NAME

libgvc – Graphviz context library

SYNOPSIS

```
#include <graphviz/gvc.h>

/* set up a graphviz context */
extern GVC_t *gvNEWcontext(const lt_symlist_t *builtins, int demand_loading);

/*  set up a graphviz context - alternative */
/*     (wraps the above two functions using info built into libgvc) */
extern GVC_t *gvContext(void);

/* parse command line args - minimally argv[0] sets layout engine */
extern int gvParseArgs(GVC_t *gvc, int argc, char **argv);
extern graph_t *gvNextInputGraph(GVC_t *gvc);

/* Compute a layout using a specified engine */
extern int gvLayout(GVC_t *gvc, graph_t *g, char *engine);

/* Compute a layout using layout engine from command line args */
extern int gvLayoutJobs(GVC_t *gvc, graph_t *g);

/* Check if a layout has been done */
extern bool gvLayoutDone(graph_t *g);

/* Render layout into string attributes of the graph */
extern void attach_attrs(graph_t *g);

/* Parse an html string */
extern char *agstrdup_html(char *s);
extern int aghtmlstr(char *s);

/* Render layout in a specified format to an open FILE */
extern int gvRender(GVC_t *gvc, graph_t *g, char *format, FILE *out);

/* Render layout in a specified format to an open FILE */
extern int gvRenderFilename(GVC_t *gvc, graph_t *g, char *format, char *filename);

/* Render layout according to -T and -o options found by gvParseArgs */
extern int gvRenderJobs(GVC_t *gvc, graph_t *g);

/* Clean up layout data structures - layouts are not nestable (yet) */
extern int gvFreeLayout(GVC_t *gvc, graph_t *g);

/* Clean up graphviz context */
extern int gvFreeContext(GVC_t *gvc);

/* Inquire about available plugins */
/* See comment in gvc.h          */
extern char** gvPluginList(GVC_t *gvc, char* kind, int* cnt, char*);
```

DESCRIPTION

libgvc provides a context for applications wishing to manipulate and render graphs. It provides command line parsing, common rendering code, and a plugin mechanism for renderers.

SEE ALSO

dot(1), **neato**(1), **cdt**(3), **cgraph**(3)

AUTHOR

John Ellson (ellson@research.att.com), AT&T