

- **energies: specify new energy binning for model fluxes**

Supply an energy-binning array to be used in model evaluations in place of their associated response energies. The calculated model spectra are then interpolated onto the response energy arrays before multiplying by the response matrix. This command replaces and enhances the **extend** command from earlier versions.

Syntax: **energies** <range specifier> [<additional range specifiers>...]
 energies <input ascii file>
 energies extend <extension specifier>
 energies reset

where the first <range specifier> ::= <low E> <high E> <nBins> log | lin

<additional range specifiers> ::= <high E> <nBins> log | lin

<extension specifier> ::= low | high <energy> <nBins> log | lin

All energies are in keV. Multiple ranges may be specified to allow for varied binning in different segments of the array, but note that no gaps are allowed in the overall array. Therefore only the first range specifier accepts a <low E> parameter. Additional ranges will automatically begin at the <high E> value of the previous range.

The extend option provides the same behavior as the old **extend** command. Models will use associated response energy arrays, with an additional low and/or high array extension. <energy> is the value to which the array is extended, using <nBins> additional log or linear bins.

With the <input ascii file> option, the user can instead supply a customized energy array from a text file. The format requirements are simply that the bin values must appear 1 to a line and in ascending sorted order. Blank lines are allowed and so are comments, which must be preceded by a '#'. A simple example:

```
# myEnergyBinning.txt
.1
1.0
10.  # now some linear bins
15.
20.
25.
```

which would actually produce the same energy array as:

```
energies .1 10. 2 log 25. 3 lin
```

Once an energy array is specified, it will apply to all models, and will be used in place of any response energy array (from actual or dummy responses) for calculating and binning the model flux. It will also apply to any models that are created after it is specified. To turn off this behavior and return all models back to using their response energies, simply type “energies reset”.

Similarly, an array extension created by the “extend” option will continue to be applied to all models until it is either overwritten by another extension, replaced by a new **energies** array, or removed with the “reset” option. This allows both low and high extensions to exist together.

When a custom-energy binned model flux array needs to be multiplied by a response matrix, xspec will temporarily rebin the flux array to match up with the response energy binning. This is done by simply scaling the flux by the fractional overlap between the custom and response bins. If there is no overlap between the custom and response energies, then the response will be multiplied by zero.

The **energies** command saves the most recently entered range and extension specifiers to be used as default values the next time it is called. The initial default range specifier is 1 range with <low E> = .1, <high E> = 10., <nBins> = 1000, and `lin`. The initial default extension specifier is `high` with <energy> = 100., <nBins> = 200, and `log`.

Examples:

```
XSPEC12> energies ,50,,log
// Creates an array from .1 to 50. of 1000 logarithmic bins.
XSPEC12> energies ,,,,100. 5 lin
// Modifies previous array by adding 5 linear bins from 50. to 100.
XSPEC12> energies ,,,,200.
// The 2nd range is now 50. to 200. in 5 linear bins.
XSPEC12> energies 1.,,100
// Array is now just 1 range, 1. to 50. in 100 logarithmic bins.
XSPEC12> energies myFile.txt
// Array is replaced with values stored in myFile.txt
XSPEC12> energies extend ,75.,,lin
// Models will go back to using response energies, but with an
// extension of the high end to 75. keV in 100 additional linear bins.
XSPEC12> energies extend low .01
// Add a low-end extension to .01 keV with 100 new linear bins.
XSPEC12> energies reset
// All models will go back to using the original energy arrays
// from responses.
```