Updating AIPS++

Wesley Young
NRAO

7 Nov, 2002

Contents

1 Introduction

The current mode of applying patches to AIPS++ is inadequate for addressing user requests for defect resolutions and enhancements. More frequent patches would be helpful but they are difficult to apply and are often precluded by the project's desire to only apply bug fixes to release. A possibly more useful model is to allow users access to the project's stable builds.

2 Non-consortium User Updates

In our current scheme, consortium sites and those sites that build from source have access to daily changes in the code tree. Available to these sites, is a stable build mechanism inhale -b. Users who install from CD don't readily have this option. Currently linux users can use the inhale binary option provided they make adjustments to their systems. The proposed scheme would simplify and widen the availabilty of binary updates.

2.1 A New Update Scheme

stable - base Our current stable.

stable - updated An updated stable with new features, enhancements, or bug fixes, stuff is plugged in but must pass all stable tests to be moved into updated stable tree.

beta First stable after slushy code freeze

released Beta that meets all release criteria.

additional explanation We continue to test for stables in our current mode, this produces our stable-base. The stable-base is copied to stable-updated. Stable-updated contains apps and scripts that are tested to work with the current stable but are not in the current stable. Essentially we replace the app and/or scripts with new ones and run an assay on the "updated stable" to make sure it still passes, if so the stable is then updated.

2.2 ISO images

We'll have ISO release images available for base stables, betas, and releases. This will require us to move these ISO images to ATNF, UIUC, and ASTRON.

2.3 Use cvs/cvsup

Put stable bin files (base and updated), libexec files, and shared libraries under cvs/cvsup.

2.4 Create a new script, aupdate

Options

stable - get updated stable version, with a noupdate option to get only the base stable.

beta - get beta version

release -only get released version (includes patches)

(no)bin - get updated bin files

(no)libexec - get updated libexec files

(no)lib - get updated shared libraries

(no)docs - get updated docs

(no)basedata - get updated base data

(no)auxdata - get updated aux data (not aux could be telescope specific)

(no)code - get updated code tree

usefile(s) - update from a local tar file.

default - is stable binary libexec lib basedata docs

silent (-q) - Just do it don't ask for confirmation.

Aupdate would normally use cvsup to update various directory trees. The usefile option allows downloading a tar file and using the tar file to update the system for those machines that don't support cvsup.

2.5 What the user does

1. Install release
2. Choose one of three update options
   (a) To update to the latest alpha release (stable): aupdate
   (b) To update to the lastest beta release (post slushy freeze): aupdate beta
   (c) To get the latest release (including patches) aupdate release
3. aupdate will tell the user something like

```
$aupdate
NOTICE - NOTICE - NOTICE

AIPS++ binary files tend to be big, you should really be on a
decent network link to down load the updates

AIPS++update ->
```

```
   Updating the following linux/bin linux/lib linux/libexec docs from stable
Is this correct?

If y then do the update, else if n then
AIPS++update ->
   Please specify stable|beta|release (no)bin (no)lib (no)libexec (no)docs
   (no)code (no)basedata
update?

Repeat until we get a y.  Using a -q or silent avoids all these questions,
and figures out what to do from the installation.
```

2.6 Notes

1. An alternative would be for the user to download the update files and use the usefile(s) option.
2. The user who owns the AIPS++ installation must run the aupdate script.

3 Action Items

1. Create the stable - update tree.
2. Put stable base/update under cvs/cvsup.
3. Write the aupdate script.
4. Create the tar files for systems that don't support cvsup.
5. Set the test procedures for adding files/apps to the updated stable.
6. Generate additional ISO images for the base stable.