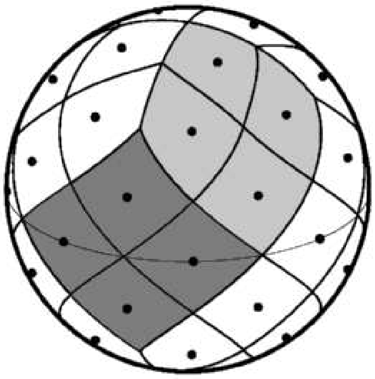
HEALPix Facility Installation Guidelines



Revision: Version 2.00; August 31, 2005

Prepared by: Anthony J. Banday, Matthias Bartelmann, Eric Hivon, Benjamin D. Wandelt, Frode K. Hansen and Krzysztof M. Górski

Abstract: This document describes the installation for the **HEALPix** facilities.

Contents

1 Introduction . . . 3

2 Installation Requirements . . . 3

3 The Installation Procedure . . . 5

3.1 ./configure . . . 6

3.2 make . . . 7

3.3 Testing the installation . . . 8

3.4 config_preview . . . 9

4 Upgrading from 1.2 to 2.0 . . . 9

5 A Note on *Re*-installation . . . 9

6 Troubleshooting and further information . . . 9

6.1 CFITSIO linking problems . . . 10

6.2 CFTISIO and Debian/Linux . . . 10

6.3 CFITSIO problems on systems with 64 bit architecture . . . 10

6.4 `diff` shows that the test files are different from the supplied files . . . 10

6.5 MIPSPro Compilers on SGI machines . . . 11

6.6 Try `unlimit` . . . 11

6.7 `hidl` usage . . . 11

7 Appendix I: Changes and New Features in Version 2.00 . . . 12

8 Appendix Ia: Changes and New Features in Version 1.20 . . . 13

9 Appendix Ib: Changes and New Features in Version 1.10 . . . 14

1 Introduction

In this document the installation procedure for the **HEALPix** distribution is outlined. **HEALPix** comprises a suite of Fortran 90, C++ and IDL routines (together with one C-library which is also provided) providing both stand-alone facilities and callable subroutines as an alternative for those users who wish to build their own tools. A set of C and Java subroutines and functions is also provided.

The distribution can be downloaded as a gzipped and tarred file, which can be unpacked by executing the commands

```
% gunzip Healpix_2.00.tar.gz
% tar -xpf Healpix_2.00.tar
```

The unpacked distribution has a directory structure as shown in Figure 1.

As with most freely available software, the distribution comes with caveats, the major one being that although we have attempted to automate the installation as much as possible, not all eventualities can ever be foreseen. We have tested the installation on the following platforms:
AIX, IRIX, IRIX64, Linux, SunOS, ALPHA and Darwin

There may be problems in the facility build due to the local system configuration which is beyond our control.

2 Installation Requirements

The major part of the **HEALPix** distribution is written in both **Fortran 90** and **C++** and so the appropriate compilers must be present (Linux and Darwin users should look at Section 7 about free F90 compilers). Many visualisation tools and map manipulation routines are provided in **IDL** (please note that at least version 5.0 is required). Some of the **HEALPix** routines are now also available in C (for further details on these routines and their installation, please report to the document ”C Subroutines Overview” and to `src/C/README`).

This section and the next focus on the compilation and installation of the ***Fortran 90*** *routines. For more information on the* ***C++*** *routines and their installation see* `src/cxx/README.compilation`.

The configure script is written in the Bourne shell. The script attempts to generate a `Makefile` which is tailored to one of the above Operating Systems (OS’s) and using `Makefile.in` as a template for non-system specific statements. Only the basic UNIX make facility is required to build the software, although we do still recommend the GNU make facility (`ftp://ftp.gnu.org/gnu/make/`). In addition, a system Profile and IDL startup file are generated/included in the distribution. These automatically establish various environment variables and aliases to make the use of the **HEALPix** package simpler.

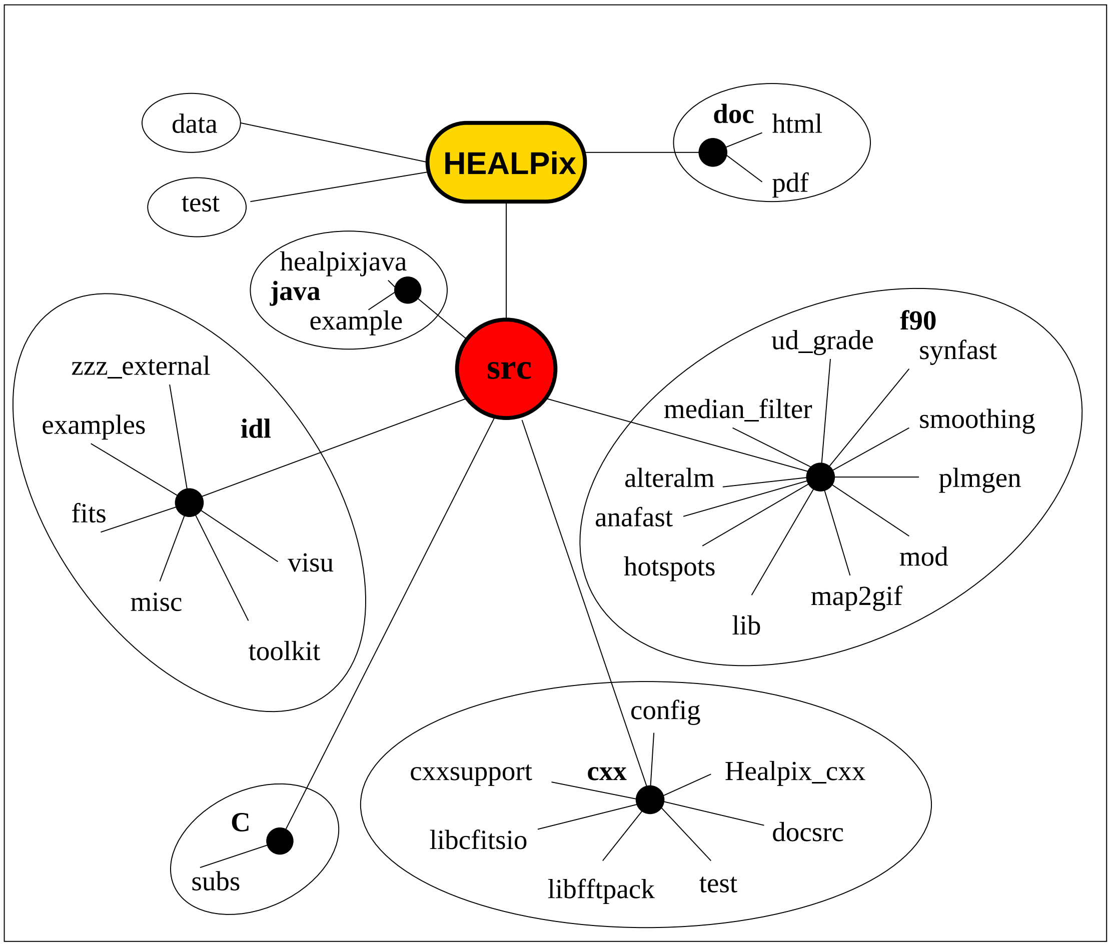


Figure 1: The directory structure for the **HEALPix** distribution.

The **HEALPix Fortran 90**, **C++** and **C** distributions also require the publicly available CFTISIO library:

Software Package	**Source**
CFITSIO V 2.xx library	`http://heasarc.gsfc.nasa.gov/docs/software/fitsio/`

And finally, the **IDL** visualization software is commercially available at

Software Package	**Source**
IDL V 5.x or 6.x	`http://www.rsinc.com/`

(versions 5.3 to 6.2 have been extensively used during the development of the **HEALPix** IDL codes).

As it was already the case in version 1.20, users no longer need to acquire the IDL Astronomy User's Library (`http://idlastro.gsfc.nasa.gov/homepage.html`) or the COBE (IDL) Analysis Software (`http://www.gsfc.nasa.gov/astro/cobe/cgis.html`), although we do recommend these packages to the user. The few routines required for version 2.00 are contained in a new subdirectory `Healpix_2.00/src/idl/zzz_external`. These procedures are included in the **HEALPix** package unchanged and solely for the purpose of making it self contained. In this way, we remove the burden of installation of additional libraries from the end user.

A parallel implementation (based on OpenMP, for shared memory architectures) of the Spherical Harmonics Transforms involved in `synfast, anafast, smoothing, plmgen` and `alteralm` is now available by default and can be readily compiled and used with the standard installation script.

A set of routines with MPI parallelization (for distributed memory architectures) is also available for Spherical Harmonics Transform, thanks to the work of H.K. Eriksen (h.k.k.eriksen@astro.uio.no) and Snorre Boasson (ITEA, NTNU). See the F90 subroutines documentation for more information on how to use those routines in your code.

We found that it was remarkably difficult to find random number generators in the public domain which are simple yet powerful and easy to use. We are providing one (both in C++ and F90) which is an adaptation of an xorshift generator described in Marsaglia (Journal of Statistical Software 2003, vol 8). It has a theoretical period of $2^{128} - 1 \approx 3.4\ 10^{38}$.

3 The Installation Procedure

If the user has one of the supported OS's, then installation proceeds utilizing the following commands. If your OS is not supported, the configuration step should be omitted, `Makefile.in` should be copied as `Makefile` and explicitly tailored to the user environment.

`% ./configure`	uses `Makefile.in` as a template to build the correct Makefile (from user inputs as required)
`% make`	builds all the facilities
`% make test`	runs a test script (see below)
`% make clean`	removes object files
`% make vclean`	removes object files, executables and libraries
`% make tidy`	same as above and restores the directories to the state of the original distribution
`% ./config_preview`	configure the IDL previewer

3.1 ./configure

When you run `./configure` on a supported system you will be prompted to enter compiler optimisation flags. We have not attempted to provide the best optimisation flags for all operating systems. The configure script will have a guess at optimisation options for some systems, but it is up to the user to figure out an optimal set[1] From our experience, we have not found significant accumulation of numerical error even when using the most aggressive optimisation level available.

A feature introduced in the previous release and enhanced in the current one, is that the configure script creates a shell profile according to shell type in which various environment variables and aliases are defined for your convenience. If you agree upon prompting, it will also change your default system profile during installation to automatically source this profile. If you do not agree to this change, you will need to explicitly source the profile for any session in which you intend to run **HEALPix** facilities. **In particular, you will have to make sure that the** `HEALPIX` **system variable is correctly defined (as the full path to the HEALPix directory) before running the package**.

And finally, the current `./configure` script allows several compilations of the Healpix routines to coexist by letting the user choose the name of directories containing the executables, libraries and include files.

A typical configuration session for a user running on a PC running under Linux, with the Intel Fortran Compiler (version 8.1 or 9.0), is as follows:

```
%./configure
you seem to be running Linux
enter name of your F90 compiler (): ifort
  Note: your Fortran compiler is Intel Fortran Compiler
  compiled Healpix products will be:
BINDIR =  ./bin[suffix]
INCDIR =  ./include[suffix]
LIBDIR =  ./lib[suffix]
 and the Makefile will be copied into Makefile[suffix]
enter suffix for directories (): _ifort
 compiled Healpix products will be:
BINDIR =  ./bin_ifort
INCDIR =  ./include_ifort
LIBDIR =  ./lib_ifort
Warning: The following directories could not be found:
./bin_ifort
./include_ifort
./lib_ifort
```

[1] In particular, the Intel Fortran Compiler, available for free for PC's with Intel-like processors, have a set of optimization options for each of Intel processor families (Pro, II, MMX, 4). Please report to the online help (`ifort -help`) or PDF documentation (`/opt/intel_fc_80/doc/` or `/opt/intel/fc/9.0/doc`) for further information.

```
Should I attempt to create these directories (Y|n)?
 enter compilation flags for ifort compiler (-I$(INCDIR) -Vaxlib -cm -w -vec_report0):
enter optimisation flags for ifort compiler (-O3):
  Fortran code will be compiled with ifort -O3 -I$(INCDIR) -Vaxlib -cm -w -vec_report0
enter name of your C compiler (cc):
enter compilation/optimisation flags for C compiler (-O):
  C subroutines will be compiled with cc -O
enter command for library archiving (ar rv):
enter full name of cfitsio library (libcfitsio.a):
enter location of cfitsio library (/usr/local/lib):
 The Spherical Harmonics Transform routines used by synfast/anafast/smoothing/plmgen
and some routines used by ud_grade and alteralm respectively
have a parallel implementation (based on OpenMP).
 It has currently been tested on IBM/xlf and Linux/ifc systems/compilers
Do you want to use :
 0) the standard scalar implementation ?
 1) the parallel implementation (slightly slower in single CPU usage with some compilers)
Enter choice                                      (0):

The following line should be inserted into your shell profile
(/home/user/.cshrc):

  source /home/user/Healpix_2.00/bin_ifort/profile.HEALPix

profile.HEALPix contains the following commands:

# tcsh configuration for HEALPix
setenv HEALPIX /home/user/Healpix_2.00
setenv HEXE    /home/user/Healpix_2.00/bin_ifort
setenv PATH    ${PATH}:${HEXE}
alias  hidl    "idl /home/user/Healpix_2.00/src/idl/HEALPix_startup"
alias  hidlde  "idlde /home/user/Healpix_2.00/src/idl/HEALPix_startup"

Do you want this modification to be done (y|N)?
Configuration finished.
Now run "(GNU)make" to build the package,
    and "(GNU)make test" to test it.
You can configure the IDL previewer by running ./config_preview
```

3.2 make

Please neglect any possible warnings at compile time. If you run into trouble please refer to the section **Troubleshooting and further information**.

After running `make`, the user must re-login to ensure that the new profiles built by the installation procedure are correctly sourced. Only then will the user have full access to the specific **HEALPix** environment variables etc.

3.3 Testing the installation

In order to test the success of the installation, one should run the test script as outlined above (ie, `make` *[-f make_file_name]* `test`). This proceeds as follows:

- moves to `test/` subdirectory
- runs `synfast` (with parameter file `syn.par`) to produce the polarised sky map file `test_map.fits` and the corresponding a_{lm} coefficients `test_alm.fits`; one can then compare this to `map.fits` and `alm.fits` (supplied)
- runs `map2gif` to make the image `test_map.gif`; one can then compare this to `map.gif` (supplied)
- runs `smoothing` (with parameter file `smo.par`) to make `test_sm.fits`; compare this to `map_sm.fits` (supplied)
- runs `map2gif` to make the image `test_sm.gif`; compare this to `map_sm.gif` (supplied)
- runs `ud_grade` (with parameter file `udg.par`) to degrade the map `test.fits` to $N_{side} = 4$. The output is `test_LOres.fits`; compare this to `map_LOres.fits` (supplied)
- runs `map2gif` to make the image `test_LOres.gif`; compare this to `map_LOres.gif` (supplied)
- runs `hotspot` (with parameter file `hot.par`) to produce a map of the local extrema in `test_LOres.fits`. The outputs are `test_ext.fits`, `test_min.asc` and `test_max.asc`. Compare this to `map_ext.fits`, `min.asc` and `max.asc` (supplied)
- runs `map2gif` to make the image `test_ext.gif`; compare this to `map_ext.gif` (supplied)
- runs `anafast` (with parameter file `ana.par`) to compute the power spectrum of the sky map `test.fits`. The output is `test_cl.fits`; compare this to `cl_out.fits` (supplied)
- runs `alteralm` (with parameter file `alt.par`) to modify the a_{lm} coefficients read from `alm.fits` into `test_almdec.fits`; compare this to `almdec.fits` (supplied)
- runs `median_filter` (with parameter file `med.par`) to produce in `test_mf.fits` the median filtered map of `map.fits` ; the former can be compared to `map_mf.fits` (supplied).
- runs `map2gif` to make the image `test_mf.gif`; compare this to `map_mf.gif` (supplied)
- comes back to main **HEALPix** directory

The supplied data and parameter files are present in the `/Healpix_2.00/test` directory, and this is also where the results files are created.

Note: the input power spectrum (in `cl.fits`) used to generate the test maps is currently the WMAP 1yr best fit, in $(\mu\mathrm{K})^2$, and is therefore different from the one included in previous releases (that can still be found in `cl_old.fits`). See `http://lambda.gsfc.nasa.gov/` for details on WMAP and its data products.

In order to test the new **HEALPix** profile set-up one can then attempt to run any F90 facility from any directory on your system. Similarly, IDL should be tested by invoking `hidl` or `hidlde`.

3.4 config_preview

The **HEALPix** IDL visualisation routines can generate PostScript, GIF or PNG files for printing or insertion in an electronic document. They offer the possibility to automatically “preview” those files as soon as they are created, so that the user does not need to start the viewing process in a different window. To do this, an external viewing code (like Ghostview, xv, netscape, ...) has to be called by IDL. As the choice of viewing codes varies from platform to platform, the `config_preview` script is designed to choose the codes to be used. It will create or update the `src/idl/visu/idl_default_previewer.pro` which is read by IDL.

4 Upgrading from 1.2 to 2.0

The internal structure of release 2.0 is quite different from release 1.2 and to avoid confusion during the compilation we highly recommend to put the new release in a *different* directory, rather than putting the new package on top of the old one. If you actually change the name of the ’active’ **HEALPix** directory care must be taken that all references to the old directory are removed from your system profile before adding the new ones (see Note on *Re*-installation).

5 A Note on *Re*-installation

As a result of the line added to your shell profile which explicitly sources the **HEALPix** profile, care must be taken if the package is reinstalled in a different directory. If such reinstallation is desired, the included line must be removed from your system profile, allowing the corrected version to be added.

6 Troubleshooting and further information

This section contains a list of difficulties which we have dealt with. It is by no means exhaustive. A troubleshooting forum has been established at `http://healpix.jpl.nasa.gov`, where we list current questions and solutions to known problems (for a given release).

6.1 CFITSIO linking problems

A particular problem encountered with the CFITSIO Version 2.0 release relates to the inclusion of various libraries within the system release for a given machine. This led to some modifications to the Makefile to include the specific library links `-lm -lnsl -lsocket` on SunOS, but only `-lm` for IRIX64. If your OS is not completely supported by the distribution, you may find this as one source of errors. The CFITSIO developers recommend compilation of the `testprog` routine. Inspection of the libraries linked after executing the `make testprog` statement will reveal those you need to include in the Makefile.

6.2 CFTISIO and Debian/Linux

Some problems have been reported on Debian/Linux systems during the linking to the CFITSIO library shipped with Linux. If these problems occur, try to recompile the CFITSIO library from scratch before linking to **HEALPix** .

6.3 CFITSIO problems on systems with 64 bit architecture

On a 64-bit architecture such as IRIX64, CFITSIO will have to be compiled in the same binary format as the **HEALPix** codes. This can be achieved by typing the following on the command line in the CFITSIO directory:

```
rm config.cache
setenv CC 'cc -n32'
./configure
make
```

Alternatively you can replace the `-n32` with `-64`. You can then force compilation to the same binary format by entering either `-n32` or `-64` when asked for the optimisation options in the **HEALPix** configure script.

6.4 `diff` shows that the test files are different from the supplied files

This by itself is no cause for concern. When comparing using a `diff` on the test files will most likely report a difference even when the installation has been successful. This may be due to the fact that different installations have different floating point representations. Also, the FITS files carry date information.

6.5 MIPSPro Compilers on SGI machines

Regrettably, the MIPSpro Compiler Version 7.20 has a compiler bug which cause run-time memory faults. We have not found any problems with Version 7.2.1.1m.

6.6 Try `unlimit`

If you have unforeseen problems at runtime try `unlimit`. It sometimes helps.

6.7 `hidl` usage

We have found that in very rare cases the alias `hidl` is not recognised by the user's system. Usually, this is related to the local system's IDL script. A quick-fix is achieved by setting the environment variable `IDL_STARTUP` to be equal to the **HEALPix** startup file `HEALPix_startup` **including** the directory path to the file. This enables the user to access the **HEALPix** IDL procedures simply by invoking IDL. For example, in the typical installation documented above for a user running the tcsh shell, the command

```
setenv IDL_STARTUP /disk1/user1/HEALPix_2.00/src/idl/HEALPix_startup
```

should be issued (or added to the user's shell profile).

If the user already has an IDL startup file, then this should be merged with `HEALPix_startup`. This temporary solution does mean that the **HEALPix** IDL procedures are available in the `IDL_PATH` at all times, which may lead to conflicts with user-defined procedures. The `hidl` invocation was intended to circumvent these issues, allowing **HEALPix** IDL procedures to be available only when desired.

A proper fix requires the user to ask the local system administrator to adjust the local IDL script.

7 Appendix I: Changes and New Features in Version 2.00

- Free / Open source software license
- New web page: `http://healpix.jpl.nasa.gov`
- HTML and PDF documentations are available in `[HEALPix]/doc/html/main.htm` and `[HEALPix]/doc/pdf/`
- C++ implementation of almost all codes and routines
- Fortran90:
  - 2 to 3 times faster Spherical Harmonics tools (ie, synfast, anafast, smoothing, plmgen)
  - parallelization for distributed memory architectures
  - More **free** Fortran90/95 compilers supported:
    - Intel Fortran Compiler for linux based computers (version 8.1 or 9.0) `http://www.intel.com/software/products/compilers/downloads/forlin.htm`
    - GNU Fortran 95 compiler (gfortran) included in GNU Compiler Collection *GCC* version 4.0.0 and up and available for Linux, Mac OSX, Windows, Sun ... platforms `http://www.gnu.org/software/gcc/fortran/` Please note that only the most recent CVS versions of gfortran (Aug 2005 and later) compile HEALPix correctly!
    - g95 compiler available for Linux, Mac OSX, Windows, Sun and HP platforms `http://g95.sf.net`
- C++ and F90 codes
  - (almost) identical user interface
  - parallelization for shared memory architectures
  - *new facilities:* median filtering of **HEALPix** map and alteration/rotation of Spherical Harmonics coefficients
  - single and double precision I/O
  - speed optimization of FITS I/O
  - more platforms/compilers supported (including Mac OS X, 32/64bit architecture, etc.)
  - *new, longer period, random number generator*
  - *linkage to FFTW has been discontinued*

  - precomputed ring weights used for quadrature of the Spherical Harmonics are now available for all resolution parameters $N_{\rm side} \leq 8192$

- Java implementation of pixel tools
- IDL:
  - median filtering facility
  - speed optimization of FITS I/O
  - more options in file reading and visualisation routines

8 Appendix Ia: Changes and New Features in Version 1.20

- An extensive HTML documentation is included in the package (in `[HEALPix]/doc/html/main.htm`). It is also available at the web page `http://healpix.jpl.nasa.gov`, as well as the PostScript documention.
- The convention used for the normalisation of the polarization power spectra has been changed to match that of CMBFAST. This subject and its implications are described in details in the **HEALPix** Primer
- The pixel window functions are now provided for both temperature and polarization. Re-calculation of the temperature window for $N_{side} > 128$ induce a slight change of these functions (see the “**HEALPix** Primer” for details).
- Because of its obsolete F66 features that limited its portability with current f90/f95 compilers, the FFTPACK module used in version 1.10 is no longer in use. **HEALPix** 1.2 can either use a different self contained FFT module shipped with the package, or use the freely available **FFTW** library (see section 2). The script file asks which FFT the user wants to use. If selected, FFTW must be correctly installed (in double precision) before installing Healpix.
- It is now possible to use an OpenMP parallelisation of the Spherical Harmonics Transforms (see § 2).
- The compilers now supported under **Linux** systems are
  - Intel Fortran Compiler, available for **free** `http://www.intel.com/software/products/compilers/downloads/forlin.htm` (tested for version 5.01, 6.01 and 7.0)
  - Lahey/Fujitsu Compiler, both **free** trial and commercial versions `http://www.lahey.com/linux.htm`
  - NAGWare f95, commercial `http://www.nag.co.uk/nagware/NP.asp`

  - Portland Group Compiler, commercial `http://www.pgroup.com/`
  - Fujitsu Compiler, commercial

- The **Mac** Operating System **Darwin** is now supported. To this date (Jan 2003) it has only been tested with NAGWare f95 `http://www.nag.co.uk/nagware/NP.asp`.

- A small subset of routines for the pixel to sky coordinate conversion is now available in C (see the documentation ”C Subroutines Overview”).

- New Fortran subroutines have been added or upgraded (see the documentation ”Fortran90 Subroutines Overview”) and most of the Fortran facilities have been upgraded (see the documentation ”Fortran90 Facilities User Guidelines”)

- several Fortran routines have been renamed and merged.

  - The module `wrap_fits` has been renamed `head_fits` to reflect its extended capabilities,
  - some routines no longer in use have been moved to `obsolete` module

- New IDL routines have been added, and several routines have been renamed or merged. See the documentation ”IDL Facilities Overview”

- The configure script automatically creates a (system specific) profile defining useful environment variables and aliases. *In particular, the system variable* `HEALPIX` *must be correctly defined (as the full path to the* ***HEALPix*** *directory) prior to running the pipeline.*

- All legacy f77 or f66 codes have been removed or replaced in order to improve the portability

- The script file has been improved to, among other things, get rid of problems appearing under SunOS

9 Appendix Ib: Changes and New Features in Version 1.10

- Only native `make` is required for building the facilities, although we do still recommend GNU make.

- The configure script automatically creates a (system specific) profile defining useful environment variables and aliases

- An IDL startup file is provided to ease use of IDL tools. Users should invoke IDL usng the command `hidl` to take advantage of these features

- Various bug fixes and code modifications have been completed to improve compatibility and usability of software with NAG F90 and F95 compilers, and the COMPAQ (DEC) Alpha compiler
- New Fortran subroutines and functions include:
  `ang2vec`, `vec2ang`, `npix2nside`, `nside2npix`
- New IDL routines include:
  `add_nside_fits.pro`, `add_units_fits.pro`, `gaussbeam.pro`, `getdisc_ring.pro`, `healpixwindow.pro`, `npix2nside.pro`, `nside2npix.pro`, `remove_dipole.pro`, `today_fits.pro`, `ud_grade.pro`
- Those routines from The IDL Astronomy User's Library and the COBE (IDL) Analysis Software library required for use by the **HEALPix** IDL facilities are bundled with the distribution