```
/**
 * @file libpaps.h
 *
 * @brief Interface for converting pango layouts into postscript.
 *
 * The library is used by first creating a paps object through
 * paps_new(). The paps object is then used to generate PostScript
 * for one or more pango layouts through paps_layout_to_postscript_strdup()
 * or paps_layout_line_to_postscript_strdup(). Once all the layouts
 * have been processed, the system may generate the PostScript header through
 * paps_get_postscript_header_strdup(). It
 * is the responsibility of the caller to place the PostScript
 * header in front of the renderings of the pango layouts.
 *
 * The PostScript of the rendered layouts is relocatable in all
 * sense so any postscript operators may be used to change their
 * colors, transformations etc.
 */
#ifndef LIBPAPS_H
#define LIBPAPS_H

#include <glib.h>
#include <pango/pango.h>

typedef void paps_t;

/**
 * Create a new paps object that may subsequently be used for
 * generating postscript strings.
 *
 * @return
 */
paps_t *paps_new();

/**
 * Delete the paps object and free any allocated memory associated with
 * it.
```

```
 *
 * @param paps
 */
void paps_free(paps_t *paps);

/**
 * Set the scales for characters.
 *
 * @param paps Paps object
 * @param scale_x x-coordinate scale
 * @param scale_y y-coordinate scale
 *
 */
void
paps_set_scale(paps_t  *paps,
               gdouble  scale_x,
               gdouble  scale_y);

/**
 * libpaps may currently be used only with a PangoContext that it
 * is creating. The context returned may of course be changed though
 * by any routines related to a PangoContext.
 *
 * @return
 */
PangoContext *paps_get_pango_context();

/**
 * paps_get_postscript_header_strdup() returns the header and the
 * font definitions related to the glyphs being used. This routine
 * must only be called after the processing of all the strings is
 * over.
 *
 * @param paps Paps object
 *
 * @return A newly allocated string containing the postscript prologue
 *         needed for the postscript strings created by libpaps.
```

```
 */
gchar *paps_get_postscript_header_strdup(paps_t *paps);

/**
 * Create postscript related to a PangoLayout at position pos_x,
 * pos_y (postscript coordinates). The related font definitions
 * are stored internally and will be returned when doing
 * paps_get_postscript_header_strdup.
 *
 * @param paps Paps object
 * @param pos_x x-position
 * @param pos_y y-position
 * @param layout Layout to render
 *
 * @return
 */
gchar *paps_layout_to_postscript_strdup(paps_t *paps,
                                        double pos_x,
                                        double pos_y,
                                        PangoLayout *layout);
/**
 * Create postscript related to a single PangoLayout line at position
 * pos_x, pos_y (postscript coordinates). The related font definitions
 * are stored internally and will be returned when doing
 * paps_get_postscript_header_strdup.
 *
 * @param paps Paps object
 * @param pos_x x-position
 * @param pos_y y-position
 * @param layout_line Layout line to render
 *
 * @return
 */
gchar *paps_layout_line_to_postscript_strdup(paps_t *paps_,
                                             double pos_x,
                                             double pos_y,
                                             PangoLayoutLine *layout_line);
```

```
/**
 * Utility function that translates from postscript points to pango
 * units.
 *
 * @param points
 *
 * @return
 */
double paps_postscript_points_to_pango(double points);

#endif /* LIBPAPS */
```