

**NAME**

`gxl2dot,dot2gxl` – GXL-DOT converters

**SYNOPSIS**

`gxl2dot` [ `-gd?` ] [ `-outfile` ] [ *files* ]

`dot2gxl` [ `-gd?` ] [ `-outfile` ] [ *files* ]

**DESCRIPTION**

`gxl2dot` converts between graphs represented in GXL and in the DOT language. Unless a conversion type is specified using a flag, `gxl2dot` will deduce the type of conversion from the suffix of the input file, a ".dot" suffix causing a conversion from DOT to GXL, and a ".gxl" suffix causing a conversion from GXL to DOT. If no suffix is available, e.g. when the input is from a pipe, and no flags are used then `gxl2dot` assumes the type of the input file from its executable name so that `gxl2dot` converts from GXL to DOT, and `dot2gxl` converts from DOT to GXL.

GXL supports a much richer graph model than DOT. `gxl2dot` will attempt to map GXL constructs into the analogous DOT construct when this is possible. If not, the GXL information is stored as an attribute. The intention is that applying `gxl2dot|dot2gxl` is semantically equivalent to the identity operator.

**OPTIONS**

The following options are supported:

`-g` The command name and input file extensions are ignored, the input is taken as a DOT file and a GXL file is generated.

`-d` The command name and input file extensions are ignored, the input is taken as a GXL file and a DOT file is generated.

`-?` Prints usage information and exits.

`-o` *outfile*

If specified, the output will be written into the file *outfile*. Otherwise, output is written to standard out.

**OPERANDS**

The following operand is supported:

*files* Names of files containing 1 or more graphs in GXL or DOT. If no *files* operand is specified, the standard input will be used.

**RETURN CODES**

Both `gxl2dot` and `dot2gxl` return **0** if there were no problems during conversion; and non-zero if any error occurred.

**BUGS**

`gxl2dot` will only convert in one direction even if given multiple files with varying suffixes.

The conversion can only handle one graph per GXL file.

There are some GXL constructs which `gxl2dot` cannot handle.

**AUTHORS**

Krishnam Pericherla <kp@research.att.com>

Emden R. Gansner <erg@research.att.com>

**SEE ALSO**

`dot(1)`, `libgraph(3)`, `libagraph(3)`, `neato(1)`, `twopi(1)`