AIPS++ DEVELOPMENT PLAN: Release 1.6

Athol Kemball and Tim Cornwell (eds.)
NRAO

7 May, 2001

Contents

1 Purpose

The purpose of this document is to define the development plan for AIPS++ release v1.6, currently scheduled for October 15, 2001.

2 Release priorities

The highest priority for the project in development cycle v1.6 is to continue the process of scientific integration. This includes:

1. Scientific completeness: Demonstrated end-to-end reduction capabilities for targeted instruments in all scientifically important observing and reduction modes.
2. Usability by the astronomical community: Quality of the user interface, applications presentation and user documentation.
3. Robustness, correctness and accuracy: Prove the correctness and robustness of existing capabilities and ensure comparable performance to other disk-based packages.
4. Continued deployment to an expanded user base: Continue to increase the scientific user base for AIPS++.

Maintaining a sound infrastructure in the project is important however, and infrastructure work has been scheduled for this cycle carefully; primarily in areas which are vital for the long-term vitality of the project, or which are in the critical path for application development.

3 Introduction

Development targets are listed separately for each major development area in AIPS++. These targets cover only the development cycle through October 2001, and do not include longer-term items, which are tracked separately for consideration in subsequent development cycles (and can be found in docs/project/deferred_1.6.ps.gz). There are 21 weeks in this development cycle, between the week of May 7, 2001 and the expected code-freeze on 1 October 2001, but deductions for defect correction and code review (30%; 1.5 days per week), user support (up to 20%), and where applicable, science time (25%), have been made. The exact numbers vary by developer and institution, and the best estimate has been made in each case. Variability in these estimates is accommodated by the inclusion of medium prioirty targets, which may need to be carried over into the next cycle. Each target is assigned a priority as high (**H**), which implies that the target is expected to be completed this cycle, and medium (**M**), which is work to be done on an as-available basis. Note also, that not all developers are available to the project on a full-time basis, and have reduced commitments as a result.

Abbr.	Developer	Affiliation
AK	Athol Kemball	NRAO
AS	Anuj Sarma	NCSA
BG	Bob Garwood	NRAO
DK	David King	NRAO
DM	Dave Mehringer	NCSA
DrS	Darrell Schiebel	NRAO
GvD	Ger van Diepen	ASTRON
GM	George Moellenbrock	NRAO
HR	Harold Ravlin	NCSA
JB	Jim Braatz	NRAO
JN	Jan Noordam	ASTRON
JM	Joe McMullin	NRAO
KG	Kumar Golap	NRAO
MM	Malte Marquarding	ATNF
MW	Mark Wieringa	ATNF
NK	Neil Killeen	ATNF
PC	Paulo Cortes	NCSA
PT	Peter Teuben	BIMA
RM	Ralph Marson	NRAO
RP	Ray Plante	NCSA
OS	Oleg Smirnov	ASTRON
TC	Tim Cornwell	NRAO
WB	Wim Brouw	ATNF
WY	Wes Young	NRAO

4 Single-dish

1. Complete GBT MS filler: Including: i) complete IF/LO conversion; ii) GBT GO FITS file to MS; iii) Tcal to SYSCAL sub-table; iv) spectrometer and spectral processor backend support; v) track M&C FITS file changes; vi) improve reliability and robustness in on-line

mode; vii) complete holography support; viii) document filler internals in preparation for filler transfer to GBT science support. (**BG, H, 9 wk**).

2. SD MS audit: Audit the MS produced by the GBT MS filler for scientific completeness. (**BG, H, 2 wk, AK, H, 1 wk**).

3. SD calibration: Complete basic SD calibration modes using synthesis infrastructure(**JM, H, 6 wk, AK, H, 2 wk**).

4. ssa enhancements: Complete ssa improvements. (**JM, H, 2 wk**).

5. IARDS: Interim real-time display system for the GBT (**JB, H, 4 wk**).

6. Holography support: Full holography support using synthesis infrastructure. (**JB, H, 4 wk**).

7. Document DCR tool: Document DCR tool and heuristics(**JB, H, 3 wk**).

8. SD imaging: Including: i) verify single-dish imaging speed for spectral line data; ii) demonstrate combined VLA and GBT imaging. (**TC, H, 6 wk**).

5 Synthesis

1. Complete the FITS-IDI filler: Including: i) support for all required MS sub-tables; ii) VLBA correlator quantization corrections; iii) testing with SMA and VLBA data. Work with SMA in these tasks. (**GM, H, 2 wk**).

2. BIMA filler revisions: Continued improvements in the BIMA filler. (**RP, H, 1 wk**).

3. Multi-MS support in imager and calibrater: Make existing multi-MS support available in the calibrater and imager tools. (**KG, H, 3 wk**).

4. Support for variable spw: Ensure proper functioning of the synthesis tools with variable-shaped spectral windows. Includes resolution of all current defects in this area (**KG, H, 2 wk**).

5. Implement initial compressed data format: Implement initial solution to the MS data volume problem caused by the MODEL_DATA, CORRECTED_DATA and IMAGING_WEIGHT scratch columns. (**GvD, H, 2 wk, AK, H, 1 wk**).

6. msplot defect correction: Fix all current, high-severity msplot defects. (**RM, H, 2 wk**).

7. Autoflag interface revisions: Change autoflag.g Glish interface to be compatible with AIPS++ user interface conventions. (**OS, H, 1 wk**).

8. Autoflag C++ code review: Complete preparation of autoflag C++ code for AIPS++ code review. (**OS, H, 2 wk, AK, H, 1 wk**).

9. Requirements for uv-component fitting: Finalize requirements and implementation document for full uv-component fitting. Including list of required changes to component models. (**WB, H, 1 wk**).

10. Analytic errors in simulator: Add basic analytic errors to simulator, including: i) Chebyshev polynomial in amplitude, phase, rate and delay (G Jones); ii) polynomial bandpass (B Jones); iii) Chebyshev polynomial in instrumental polarization (D Jones). Requires Chebyshev polynomial implementation. (**RP, H, 2 wk**).

11. calibrater enhancements: Includes: i) identified improvements in the MS access classes; ii) full interpolation and parametrization modes. (**GM, H, 4 wk**).

12. Ionosphere C++ code review: Prepare Ionosphere C++ code for code review. (**OS, H, 2 wk**).

13. Ionsophere correction testing: Test first-order FJones correction on WSRT and other data. (**OS, H, 3 wk**).

14. Second-order ionosphere corrections: Derive initial second-order FJones corrections to the PIM model. (**OS, H, 6 wk**).

15. Automated wide-field imaging: Refine the packaging of the existing wide-field imaging capabilities. Includes: i) improved mask setting, including automated options; ii) improvements to dragon.g. (**KG, H, 3 wk**).

16. SD mosaic combinations: Improved single-dish mosaicing data combination options. (**KG, H, 3 wk**).

17. Planet flux density calibration: Develop flux density calibration using planets. (**RP, H, 2 wk**).

18. imager test scripts: Revise all imager test scripts to use simulated data for test generation and verification. Includes coverage of all observing contexts (single-field, multi-field, mosaiced, and wide-field), and all available deconvolution algorithms. (**DM, H, 4 wk, KG, H, 1 wk**).

19. ATCA test scripts: Test scripts and associated data demonstrating ATCA reduction for continuum, continuum polarimetry and spectral line reduction modes. (**MW, H, 3 wk**).

20. WSRT test scripts: Test scripts and associated data demonstrating WSRT reduction for continuum, continuum polarimetry and spectral line reduction modes. (**JN, H, 4 wk**).

21. BIMA test scripts: Additions to the existing BIMA test scripts to cover new observing modes. (**DM, H, 2 wk**).

22. VLA test scripts: Expand the existing VLA test scripts to cover spectral-line modes and additional data sets. (**AK, H, 1 wk**).

23. IRAM AIPS++ test: Verify AIPS++ reduction using IRAM test data. (**AK, H, 5 wk, KG, M, 2 wk**).

6 Documentation

1. Editing of Getting Results: Includes: i) edit existing and new GR chapters to conform to a common framework; ii) identify missing areas in GR; iii) incorporate user group feedback in the editing process. (**NK, H, 4 wk**).

2. Single-dish documentation editing: Improve the existing SD documentation and add new material (**JM, H, 2 wk**).

3. ATCA Getting Results chapter: Getting Results chapter for ATCA continuum, continuum polarimetry and spectral line reduction modes. (**MW, H, 2 wk**).

4. BIMA Getting Results chapter: Getting Results chapter for BIMA continuum, continuum polarimetry and spectral line reduction modes. (**AS, H, 2 wk**).

5. VLA Getting Results chapter: Edit and extend the existing VLA Getting Results chapter to cover spectral line, and improve existing material. (**GM, H, 2 wk**).

6. Recipes: Revitalize the existing recipes repository. Change the format as required, and add new recipes in all areas. (**GM, H, 3 wk**).

7 Applications integration

This includes development aimed at improving the overall integration of applications in AIPS++ in a common framework, using common interfaces and services.

1. History propagation: Develop infrastructure for history propagation. (**RP, H, 3 wk**).

2. Programmer documentation: Update the existing programmer documentation. (**AK, H, 1 wk**).

8 Build and code distribution system

This covers all system work, excluding maintenance and support of the basic library.

1. Documentation system: Develop written requirements for the documentation build system; audit current capabilities. Explore central documentation build and remote distribution using cvsup (**WY, H, 2 wk**).

2. Glish documentation utilities: Develop utilities to assist in generating Glish-level tool documentation, and to verify completeness and correctness of existing Glish documentation. (**WY, H, 3 wk**).

3. Build and release maintenance: Maintenance of existing builds under Linux, Solaris and IRIX. Patch generation and distribution for the release master. (**WY, H, 4 wk**).

4. Solaris native C++ port: Complete port of AIPS++ to the Solaris native C++ compiler. Document all required changes as a note. (**WY, H, 2 wk**).

9 Glish

1. GlishTk performance: Continue evaluation of GlishTk performance factors, including: i) dependence on pipe buffer size; ii) other short-term optimizations to mitigate dynamic scheduling losses. (**DrS, H, 3 wk**).

2. Glish threading: Assess cost of Glish threading. Prototype threading in parts of the Glish library and determine likely GTk performance enhancements. (**DrS, H, 3 wk**).

3. fail on return: Implement fail on return handling in Glish at function end-points. (**DrS, H, 1 wk**).

4. Defect backlog: Address current defect backlog, including: i) slider problems; ii) hourglass packing; iii) remaining JIVE reference counting defects. (**DrS, H, 3 wk**).

5. Tab widget: Add the Tab widget. (**DrS, H, 2 wk**).

10 User interface and tasking

1. GUI performance improvements: Improve GUI performance by investigating the use of larger-scale widgets, and/or optimization of existing Glish GUI code. (**RM, H, 3 wk**).

2. Data item propagation: Ensure use of all existing data items throughout the user interface, where possible. (**RM, H, 2 wk**).

3. guientry defects: Resolve all current guientry.g defects. (**RM, H, 2 wk**).

4. toolmanager enhancements: Includes: i) allow aipsrc customization of tools visible in toolmanager; ii) trial implementation of default sets for one tool. (**RM, H, 2 wk**).

5. Revised tasking requirements: Develop requirements for an improved Tasking system. (**AK, H, 1 wk**).

6. CORBA evaluation: Develop implementation plan for CORBA adoption in the Tasking system and Glish. Includes some CORBA prototyping. (**DrS, H, 3 wk**).

11 Basic library

1. Overhaul Functionals: Overhaul Functionals so that they can be used more extensively in other parts of the library, including Fitting. Includes an implementation of optimized auto-differentiation. (**WB, H, 4 wk**).

2. C++ standard proposal: Draft a proposal outlining strategies for implementing further C++ standard library compliance in AIPS++. (**WB, H, 1 wk**).

3. C++ interface to non-linear fitting: Improve the C++ interface to existing non-linear fitting capabilities. Requires improvements in Functionals (**WB, H, 2 wk**).

4. Deep copy: Allow deep copy of tables. (**GvD, H, 1 wk**).

5. Measures frame handling: Improved frame handling in Measures. (**WB, M, 1 wk**).

12 Images

1. LEL spectral index function: Implement a spectral index function in LEL (**GvD, H, 0.5 wk**).

2. Support non-cardinal axes: Support slicing along non-cardinal axes, which is required for the pvslice visualization application. Requires addition of Lattice interpolation along non-cardinal axes. (**NK, H, 2 wk, GvD, H, 3 wk**).

3. Mask generation from cubes: Support mask generation from cubes, as required to support a range of visualization and image analysis applications. (**NK, H, 3 wk**).

4. Multi-dimensional LEL: Add facility for handling multi-dimensional images in LEL. (**GvD, H, 1 wk**).

5. Unified logsink use in Image classes: Unify logsink use in derived Image classes. (**GvD, H, 1 wk**).

6. Miriad image handling: Support MIRIAD image handling. (**PT, M, 4 wk**).

13 Visualization

This includes work in the Display Library (DL), and in applications using this library.

1. pvslice application: Implement high-level application supporting pv-slicing. (**MM, H, 4 wk**).

2. Color bar wedge: Implement color bar wedge support in the DL and viewer. (**MM, H, 2 wk**).

3. Multiple regions per panel: Support multiple regions per display panel. (**MM, H, 1 wk**).

4. Unify CachingDD and PADD code branches: Unify the existing caching and principal axes DisplayData code branches. Preserve the current API for each branch in the interim to allow uv DD development to continue. (**MM, H, 5 wk**).

5. Proposal for coordinates on world canvas: Prepare a proposal/design for the implementation of coordinates on the WorldCanvas. (**MM, H, 2 wk**).

6. uv DisplayDatas: Complete: i) MSAsRaster; ii) MSAsContour; iii) MSAsXY; iv) TableAsRaster; v) TableAsContour; vi) TableAsXY. (**DK, H, 6 wk**).

7. Integrated uv-visualization: Provide full integration of the uv-DD through to the viewer, and make available as re-usable uv-visualization components. (**DK, H, 4 wk**).

8. Editing event-handlers: Develop editing capabilities as DL event handlers. (**DK, H, 4 wk**).

9. Animation increments in world coordinates: Allow animation increments in world coordinates. (**MM, M, 1 wk**).

10. OpenGL PixelCanvas: Continue development of an OpenGL or vtk PixelCanvas. (**HR, M, 4 wk**).

14 Parallelization and high-performance computing

1. Revise test scripts for parallel problems: Revise the test scripts for large parallelized algorithms to use simulated test data and verification. (**DM, H, 2 wk**).

2. Key science processing and profiling: Reduce the identified key science datasets on DSM and cluster systems, with and without associated performance and I/O profiling. (**AS, H, 6 wk**).

3. IA64 port: AIPS++ port to the IA-64 Linux cluster. (**PC, H, 4 wk**).

4. Parallel I/O: Initial implementation of parallel I/O capabilities. (**WY, H, 3 wk**).