

Changes between v11 and v12

In 1998 we decided to re-engineer XSPEC using modern computer science methods so it could continue fulfilling its role as a mission-independent X-ray spectral fitting program.

The program's internal design, layout, and data structures have largely been rewritten in ANSI C++ using **object oriented design techniques**, generic programming techniques, and **design patterns**. The thoroughgoing reanalysis has also allowed a number of improvements in overall design and, at robustness, as well as maintainability, *without changing the familiar syntax*. With a few exceptions here and there, the new program syntax is fully backward-compatible with that of v11: most of the exceptions support new features that are enhancements (and can be ignored if not relevant to the user's problems). Some features of v11 previously declared to be deprecated have been removed.

At the same time, the core of the XSPEC calculation scheme has been retained, in particular the models library, written almost exclusively in fortran77.

Model implementation has been rewritten to support allow models written not only in single precision fortran, but double precision fortran, C, and C++. Further, XSPEC can now be used as a development environment for local models by allowing recompilation from the command prompt.

In v12, spectra can be fit with more than one distinct model simultaneously, provided separate model components can be assigned distinct response functions. This is particularly useful for spectra from coded aperture masks.

A new internal dynamic expression implementation allows more complex (multiply-nested) models, and also allows parameter links to be polynomial functions of one or more parameters.

Great care has been taken to optimize the program for memory usage and execution speed. A revision of the numerical derivative algorithm has reduced the number of convolution operations required during fitting. On the other hand, v12 performs its calculations in double precision (apart from the models library), and this with the more complex model expression evaluations reduces execution speed. Taken together, v12 should outperform v11 when the number of channels is large and the model to be fitted is relatively simple and should be comparable in other circumstances.

The default fitting algorithm (Levenberg-Marquadt) has been retained intact. New fitting algorithms and objective functions (statistics) may be added to the program at runtime. The CERN Minuit/migrad algorithm has been better integrated into the code and its documentation is now directly accessible during XSPEC sessions.

Type II (multi-spectrum) OGIP files are now fully supported. Multiple ranges can be selected in the data command, and support is present for Type II background and arf files. Observation simulations (the **fakeit** command) now operate on Type II inputs.

The online documentation scheme is now implemented using pdf or html files, replacing the older VMS-style help system. The help scheme can be configured to use external applications such as Adobe Acrobat or the xpdf readers as well as web browsers. Users can document their own local models and tcl-scripted procedures in pdf and html files and add them to the help system.

Plotting within v12 is backward compatible with a few small extensions. Although it is currently implemented using PLT, explicit dependence on the plot library has been removed. This will allow alternative plotting libraries to be used in future. The PLT plotting package is described briefly in Appendix D and in more detail in the “QDP/PLT User's Guide” (Tennant, 1989).

v12 communicates with the user through the familiar command line interface. The input/output streams, however, can in future be easily redirected to communicate with the user through a graphical user interface (GUI).

Finally, the design implements a new error handling system can return the program safely to the user prompt when an error occurs and leave the program in a state from which the user can continue working. Also, for the first time there is now an **undo** command.

Integral Spectrometer/Coded Mask Instrument Support

The **INTEGRAL** Spectrometer (**SPI**) is a coded-mask telescope, with a 19-element Germanium detector array. There are several complications regarding the spectral de-convolution of coded-aperture data. For XSPEC the most obvious problem is the source confusion issue; as there may be multiple sources in the FoV leading to different degrees of shadowing on different detectors. Thus, a separate instrumental response must be applied to a spectral model for each possible source, for each detector. If there are multiple sources in the FoV, then additional spectral models can be applied to an additional set of response matrices, enumerated as before over detector and dither pointing. This capability—to model more than one source at a time in a given minimization procedure—did not exist in XSPEC prior to v12. The other unique aspect of the INTEGRAL analysis is that the background is modeled along with the source(s) in a single de-convolution.

XSPEC analysis of INTEGRAL/SPI data is very different from other instruments is the manner in which the response matrices are handled. Since there are a large number of responses involved in the de-convolution problem, memory use becomes a concern. To load the required response matrices (as XSPEC normally does), would require $\sim(N_{\text{ch}})^2 \times N_p \times N_d$ floating-point memory locations per source. This could become quite large for high-spectral resolution and/or long observation scenarios. To address this problem, a methodology has been developed to reconstruct the required 2-D response matrices from a basis set, consisting of a small number (3) of 2-D objects (template RMFs), and a larger number of 1-D objects (component ARFs). The full matrices can then be reconstructed "on the fly" at the minimization step of the calculation, and discarded after each use. This, in principle, occurs all very transparently to the user.

A fuller description of Integral data analysis appears in section 2 of this manual and a walkthrough example is given in 4.6.

Current Exclusions

The v11 commands and features not provided in v12 are:

Feature	Rationale for exclusion
recornorm	With version 12.5.0, this has been replaced and improved upon by the recorn mixing model.
thleqw	Rarely used command not yet implemented.
extend	Beginning with version 12.3.0, this has been replaced by the more flexible energies command.
background models	This has been replaced by v12's multiple source modeling techniques

Additionally, we have withdrawn seldom-used fitting methods anneal and genetic. Future development will add new techniques.