

makesourcedb

A source table is used by BBS to know the available sources and their parameters. It is a special form of a Parameter Table and is described in detail in [parmdb.pdf](#).

The sources in a source table can optionally be grouped into so-called patches. BBS makes use of patches by, for instance, combining them when doing parameter fits. It is possible in makesourcedb to define patches and tell to which patch a source belongs. If no patch is defined for a source, the source is its own patch. Thus for each such source a patch is created, where it is possible to add a prefix and/or suffix to the patch name making it possible to use patch patterns in BBS.

The program makesourcedb can be used to fill or append a source table with (selective) sources read from an ASCII file. Note that the script [gsm.py](#) can be used to extract sources from the GSM and put them into a file that can directly be used as input for makesourcedb.

It can be run like:

```
makesourcedb key=value key=value ...
```

The possible keywords and values are described in the following table.

| Keyword | Type | Default | Description |
|-------------|--------|---|--|
| in | string | | Name of the input ASCII file |
| out | string | | Name of the source table |
| outtype | string | casa | The output type of the source table which can be casa or blob. Casa is the standard type which can be used for all purposes. Blob is a special type that can only be used for NDPPP demixing. Its advantage is that it is much faster for the large source models used in demixing. |
| format | string | Name,Type,Ra,Dec,I,Q,U,V,Splnx, MajorAxis,MinorAxis,Orientation | The format of the ASCII file. It is described in detail in the next section. A value "<filename" means that the format is read from the given file. If no file name is given, it is read from the input file. In a file the format string must be given in a single line. It can be specified in two forms; the first form is preferred. format = format-string #(format-string) = format |
| patchprefix | string | "" | prefix for the patch name implicitly created from a source. |
| patchsuffix | string | "" | suffix for the patch name implicitly created from a source. |
| append | bool | False | Append to the source table? If not, a new source table is created. |
| check | bool | False | Check for each entry if the patch or source name is duplicate? True makes the program somewhat slower. At the end it is always checked that no duplicate entries exist. |

| Keyword | Type | Default | Description |
|---------|-------|---------|--|
| center | angle | | If given, only sources in the given circle or box around the center are written into the source table. The center has to be given as “ra,dec” given in casacore's MVAngle format (discussed below). |
| radius | angle | | Radius of a circle (in MVAngle format) around the center to select sources. |
| width | angle | | Full width of a box around the center to select sources to be given as “width” (for a square box) or “rawidth,decwidth” (in MVAngle format). |

makesourcedb reads lines from the input file, interprets them using the format string (described below), and writes the ones matching the optionally given circle or box into the Source Table. Possible comment lines (indicated by #), empty lines, and format line in the input file are ignored when reading the values from the file. Whitespace (one or more blanks and/or tabs) is ignored, unless used as separator. Leading and trailing whitespace is always ignored.

Adding a source to the source table means two things:

- it is added to the source table itself
- for all source parameters appropriate entries are added to the associated parm table. The names of those parameters look like sourcename:fieldname where field name is the name of a field given in the format string.

Format string

The format string defines which fields are present in the ASCII file and which separators are used. For example:

```
Name, Ra, Dec
```

tells that the lines in the input file contain source name, ra, and dec. It also tells that the values on each line are separated by commas. A field name can consist of alphanumeric characters, underscores and colons. A colon cannot be the first character. All other characters (including whitespace) can be used as separator. A colon can also be used as field separator as long as it is surrounded by whitespace in the format string (to distinguish it from a colon in a field name).

A few special options are available in the format string.

- Default values are used for missing or empty field values in input lines. Standard default values (as listed in the table below) exist, but it is possible to specify an explicit default value in the format string. It can be done by giving a value (enclosed in quotes) for a field like:

```
Name, Ra, Dec, I='1.5'
```

- It is also possible to define a field with a fixed value that does not appear in the input lines like:

```
Name, Ra, Dec, Category=fixed'2'
```

In this way the Category parameter will be defined for all sources, while the input lines do not contain the Category field.

- If the input lines contain fields that have to be ignored, they can be skipped by giving no field name in the format line or by using the field name dummy. For example:

```
Name, Ra, ,Dec
```

means that the third field is skipped. Note that the name dummy has to be used if whitespace is used as separator like:

```
Name Ra dummy Dec
```

The format strings can be given in a few ways:

- Directly in the program format keyword.
- As a format line in the input file (before the first value line) or optionally in a separate text file. Such a format line has to start with `format=` (whitespace is allowed). For backward compatibility the format line can also be given as a special comment line like `#(...) =format`, where ... is the format string. Whitespace can be used.

Field types

makesourcedb distinguishes two types of fields: standard fields and other fields. The standard fields have a predefined (case-insensitive) name. However, the name is case-sensitive in the sense that the name given in the table below is used as the parameter name in the parm table. The following standard fields are known:

| Field | Type | Unit | Default | Description |
|-----------------------|---------|--------|---------|---|
| Name | string | | | Source Name |
| Type | string | | | Source type. Currently only point, gaussian, or shapelet is possible. |
| Ra | angle | | | Right ascension in MVAngle format (see below) |
| Dec | angle | | | Declination in MVAngle format (see below) |
| I | double | Jy | 1 | Stokes I flux (for Gaussian sources this the <i>total</i> flux) |
| Q | double | Jy | 0 | Stokes Q flux |
| U | double | Jy | 0 | Stokes U flux |
| V | double | Jy | 0 | Stokes V flux |
| MajorAxis | double | arcsec | 1 | FWHM of major axis for a Gaussian source |
| MinorAxis | double | arcsec | 1 | FWHM of minor axis for a Gaussian source |
| Orientation | double | deg | 1 | Orientation of the major axis in degrees North over East |
| OrientationIsAbsolute | boolean | | false | Whether the Orientation is absolute (w.r.t. declination axis) or relative (w.r.t. m axis in not-specified field). See the note below. |

| Field | Type | Unit | Default | Description |
|---------------------|---------|--------------------|--|---|
| IShapelet | string | | | Name of the ASCII file defining the shapelet coefficients and scale of the I flux in the 'Sarod' format. Note that field I gives the total flux, while IShapelet defines the shape. |
| QShapelet | string | | | Idem for Q flux. An empty name means that no Q flux is present. Note that the Q (and U) flux can still be given using RotationMeasure. |
| UShapelet | string | | | Idem for U flux. |
| VShapelet | string | | | Idem for V flux. |
| SpectralIndex | doubles | | [] | Spectral index. Multiple values can be given enclosed in square brackets and separated by commas. The n-th value defines the n-th order term. Note that the old way of specifying the spectral index (using SpectralIndexDegree and SpectralIndex:i) is not supported anymore. If used, an error is given. |
| LogarithmicSI | bool | true | Determines the meaning of the spectral index. See Logarithmic SI below | |
| ReferenceFrequency | double | Hz | | Reference frequency (in Hz) for the spectral index |
| RotationMeasure | double | rad/m ² | | Rotation measure for the polarized flux Q and U |
| PolarizationAngle | double | rad | 0 | Polarization angle (in radians) for the polarized flux |
| PolarizedFraction | double | | 0 | Fraction of I that is polarized |
| ReferenceWavelength | double | m | | Reference wavelength (in m) if RM is specified using Q and U |
| Category | integer | | 2 | Source category (must be 1, 2, or 3) |
| Patch | string | | | Patch name |
| rah | double | hr | 0 | Special way to specify RA (see below) |
| rad | double | deg | 0 | |
| ram | double | min | 0 | |
| ras | double | sec | 0 | |
| dech | double | hr | 0 | Special way to specify DEC (see below) |
| decd | double | deg | 0 | |
| decm | double | min | 0 | |
| decs | double | sec | 0 | |

All non-standard fields have type 'double' and a default of 0. The name as given in the format string is used as the parameter name in the parm table.

Note that the polarized flux can be given directly (as Q and U) or by means of the rotation measure,

polarization angle, and polarized fraction. An error is given if both are given for the same source.

Logarithmic Spectral Index

If `LogarithmicSI` is `true` (the default), the source intensity '*I*' is computed as

$$I = I_0 * (v/v_0)^{c_0 + c_1 * \log_{10}(v/v_0) + c_2 * \log_{10}(v/v_0)^2 + \dots}$$
, where '*v*' is the frequency and '*v*₀' is the reference frequency.

If `LogarithmicSI` is `false`, the spectral index is interpreted in a way compatible with `wsclean`, namely as

$$I = I_0 * (c_0 * (v/v_0) + c_1 * (v/v_0)^2 + c_2 * (v/v_0)^3 + \dots)$$

Using Rotation Measure

The flux of a source can be defined using a rotation measure by means of the field name `RotationMeasure`. The required polarization info can be given in two ways:

- Directly by means of the fields `PolarizationAngle` and `PolarizedFraction`. They define the angle and fraction for wavelength 0. If the field `ReferenceWavelength` is also given, it must be 0.
- Indirectly by means of the `Q` and `U` flux. `ReferenceWavelength` should be given as well to define the wavelength they are valid for. If not given, it will be derived from the spectral index `ReferenceFrequency`. The polarization angle and fraction will be calculated taking the spectral index into account like:

$$\text{polang} = \text{atan2}(U, Q) / 2 - \text{refwvl}^2 * \text{rm}$$

$$\text{polfrac} = \sqrt{Q^2 + U^2} / I(\text{refwvl})$$
`ReferenceWavelength` must be > 0 if a spectral index is given.

Defining Patches

A patch is defined when an input line contains a patch name and no source name. For a patch the fields `Category`, `Ra`, `Dec`, and `I` are also used which define its category, center, and total flux. In a future version of `makesourcedb` the center and total flux can be calculated from the sources that are part of the patch.

A source is attached to a patch if its input line contains a patch name. A line defining that patch should have been processed before this line.

If no patch name is given for a source (as is usually the case), the source forms a patch in itself and the source name is used as the patch name (with an optional prefix/suffix). In this way BBS can always deal with patches.

Orientation of Gaussian sources

Until 2022, the orientation of Gaussian sources has been interpreted somewhat wrong. It was

interpreted (both by DP3 and pybdsf) as the angle with respect to the m axis of a certain image. The correct (implicit) definition is the orientation w.r.t. the declination axis. For fields far away from the NCP, and/or sources not too far from the field center, these two definitions are almost identical.

In DP3 v5.3 a fix was implemented to take the projection effects into account properly. This fix is only called when OrientationIsAbsolute is true. A script will be added to LSMTool to convert from 'relative' orientation to absolute ones (this script needs the phase center of the original image as an input).

Angle specification

Angles can be given in a variety of ways using the MVAngle format available in casacore.

- In sexagesimal HMS format like 12:34:56.78 or 12h34m56.78. Note that a colon defines hours. To avoid mistakes, it is not allowed to use colons in declination values (but see below).
- In sexagesimal DMS format like 12.34.56.78 or 12d34m56.78.
- As a double value optionally followed by a unit like 12.34deg or 12.34rad. If no unit is given, radians are used.

In all cases negative values are possible. It is checked if RA is between -360 and +360 degrees and DEC between -90 and +90 degrees.

Some catalogues use a different way to specify Ra or Dec. For this reason the special field names rah, ram, and ras can be used in the format string. They define the hour, minute, and second part of an RA. If the Ra is in degrees, rad should be used instead of rah. Similarly dech, decd, decm, and decs can be used for declination. A negative hour or degree part means that the entire value is negative. For example:

```
rah ram ras
```

can be used in the format string to interpret RA values like:

```
12 34 56.78  
-12 34 56.78
```

Note that these special field names can also be used to interpret declination values given in degrees using colons. For instance:

```
decd : decm : decs
```

to interpret

```
12:34:56.78
```

mergesourcedb

This program makes it possible to merge two SourceDB data sets or to copy (part of) a SourceDB.

Patches for which the positions in both SourceDBs match (within a given uncertainty radius) are merged. It means that the patch in the second SourceDB will be used!! The mode parameter tells if non-matching patches in the second SourceDB should also be copied.

It is possible to select patches using a name pattern.

It can be run like:

```
mergesourcedb key=value key=value ...
```

The possible keywords and values are described in the following table.

| Keyword | Type | Default | Description |
|-------------|--------|----------|---|
| in1 | string | | Name of the first SourceDB; this must be given |
| in2 | string | "" | Name of the second SourceDB. If empty string, no merge will be done, but the patches of in1 are copied. |
| out | string | | Name of the output SourceDB |
| outtype | string | casa | The output type of the source table which can be casa or blob. Casa is the standard type which can be used for all purposes. Blob is a special type that can only be used for NDPPP demixing. Its advantage is that it is much faster for the large source models used in demixing. |
| append | bool | False | Append to the source table? If not, a new source table is created. |
| in1.patches | string | * | Pattern telling the names of the patches to take from in1 |
| in2.patches | string | * | Pattern telling the names of the patches to take from in2 |
| mode | string | all | match = only use in2 patches matching an in1 patch position and replace that in1 patch all = also copy the other in2 patches matching the pattern |
| radius | string | 10arcsec | Uncertainty radius for matching patches in both SourceDBs |

showsourcedb

This program can be used to show the contents of a SourceDB data set.

It can be run like: It can be run like:

```
showsourcedb key=value key=value ...
```

The possible keywords and values are described in the following table.

| Keyword | Type | Default | Description |
|---------|--------|---------|---|
| in | string | | Name of the SourceDB; this must be given |
| mode | string | all | patch = show all patches source = show all sources all = show patches and sources skymodel = show all in skymodel format |
| patches | string | * | Pattern telling the names of the patches to take |

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:user_software:documentation:makesourcedb&rev=1666347964

Last update: **2022-10-21 10:26**

