2.15 External Packages

Not All Packages are the Same

- **Active/Supported**
  - MSCRED, FITSUTIL, NFEXTERN, etc

- **Script-Only Packages**
  - ESOWFI, CFH12K, etc

- **Deprecated**
  - ARED, EUV, IUE, etc

- **Externally Supported**
  - STSDAS/TABLES, GEMINI, RVSAO, etc

- **Obsolete**
  - DIGIPHOTX, NMISC, MFILTERS, etc
V2.15 Package Reorganization

- **Moved to Core System**
  - COLOR → PROTO
  - VOL → PROTO

- **TABLES included**
  - libtbtables part of core system
  - TTOOLS pkg now NTTOOLS in UTILITIES

- **Other Changes:**
  - FITSUTIL.NHEDIT → GMISC
  - NLOCAL.BASES → UTILITIES
IRAF v2.15 Release Plans

- Originally, before ADASS
- Now, just following ADASS
  - Last-minute inclusion of tbtables/ttools
- Linux and Mac only
  - Other platforms to follow in v2.15.1
- Simplified install a separate project
Other News

What about the old website?

- *iraf.noao.edu* to get a makeover
- Look-n-Feel of NOAO
- "*Official*" distributions of software
- Content shared w/ iraf.net (announcements, RSS feeds of forum etc).
- Integrated SVN and Trac
Roadmap Questions

- What should our 5-year strategy be?
  - Preserve the trusted science code
- Should catalog science be our next frontier?
  - Interfaces to DB and distributed data
- Is user-scriptability still an issue? What about compiled code?
- New Frameworks….
  - VAO study project, other deployments

<Your thoughts here>