



AHELP for CIAO 3.4

modelexpr

Context: [sherpa](#)

Jump to: [Description](#) [Examples](#) [Bugs](#) [See Also](#)

Synopsis

Model components can be used to create model expressions, in conjunction with the SOURCE and BACKGROUND commands (which then are used to fit the data), or with model stacks:

Syntax

```
sherpa> {SOURCE | BACKGROUND | <model_stackname>} = <modelExpr>
```

Description

Where the model expression, <modelExpr>, may be composed of one or more (algebraically-combined) of the following elements:

Model Expression Element:	Description:
<sherpa_modelname>	A Sherpa default model name
<sherpa_modelname>[<modelname>]	A name that is being given to a Sherpa model component (<modelname> must be enclosed in brackets, [])
<modelname>	A name that has been given to a Sherpa model component
<nested_model>	A nested model
<jointmode_model>	A joint-mode model

A model expression may be an algebraic combination of these elements that may include numerical values and the following operators: + - * / () < > { }.

By default, if a model expression includes a model component that has not previously been established, Sherpa will prompt the user for the initial model parameter values for that model component. Parameter prompting can be turned off using the PARAMPROMPT OFF command.

Example 1

Create a simple source model expression containing one model component:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> POLYNOM1D[modela]
sherpa> SOURCE 1 = modela
```

The last command in this series assigns a model expression to the source model for dataset number 1. This model expression contains one model component, modela (which is an arbitrary name that the user has given to the Sherpa model component POLYNOM1D).

Example 2

Create a source model expression containing multiple model components:

```
sherpa> ATTEN[modelm]
sherpa> BBODY[modelh]
sherpa> SOURCE 1 = modelm * modelh
```

This example establishes two model components, modelm and modelh (which are arbitrary names that the user has given to the Sherpa model components ATTEN and BBODY, respectively). The last command assigns the model expression modelm * modelh to the source model for dataset number 1. The following commands each assign various other model expressions to source models for other datasets:

```
sherpa> SOURCE 2 = 10*(modelm + modelh)
sherpa> SOURCE 3 = (modelm - modelh)/2
sherpa> SOURCE 4 = 0.5*modelm + 0.7*modelh
```

Example 3

Create a source model expression and establish model components simultaneously:

```
sherpa> ERASE ALL
sherpa> SOURCE 1 = ATTEN[modelm] * BBODY[modelh]
```

In this example the user assigns the model expression ATTEN[modelm] * BBODY[modelh] to the source model for dataset number 1.

Example 4

Create a background model expression:

```
sherpa> ERASE ALL
sherpa> PARAMPROMPT ON
Model parameter prompting is on
sherpa> BACKGROUND = (POW[modelc])/2
modelc.gamma parameter value [0]
modelc.ref parameter value [1]
modelc.ampl parameter value [1]
```

This command assigns the model expression (POW[modelc])/2, to the background model for dataset number 1. In this example, the user accepted the initial guessed values for all of the parameters using the <RETURN> key.

Example 5

Create a source model expression that includes a nested model:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> shape = GAUSS
sherpa> SIN[modeli]
sherpa> COS[modelj]
sherpa> SRC = modeli{shape} + modelj
```

The second command, `shape = GAUSS`, creates a model stack. The commands `SIN[modeli]` and `COS[modelj]` establish and assign names to the Sherpa model components `SIN` and `COS`, respectively. The final command assigns the model expression `modeli{shape} + modelj`, to the source model for dataset number 1. Note that `modeli` will be evaluated on the output of the model expression `shape`, i.e., $SRC = \text{SIN}(\text{GAUSS}(x)) + \text{COS}(x)$.

Example 6

Create a source model expression involving joint mode 2–D models:

```
sherpa> DATA example_img.fits FITSIMAGE
sherpa> POLYNOM1D[modelaForAxis1]
sherpa> GAUSS[modelbForAxis2]
sherpa> SRC = modelaForAxis1{x1}*modelbForAxis2{x2}
```

The command `POLYNOM1D[modelaForAxis1]` assigns the name `modelaForAxis1` to the Sherpa model component `POLYNOM1D`; likewise the command `GAUSS[modelbForAxis2]` assigns the name `modelbForAxis2` to the Sherpa model component `GAUSS`. The final command assigns the model expression `modelaForAxis1{x1}*modelbForAxis2{x2}`, to the source model for dataset number 1. This model expression is an algebraic combination of joint–mode models.

Bugs

See the [Sherpa bug pages](#) online for an up–to–date listing of known bugs.

See Also

sherpa

[autoest](#), [background](#), [create](#), [create model](#), [createparamset](#), [fit](#), [freeze](#), [get defined models](#), [get model params](#), [get models](#), [get num par](#), [get par](#), [get stackexpr](#), [getx](#), [gety](#), [guess](#), [instrument](#), [integrate](#), [is paramset](#), [jointmode](#), [kernel](#), [lineid](#), [linkparam](#), [mdl](#), [modelstack](#), [nestedmodel](#), [noise](#), [paramprompt](#), [paramset](#), [pileup](#), [rename](#), [run fit](#), [set par](#), [set paramset](#), [set stackexpr](#), [source](#), [thaw](#), [truncate](#), [unlink](#)

