
Yaxx Installation Guide

This file describes the installation process for **yaxx** (Yet Another X-ray Xtractor). **Yaxx** is a perl script which automates the process of extracting and fitting Chandra ACIS spectra using CIAO tools and Sherpa. It depends only on free open source software (apart from certain exceptions within CIAO itself).

Requirements

Operating system

Yaxx has been tested on these operating systems:

- **Linux: RedHat (FC2, FC3, FC4), Debian (sarge)**
- **Solaris: Version 2.8.**

Support for MacOS X is awaiting a patch to a bug in the CIAO *dmcoords* tool. Parties interested in using **yaxx** on MacOS should contact the author directly about a possible workaround.

Perl

Perl version 5.8.0 or newer. All non-standard packages used by **yaxx** are included in this distribution.

CIAO

CIAO version 3.3 or newer.

LaTeX

Installation

To install and test **yaxx**, do the following:

1. **Enter the CIAO environment**
   Source the appropriate initialization script (as instructed in the CIAO documentation) to enter the CIAO analysis environment. For `csh` or `tcsh` do

   ```
   source <CIAO_HOME>/bin/ciao.csh
   ```

   Note that if the FTOOLS package is used within the same session as CIAO then the FTOOLS initialization must be done *before* CIAO initialization. Search for ‘‘Other Software Packages’’ in the CIAO documentation for further details.

2. **Check perl version**
   Make sure your perl has version 5.8.0 or newer by checking the output of entering ’perl -v’ at your command prompt:

   ```
   perl -v
   ```

3. **Change dir to install directory**
   Change dir to the directory where you want the **yaxx** source tree to reside. For example if you just go to your home directory with

   ```
   cd
   ```

   then **yaxx** will be installed in *~/yaxx/*.

4. **Untar the source**
   Untar the **yaxx** source tree by doing the following:

   ```
   gunzip --stdout yaxx.tar.gz | tar xvf -
   ```

   This will create a ’yaxx’ directory and put the contents there.

5. **Change directory to the yaxx directory**

   ```
   cd yaxx
   ```

6. **Make the required libraries and modules**
   This is the longest step and can take over an hour on a Solaris machine, but a modern linux box should be done within 20 minutes. At the command prompt enter:

   ```
   make
   ```

   This runs the perl script *install_yaxx.pl* which puts the necessary perl modules in the *yaxx-perl* directory of the source distribution.

   If any library or module fails to compile or pass its internal tests, the install script will ask if you want to quit or continue. Depending on the severity of the failure, it may be possible to push through and still have a working system. Please email me (taldcroft at cfa.harvard.edu) with details if any modules do not compile. See also the section on Known Issues.

7. **Test yaxx**
   Test **yaxx** by doing:

   ```
   make test
   ```

   The final product is a report (*Test/Data/obs877/src1/report.ps*) which summarizes the fitting.

   If this does not work, please email me with the output.

8. **Create an executable in your path**
   The 'make' step also creates a customized shell script 'yaxx' in the source directory which should be used to actually start **yaxx** for analysis. Copy this file to a directory which is in your PATH (e.g. ~/bin) and do 'rehash' (if you are running csh or tcsh).

Known issues

ExtUtils::F77 Version 1.15 and recent Solaris compilers
The Solaris compiler suite *Studio10* is not compatible with ExtUtils::F77 version 1.15 and earlier. We have patched this in the **yaxx** distribution and submitted a bug report to the ExtUtils::F77 developer.

PGPLOT Version 2.19
This module specifies the order of load libraries in a way that is incompatible with the Solaris compilers. We have patched this in the **yaxx** distribution and submitted a bug report to the PGPLOT developer.

Perl installation must be compiled with the current compiler suite
One issue arose because the compiler suite had been upgraded without recompiling Perl. This can lead to obscure errors when building the modules, in particular PGPLOT.

This came up on two Solaris systems. In these cases the issue was resolved without having to recompile perl by reverting to an older compiler with a command similar to:

```
set path = ( /opt/SUNWspro_6.1/bin $path )
```

Details will vary with your installation.

Compress::Zlib
On one linux machine (Fedora Core 4 with Perl 5.8.6) the Compress::Zlib library failed a self-test. The installation process was continued with no further issues and was ultimately successful.