

# The source model definition tool modeldef a.k.a. U7

D. Petry (GSFC/UMBC)

For the impatient:

- Look at the example file `example_inputmodel-1.txt` to see the definition of the input file syntax
- Write your `inputfile.txt`
- Run `modeldef < inputfile.txt`
- Use the resulting `_model.xml` and `_roi.xml` files with the Likelihood tool.

## Purpose

modeldef serves to simplify the generation of XML input files describing the source model and the region of interest (ROI) for the Likelihood tool.

## Usage

The executable modeldef.exe reads from stdin and writes to two files named X\_model.xml and X\_roi.xml

where X is the name given to the source model in the user input.

User can type his/her input directly at the prompt or prepare the input in an ASCII file using an editor and then pipe this file into modeldef.exe

```
modeldef.exe < inputfile.txt
```

If you don't need a function library, give the name none.

```

# The region of interest definition:
# -----
# (comments start with # followed by blank)
# Example for an input model
# $Id: example_inputmodel-1.txt,v 1.3 2003/12/03 21:57:56 petry Exp $
source distribution. Here the file example_inputmodel-1.txt is rendered:
The format of inputfile.txt is described in the examples in directory Data of the
roi example1 11 04 27.3139 +38 12 31.799 4.0 A1_Functions.xml
roi example1 11 04 27.3139 +38 12 31.799 4.0 A1_Functions.xml

```

---

```

# -----
# Source definitions follow the pattern
# source-type name parameters
# where parameters depend on the source-type.
# Permitted source types are POINT, DIFFUSE
#
# For source-type "point" the parameters are
# ra dec spectrum
#
# For source-type "diffuse" the parameters are
# mapfilname spectrum
# If mapfilname = "const", the source is assumed to be isotropic.
#
# In all cases the spectrum parameters are of the form
# spectral-type p1 p2 ...
# where the pi are the parameters defining the spectrum.

```

(example\_inputmodel-1.txt continued)

```
(example_inputmodel-1.txt continued)
# Permitted generic spectral types are POWERLAW, BROKENPOWER, ABSEGE, GAUSSIAN.
#
# The parameters of these generic types are as follows (x denotes energy/MeV)
# POWERLAW prefactor index scale
# differential flux (cm^-2s^-1MeV^-1) = prefactor * pow(x/scale, index)
#
# BROKENPOWER prefactor index1 index2 break
# diff. flux (cm^-2s^-1MeV^-1) = prefactor * pow(x/break, index1) for x<break
# prefactor * pow(x/break, index2) for x>=break
#
# ABSEGE tau0 e0 index
# diff. flux (cm^-2s^-1MeV^-1) = 1
# for x<e0
# = exp(-tau0 * pow(x/e0, index))
# for x>=e0
#
# GAUSSIAN prefactor mean sigma
# diff. flux (cm^-2s^-1MeV^-1) = prefactor/(sqrt(2.*PI)*sigma)
# * exp(-pow(x-mean)/sigma, 2)/2.)
```

```

# a point source with a generic powerlaw spectrum
point MknABC 11 01 27.3139 +38 12 31.799 powerlaw 1e-11 2 1

# a point source with a generic broken powerlaw spectrum
point MknDEF 11 02 27.3139 +38 12 31.799 brokenpower 1e-11 2.3 2.6 1

# a non-isotropic diffuse source with a broken powerlaw spectrum
diffuse backgd gas.cel brokenpower 1e-11 2.6 2.7 0.5

# an isotropic diffuse source with a powerlaw spectrum
diffuse mystery const powerlaw 1e-11 2 1

# a point source with a spectrum of a type given in the function library
point Mkn421 11 04 27.3139 +38 12 31.799 type "Bright Source Power-law"

# a non-isotropic diffuse source with a type given in the function library
diffuse backgd2 gas.cel type "Galactic Diffuse Power-law"

```

Piping `example-inputmodel-1.txt` into `modeldef.exe` will result in the generation of the files `example1_model.xml` and `example1_roi.xml`. Refer to the Likelihood package documentation for the details of how to use these files.

## The XML format (briefly)

The contents of the file `example1_roi.xml` generated from the input discussed above

```
<Region-of-Interest title="example1">
  <timeInterval stop="1e+12" unit="seconds" start="0" />
  <energies emax="316230" emin="30" unit="MeV" />
  <acceptanceCone ra="166.114" dec="38.2088" radius="4" coordsys="Equatorial" />
</Region-of-Interest>
```



The contents of the file `example1_model.xml` generated from the input discussed above:

```
<source_library title="example1" function_library="A1_Functions.xml">
  <source name="MknABC" type="PointSource">
    <spectrum type="PowerLaw">
      <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="Prefactor" scale="1" value="1e-11" />
      <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="Index" scale="1" value="2" />
      <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="Scale" scale="1" value="1" />
    </spectrum>
    <spatialModel type="SkyDirFunction">
      <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="RA" scale="1" value="165.364" />
      <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="DEC" scale="1" value="38.2088" />
    </spatialModel>
  </source>
  ...
</source_library>
```

(Source MknABC was not using the function library.)

The contents of the file `example1_model.xml` (continued)

```

...
<source name="background" type="DiffuseSource">
  <spectrum type="BrokenPowerLaw">
    <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="Prefactor" scale="1" value="1e-11" />
    <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="Index1" scale="1" value="2.6" />
    <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="Index2" scale="1" value="2.7" />
    <parameter max="3.40282e+38" min="-3.40282e+38" free="1" name="BreakValue" scale="1" value="0.5" />
  </spectrum>
  <spatialModel file="gas.cel" type="SpatialMap">
    <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="Prefactor" scale="1" value="1" />
  </spatialModel>
</source>
...

```

(Source `background` was not using the function `library`.)

(Source Mkn 421 was using the spectral prototype “Bright Source Power-law”.)

```

...
<source name="Mkn421" type="PointSource">
  <spectrum type="PowerLaw">
    <parameter max="1000" min="0.001" free="1" name="Prefactor" scale="1e-09" value="10" />
    <parameter max="-1" min="-3.5" free="1" name="Index" scale="1" value="-2" />
    <parameter max="200" min="50" free="0" name="Scale" scale="1" value="100" />
  </spectrum>
  <spatialModel type="SkyDirFunction">
    <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="RA" scale="1" value="166.114" />
    <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="DEC" scale="1" value="38.2088" />
  </spatialModel>
</source>
...

```

The contents of the file `example1_model.xml` (continued)

The contents of the file `example1_model.xml` (continued)

```

...
<source name="background2" type="DiffuseSource">
  <spectrum type="PowerLaw">
    <parameter max="1000" min="0.001" free="0" name="Prefactor" scale="0.001" value="6.31" />
    <parameter max="-1" min="-3.5" free="0" name="Index" scale="1" value="-2.1" />
    <parameter max="200" min="50" free="0" name="Scale" scale="1" value="100" />
  </spectrum>
  <spatialModel file="gas.cel" type="SpatialMap">
    <parameter max="3.40282e+38" min="-3.40282e+38" free="0" name="Prefactor" scale="1" value="1" />
  </spatialModel>
</source>
</source_library>
...

```

(Source `background2` was using the spectral prototype "Galactic Diffuse Power-law".)

# The function library

A function library has the format

```
<function_library title="prototype Functions">
  <function name="Bright Source Power-law" type="PowerLaw">
    <!-- This prototype provides *very roughly* appropriate values for
    bright EGRET point sources. -->
    <parameter name="Prefactor"
    value="10" unit="cm^-2s^-1MeV^-1" min="1e-3" max="1e3"
    free="true" scale="1e-9"/>
    <parameter name="Index"
    value="-2" unit="none" min="-3.5" max="-1"
    free="true" scale="1"/>
    <parameter name="Scale"
    value="100" unit="none" min="50" max="200"
    free="false" scale="1"/>
  </function>
  ...
</function_library>
```

Have to be generated "by hand" (so far).

## Other features of the modeldef package

The example input model file `egret3rdcat_inputmodel.txt` in directory `Data` contains all the 256 sources from the 3rd EGRET catalog (except the solar flare) which have spectral information. By running

```
modeldef.exe > egret3rdcat_inputmodel.txt
```

one obtains `3rd_EGRET_Catalog_model.xml`.

Note that presently the import functionality does not support the import of models with diffuse sources (still being debugged).

```
point MySource 11 02 27.3139 +38 12 31.799 brokenpower 1e-11 2.3 2.6 1
```

```
import 3rd_EGRET_catalog_model.xml
```

```
roi example4 11 04 27.3139 +38 12 31.799 2.0 none
```

```
# Example for an input model using the "import" option
# $Id: example-inputmodel-4.txt,v 1.1 2003/12/04 21:05:36 petry Exp $
```

input model without copying all the lines:

Can use the import feature to add 3rd\_EGRET\_catalog\_model.xml to any

Note also that `modeldef` is ultimately meant to become a tool with a graphical user interface.

