Graphics Workstation Research Group

Technical Report #4

University of Illinois PHIGS PLUS C Language Binding

Parris K. Egbert

Department of Computer Science

University of Illinois at Urbana-Champaign

Urbana, Illinois 61801

April 1990

1. PHIGS PLUS Data Structures

Parea_data -- area flags, facets, etc

```
typedef struct {
        Pint            fflag;          /* facet flag                   */
        Pint            vflag;          /* vertex flag                  */
        Pint            colr_type;      /* color type                   */
        Popt_data       *fdata          /* facet data                   */
} Parea_data;
```

Parea_prop -- area properties

```
typedef struct {
        Pfloat          amb;            /* ambient reflection coefficient   */
        Pfloat          dif;            /* diffuse reflection coefficient   */
        Pfloat          spec;           /* specular reflection coefficient  */
        Pgcolr          colr;           /* specular color               */
        Pfloat          exp;            /* specular exponent            */
        Pfloat          trans;          /* transparency coefficient */
} Parea_prop;
```

Parea_set_data -- area set flags, facets, edges, etc

```
typedef struct {
        Pint            fflag;          /* facet flag                   */
        Pint            eflag;          /* edge flag                    */
        Pint            vflag;          /* vertex flag                  */
        Pint            colr_type;      /* color type                   */
        Popt_data       *fdata;         /* facet data                   */
} Parea_set_data;
```

Paspect -- aspect

```
typedef enum {
        PLINETYPE,
        PLINEWIDTH_SCALE_FACTOR,
        PPOLYLINE_COLOR_INDEX,
        PMARKERTYPE,
        PMARKERSIZE_SCALE_FACTOR,
        PPOLYMARKER_COLOR_INDEX,
        PTEXT_FONT,
```

```
        PTEXT_PRECISION,
        PCHARACTER_EXPANSION_FACTOR,
        PCHARACTER_SPACING,
        PTEXT_COLOR_INDEX,
        PINTERIOR_STYLE,
        PINTERIOR_STYLE_INDEX,
        PINTERIOR_COLOR_INDEX,
        PEDGE_FLAG,
        PEDGETYPE,
        PEDGEWIDTH_SCALE_FACTOR,
        PEDGE_COLOR_INDEX,

        /* now start PHIGS PLUS attribute ids */

        PBACK_INTERIOR_COLOR,
        PBACK_INTERIOR_STYLE,
        PBACK_INTERIOR_STYLE_INDEX,
        PBACK_INTERIOR_REFLECTANCE_EQUATION,
        PBACK_INTERIOR_SHADING_METHOD,
        PBACK_AREA_PROPERTIES,
        PPOLYLINE_SHADING_METHOD,
        PINTERIOR_REFLECTANCE_EQUATION,
        PINTERIOR_SHADING_METHOD,
        PAREA_PROPERTIES,
        PCURVE_APPROXIMATION_CRITERIA,
        PSURFACE_APPROXIMATION_CRITERIA,
        PTRIMMING_CURVE_APPROXIMATION_CRITERIA
} Paspect;
```

Pboolean -- TRUE or FALSE enumerated type

```
typedef enum {
        PFALSE,
        PTRUE
} Pboolean;
```

Pcolr_map_bundl -- color mapping bundle

```
typedef struct {
        Pint            method; /* color mapping method   */
        Pcolr_map_rec   *rec;           /* data record                  */
} Pcolr_map_bundl;
```

Pcolr_map_fac -- color mapping facilities

```
typedef struct {
        Pint_list methods;          /* list of color mapping methods */
        Pint            predefined;     /* # of pred. color map indices     */
} Pcolr_map_fac;
```

Pcolr_map_method_rec -- color mapping data record

```
typedef union {
        struct {
                Pint num_colrs;         /* number of colors for true map */
        } true_map;
        struct {
                Pint num_colrs;         /* number of avail pseudo colors */
        } pseudo;
        struct {
                Pchar *dummy;           /* no record from pseudo_n       */
        } pseudo_n
} Pcolr_map_method_rec;
```

Pcolr_map_rec -- color mapping data record

```
typedef union {
        struct {
                Pchar *data;            /* no data for true mapping      */
        } true_map;
        struct {
                Pint       model;       /* color model                   */
                Pcolr_bundl weight;     /* weight vector                 */
                Pint       num_colrs;   /* number of color values  */
                Pcolrv     *colrs;      /* list of colors                */
        } pseudo;
        struct {
                Pint       model;       /* color model                   */
                Pint       num_lists;   /* number of lists          */
                Pfloat_list *coords;    /* color coordinates             */
        } pseudo_n
} Pcolr_map_rec;
```

Pcolrv -- color value

```
typedef union {
        Pint            ival;           /* integer index                */
        Ppoint3         fval;           /* 3 floating point values   */
} Pcolrv;
```

Pcull_mode -- face culling mode

```
typedef enum {
        PNONE,
        PBACKFACING,
        PFRONTFACING
} Pcull_mode;
```

Pcurve_point -- 2 or 3d point used for curve control points

```
typedef union {
        Ppoint          point2d; /* 2d point                    */
        Ppoint3         point3d; /* 3d point                    */
} Pcurve_point;
```

Pcurve_surf_fac -- curve and surface facilities

```
typedef struct {
        Pint            max_nurb;       /* max nurb order supported          */
        Pint            max_trim;       /* max trim curve order supported   */
        Pint_list surfaces;/* list of para surface types supp.     */
        Pint_list curve_approx;     /* list of curve approx types supp.   */
        Pint_list surface_approx;   /* list of surface approx types supp.*/
        Pint_list trim_approx;      /* list of trim crv approx types supp.       */
} Pcurve_surf_fac;
```

Pcurve_surf_list -- curve and surface attribute list

```
typedef struct {
        Pint            surfaces;/* parametric surfaces                */
        Pint            curve_approx;   /* curve approximation criteria          */
        Pint            surface_approx; /* surface approximation criteria        */
        Pint            trim_approx;    /* trimming curve approx crit            */
} Pcurve_surf_list;
```

Pcurve_type -- curve type

```
typedef Psurf_type          Pcurve_type;
```

Pdepth_mode -- depth cue modes

```
typedef enum {
        PSUPPRESSED,
        PALLOWED
} Pdepth_mode;
```

Pdepth_bundl -- depth cue bundle

```
typedef struct {
        Pdepth_mode     mode;           /* depth cue mode                */
        Pfloat          front_plane;    /* front depth cue ref plane     */
        Pfloat          back_plane;     /* back depth cue ref plane      */
        Pfloat          front_scale;    /* front depth cue scaling   */
        Pfloat          back_scale;     /* back depth cue scaling    */
        Pgcolr          colr;           /* depth cue color               */
} Pdepth_bundl;
```

Pdist_mode -- face distinguishing mode

```
typedef enum {
        PNO,
        PYES
} Pdist_mode;
```

Pedge_bundl_plus -- edge bundle

```
typedef struct {
        Pedge_flag      edg_flag;       /* edge flag                     */
        Pint            type;           /* edgetype                      */
        Pfloat          scale;          /* edgewidth scale factor        */
        Pgcolr          colr;           /* edge color index              */
```

} **Pedge_bundl_plus**;

Pedge_flag_list -- list of edge flags

```
typedef struct {
        Pint            num_edges;      /* number of edge flags         */
        Pedge_flag      *edges;         /* list of edge flags           */
} Pedge_flag_list;
```

Pedge_flag_list_list -- list of list of edge flags

```
typedef struct {
        Pint            num_lists;      /* number of edge flag lists    */
        Pedge_flag      *list;          /* list of edge flag lists      */
} Pedge_flag_list_list;
```

Pedge_list -- list of edge flags

```
typedef struct {
        Pint            num;            /* number of edges              */
        Pedge_flag      *edge;          /* list of edge flags           */
} Pedge_list;
```

Pedge_list_list -- list of list of edge

```
typedef struct {
        Pint            num;            /* number of lists      */
        Pedge_list      *list;          /* list of lists                */
} Pedge_list_list;
```

Pel_data -- element data

```
typedef union {
        Pint            idata;          /* integer valued data          */
        Pfloat          fdata;          /* float valued data            */
        Ppoint_list3    pts3;           /* list of 3d points            */
```

```
Ppoint_list         pts;            /* list of 2d points                */
struct {
        Ppoint3         pos;            /* text point               */
        Pvector3dir[2];         /* direction vectors        */
        Pchar           *str;           /* text string              */
} tx3;
struct {
        Ppoint          pos;            /* text point               */
        Pchar           *str;           /* text string              */
} tx;
struct {
        Ppoint3         ref_pt;         /* reference point*/
        Ppoint3         offset;         /* annotation offset        */
        Pchar           *str;           /* text string              */
} atx3;
struct {
        Ppoint          ref_pt;         /* reference point*/
        Ppoint          offset;         /* annotation offset        */
        Pchar           *str;           /* text string              */
} atx;
struct {
        Pint            num_sets;       /* number of sets */
        Ppoint_list3    *sets;          /* list of sets             */
} fas3;
struct {
        Pint            num_sets;       /* number of sets */
        Ppoint_list     *sets;          /* list of sets             */
} fas;
struct {
        Pparallelogram  pgram;          /* parallelogram  */
        Pdim            dim;            /* dimension                */
        Pint            *colrs;         /* color array              */
} ca3;
struct {
        Prect           rect;           /* rectangle                */
        Pdim            dim;            /* dimension                */
        Pint            *colrs;         /* color array              */
} ca;
struct {
        Pint            id;             /* GDP3 id                  */
        Ppoint_list3    pts;            /* pt list          */
        Pgdp_data3      rec;            /* data record              */
} gdp3;
struct {
        Pint            id;             /* GDP id                   */
        Ppoint_list     pts;            /* point list               */
        Pgdp_data       rec;            /* data record              */
} gdp;
Ptx_prectext_prec;              /* text precision           */
Pvector             char_up;    /* char up vector  */
Ptx_pathtx_path;                /* text path                */
```

```
Ptx_align       tx_align;       /* text alignment  */
Pint_style      int_style;              /* interior style          */
Pedge_flag      edge_flag;              /* edge flag               */
Ppoint          pt;                     /* pat. size or ref pt     */
struct {
        Ppoint3         pt;             /* pattern ref point       */
        Pvector3vec[2];         /* vectors                 */
} pat;
Pint_list name_set;             /* name sets               */
struct {
        Paspect         id;             /* attribute id            */
        Pasf            asf;            /* asf                     */
} asf;
struct {
        Pcomp_type      comp;           /* composition type        */
        Pmatrix3        mat;            /* matrix                  */
} loc_tran3;
struct {
        Pcomp_type      comp;           /* composition type        */
        Pmatrix         mat;            /* matrix                  */
} loc_tran;
Pmatrix3        gbl_tran3;              /* global transform 3      */
Pmatrix         gbl_tran;               /* global transform        */
struct {
        Pint            op;             /* operator                */
        Phalf_space_list3 hsp;          /* half space list         */
} mclip3;
struct {
        Pint            op;             /* operator                */
        Phalf_space_list hsp;           /* half space list         */
} mclip;
Pclip           clip_ind;       /* clipping indicator      */
Pdata           appl_data;              /* application data        */
struct {
        Pint            id;             /* GSE id                  */
        Pgse_data       rec;            /* GSE data record         */
} gse;

/**** now start the PHIGS PLUS attributes ****/

Pgcolr          colr;                   /* general color           */
struct {
        Pint_list act_list;     /* activation list         */
        Pint_list deact_list;   /* deactivation list       */
} light_source_st;
Pdist_mode      face_dist_mode;         /* face distinguish mode*/
struct {
        Pint            type;           /* curve approx type       */
        Pfloat          value;          /* approx value            */
} curve_approx;
struct {
```

```
            Pint            type;           /* curve approx type    */
            Pfloat          values[2];      /* approx values  */
        } surface_approx;
        Parea_prop          area_prop;      /* area properties */
        struct {
            Ppoint3         parallelogram[3]; /* cell array         */
            Pdim            dim;            /* dimension of cell    */
            Pint            colr_type;      /* color type being used */
            Pcolrv          *colr_array;    /* array of color values */
        } cell_array_plus;
        struct {
            Pvertex_data    data;           /* per vertex data */
            Pvertex_list_ptr vertices;      /* list of vertex lists */
        } polyline_set;
        struct {
            Pvertex_list    vertices;       /* list of vertices     */
            Parea_data      data;           /* flag and color type          */
        } fill_area_plus;
        struct {
            Pint            num_lists;      /* number of lists      */
            Pvertex_edge_list *vertices;    /* list of list of vert         */
            Parea_set_data  data;           /* flags and color      */
        } fill_area_set_plus;
        struct {
            Pdim            dim;            /* dimension of mesh (X by Y)   */
            Pvertex         *vertices;      /* list of vertices     */
            Parea_data      data;           /* flags, color_type, etc.      */
        } quad_mesh;
        struct {
            Pint            order;          /* order of the curve           */
            Pint            num_knots;      /* number of knots in curve     */
            Pfloat          *knots;         /* list of knots                */
            Psurf_type      type;           /* RATIONAL or NON-RATIONAL         */
            Pint            num_points;     /* number of control points     */
            Psurf_point     *points;        /* list of control points       */
            Pfloat          range[2];       /* tmin, tmax                   */
        } nurbs_curve;
        struct {
            Pnurbs_data     udata;          /* u order and knots            */
            Pnurbs_data     vdata;          /* v order and knots            */
            Psurf_type      type;           /* RATIONAL or NON-RATIONAL         */
            Pdim            dim;            /* dim of control point list */
            Psurf_point     *points;        /* list of control pts          */
            Pfloat          range[4];       /* umin, umax, vmin, vmax       */
            Pint            num_lists;      /* number of trim curve lists   */
            Ptrim_list      *trim;          /* list of list of trim curves */
        } nurbs_surf;

} Pel_data;
```

Pel_type -- element type

```
typedef enum {
        PEL_NIL,
        PEL_POLYLINE3,
        PEL_POLYLINE,
        PEL_POLYMARKER3,
        PEL_POLYMARKER,
        PEL_TEXT3,
        PEL_TEXT,
        PEL_ANNO_TEXT_REL3,
        PEL_ANNO_TEXT_REL,
        PEL_FILL_AREA3,
        PEL_FILL_AREA,
        PEL_FILL_AREA_SET3,
        PEL_FILL_AREA_SET,
        PEL_CELL_ARRAY3,
        PEL_CELL_ARRAY,
        PEL_GDP3,
        PEL_GDP,
        PEL_LINE_IND,
        PEL_MARKER_IND,
        PEL_TEXT_IND,
        PEL_INT_IND,
        PEL_EDGE_IND,
        PEL_LINETYPE,
        PEL_LINEWIDTH_SCALE,
        PEL_LINE_COLR_IND,
        PEL_MARKER_TYPE,
        PEL_MARKER_SIZE_SCALE,
        PEL_MARKER_COLR_IND,
        PEL_TEXT_FONT,
        PEL_TEXT_PREC,
        PEL_CHAR_EXPAN,
        PEL_CHAR_SPACE,
        PEL_TEXT_COLR_IND,
        PEL_CHAR_HEIGHT,
        PEL_CHAR_UP_VECTOR,
        PEL_TEXT_PATH,
        PEL_TEXT_ALIGN,
        PEL_ANNO_CHAR_HEIGHT,
        PEL_ANNO_CHAR_UP_VECTOR,
        PEL_ANNO_PATH,
        PEL_ANNO_ALIGN,
        PEL_ANNO_STYLE,
        PEL_INT_STYLE,
        PEL_INT_STYLE_IND,
        PEL_INT_COLR_IND,
        PEL_EDGE_FLAG,
        PEL_EDGETYPE,
        PEL_EDGEWIDTH_SCALE,
```

PEL_EDGE_COLR_IND,
PEL_PATTERN_SIZE,
PEL_PATTERN_REF_PT_VECTOR,
PEL_PATTERN_REF_PT,
PEL_ADD_NAME_SET,
PEL_REMOVE_NAME_SET,
PEL_INDIV_ASF,
PEL_HLHSR_ID,
PEL_LOCAL_MODEL_TRAN3,
PEL_LOCAL_MODEL_TRAN,
PEL_GLOBAL_MODEL_TRAN3,
PEL_GLOBAL_MODEL_TRAN,
PEL_MODEL_CLIP_VOL3,
PEL_MODEL_CLIP_VOL,
PEL_MODEL_CLIP_INDICATOR,
PEL_RESTORE_MODEL_CLIP_VOL,
PEL_VIEW_IND,
PEL_EXECUTE_STRUCT,
PEL_LABEL,
PEL_APPLICATION_DATA,
PEL_GSE,
PEL_PICK_ID,

/* PHIGS PLUS Element Types */

PEL_POLYLINE_SET_3_WITH_DATA,
PEL_FILL_AREA_3_WITH_DATA,
PEL_FILL_AREA_SET_3_WITH_DATA,
PEL_TRIANGLE_STRIP_3_WITH_DATA,
PEL_QUADRILATERAL_MESH_3_WITH_DATA,
PEL_SET_OF_FILL_AREA_SET_3_WITH_DATA,
PEL_NON_UNIFORM_B_SPLINE_CURVE,
PEL_NON_UNIFORM_B_SPLINE_SURFACE,
PEL_CELL_ARRAY_3_PLUS,
PEL_TEXT_COLR,
PEL_MARKER_COLR,
PEL_EDGE_COLR,
PEL_POLYLINE_COLR,
PEL_CURVE_APPROX_CRITERIA,
PEL_POLYLINE_SHADING_METHOD,
PEL_INT_COLR,
PEL_BACK_INT_COLR,
PEL_BACK_INT_STYLE,
PEL_BACK_INT_STYLE_IND,
PEL_AREA_PROPERTIES,
PEL_BACK_AREA_PROPERTIES,
PEL_INT_SHADING_METHOD,
PEL_BACK_INT_SHADING_METHOD,
PEL_INT_REFLECTANCE_EQUATION,
PEL_BACK_INT_REFLECTANCE_EQUATION,
PEL_SURFACE_APPROX_CRITERIA,

```
        PEL_TRIM_CURVE_APPROX_CRITERIA,
        PEL_FACE_DISTINGUISHING_MODE,
        PEL_FACE_CULLING_MODE,
        PEL_LIGHT_SOURCE_STATE,
        PEL_DEPTH_CUE_IND,
        PEL_COLR_MAPPING_IND,
        PEL_RENDERING_COLR_MODEL,
        PEL_PARAM_SURFACE_CHAR,
        PEL_BACK_PARAM_SURFACE_CHAR,
        PEL_ALL
} Pel_type;
```

Pfacet -- a facet and its associated attributes

```
typedef struct {
        Pcolrv          colr;           /* facet color                  */
        Pnorm           norm;           /* facet normal vector          */
} Pfacet;
```

Pfacet_data -- facet information

```
typedef struct {
        Pint            fflag;          /* data per facet flag          */
        Pint            colr_type;      /* color type                   */
} Pfacet_data;
```

Pgcolr -- general color

```
typedef struct {
        Pint            type;           /* color type                   */
        Pcolrv          value;          /* color value                  */
} Pgcolr;
```

Pgint_rec -- general interior record

```
typedef struct {
        Pchar           *data;          /* data record                  */
} Pgint_rec;
```

Pgint_bundl -- general interior bundle

```
typedef struct {
        Pint            type;           /* general interior type        */
        Pgint_rec       rec;            /* data record                  */
} Pgint_bundl;
```

Pindex_list_list -- list of list of indices

```
typedef struct {
        Pint            num;            /* number of lists      */
        Pint_list *list;            /* list of lists            */
} Pindex_list_list;
```

Pint_attr_list -- interior attributes list

```
typedef struct {
        Pint            style;          /* interior style               */
        Pint            hatch;          /* interior hatch               */
        Pint            refl;           /* interior reflection          */
        Pint            shade;          /* interior shading             */
} Pint_attr_list;
```

Pint_bundl_plus -- extended interior bundle

```
typedef struct {
        Pint_style      int_style;      /* interior style               */
        Pint            int_index;      /* interior style index         */
        Pgcolr          int_colr; /* interior color index       */
        Pint_style      back_style;     /* interior style               */
        Pint            back_index;     /* interior style index         */
        Pgcolr          back_colr;      /* interior color index         */
        Pint            int_refl;       /* interior reflectance equation*/
        Pint            back_refl;      /* back reflectance equation    */
        Pint            int_shade;      /* interior shading method */
        Pint            back_shade;     /* back shading method          */
        Parea_prop      int_prop;       /* interior reflect. properties */
        Parea_prop      back_prop;      /* back reflect. properties  */
        Pint            approx_type;    /* surface approx. crit. type   */
```

```
        Ppoint          approx_vals;    /* surface approx. crit. vals       */
        Pint            trim_type;      /* trim curve approx. crit. type    */
        Pfloat          trim_val;       /* trim curve approx. crit. val     */
} Pint_bundl_plus;
```

Pint_fac_plus -- extended interior facilities

```
typedef struct {
        Pint            num_styles;     /* number of interior styles        */
        Pint_style      *styles;        /* list of avail interior styles    */
        Pint_list hatches;          /* list of avail hatch styles */
        Pint_list refl;             /* list of avail reflect. equations  */
        Pint_list shade;            /* list of avail shading methods     */
        Pint            num_ind;        /* number of predefined bundles     */
} Pint_fac_plus;
```

Plight_rec -- light source record

```
typedef union {
        struct {
          Pgcolr colr;              /* light source color               */
        } ambient;

        struct {
          Pgcolr colr;              /* light source color               */
          Pvector3      dir;            /* light source direction           */
        } directional;

        struct {
          Pgcolr colr;              /* light source color               */
          Ppoint3       pos;            /* light source position            */
          Pfloat  att_coef1;        /* attenuation coefficient 1*/
          Pfloat  att_coef2;        /* attenuation coefficient 2*/
        } positional;

        struct {
          Pgcolr colr;              /* light source color               */
          Ppoint3       pos;            /* light source position            */
          Pvector3      dir;            /* light source direction           */
          Pfloat  exp;              /* concentration exponent */
          Pfloat  att_coef1;        /* attenuation coefficient 1*/
          Pfloat  att_coef2;        /* attenuation coefficient 2*/
          Pfloat  angle;            /* spread angle                     */
        } spot;
} Plight_rec;
```

Plight_bundl -- light source bundle

```
typedef struct {
        Pint            type;           /* light source type            */
        Plight_rec      *rec;           /* data record                  */
} Plight_bundl;
```

Plight_fac -- light source facilities

```
typedef struct {
        Pint_list types;            /* list of light src types supported   */
        Pint            max_active;     /* number of max simul active lights       */
        Pint            num_ind;        /* # of pred. light src indices            */
} Plight_fac;
```

Pln_bundl_plus -- extended polyline bundle

```
typedef struct {
        Pint            type;           /* polyline type                */
        Pfloat          width;          /* polyline width           */
        Pint            shade;          /* polyline shading method      */
        Pint            curve_type;     /* curve approx criteria type   */
        Pfloat          curve_val;      /* curve approx criteria value  */
        Pgcolr          colr;           /* polyline color           */
} Pln_bundl_plus;
```

Pln_fac_plus -- extended polyline facilities

```
typedef struct {
        Pint_list types;            /* list of line types               */
        Pint            num_widths;     /* number of avail line widths  */
        Pfloat          nom_width;      /* nominal line width           */
        Pfloat          min_width;      /* min line width           */
        Pfloat          max_width;      /* max line width           */
        Pint            num_ind;        /* number of predefined bundles */
        Pint_list shades;           /* list of avail shade methods      */
} Pln_fac_plus;
```

Pmk_bundl_plus -- polymarker bundle

```
typedef struct {
        Pint            type;           /* makrer type                  */
        Pfloat          size;           /* makrer size scale factor */
        Pgcolr          colr;           /* marker color                 */
} Pmk_bundl_plus;
```

Pmod_ws_plus -- extended dynamic modification of wks attributes

```
typedef struct {
        Pmod_type       depth;          /* depth cue                    */
        Pmod_type       light;          /* light source                 */
        Pmod_type       colr_map;       /* color mapping        */
} Pmod_ws_plus;
```

Pnorm -- normal

```
typedef  Ppoint3  Pnorm;
```

Pnurbs_data -- data corresponding to nurbs

```
typedef struct {
        Pint            order;          /* order of the curve           */
        Pint            num_knots;      /* number of knots              */
        Pfloat          *knots;         /* list of knots                */
} Pnurbs_data;
```

Popt_data -- optional data

```
typedef struct {
        Pcolrv          colr;           /* colr                         */
        Pnorm           normal;         /* normal                       */
} Popt_data;
```

Ppline_vertex -- polyline set vertex

```
typedef struct {
    Ppoint3         point;      /* polyline set point          */
    Pcolrv          colr;               /* vertex color                */
} Ppline_vertex;
```

Ppline_vertex_list -- polyline set vertex list

```
typedef struct {
    Pint            length;     /* length of the vertex array  */
    Ppline_vertex   *vertices;   /* Array of vertices          */
} Ppline_vertex_list;
```

Ppoint_list4 -- list of point 4

```
typedef struct {
        Pint        num_points;     /* number of points in list */
        Ppoint4     *points;        /* list of points           */
} Ppoint_list4;
```

Ppoint4 -- point 4

```
typedef struct {
        Pfloat      x;              /* x coordinate             */
        Pfloat      y;              /* y coordinate             */
        Pfloat      z;              /* z coordinate             */
        Pfloat      w;              /* w coordinate             */
} Ppoint4;
```

Ppt_bundl_plus -- extended pattern bundle

```
typedef struct {
        Pdim        dim;            /* pattern's dimensions     */
        Pint        colr_type;      /* color type               */
        Pcolrv      *pat_array;     /* color index array        */
} Ppt_bundl_plus;
```

Pset_of_fill_area3_data -- set of fill area3 data flags, facets, edges, etc

```
typedef struct {
        Pint            fflag;          /* facet flag                   */
        Pint            eflag;          /* edge flag                    */
        Pint            vflag;          /* vertex flag                  */
        Pint            colr_type;      /* color type                   */
        Popt_data       *fdata;         /* facet data                   */
        Pedge_list_list *edata;         /* edge data                    */
} Pset_of_fill_area3_data;
```

Psurface -- surface

```
typedef enum {
        PSURF_EMPTY,
        PSURF_NOTEMPTY
} Psurface;
```

Psurf_type -- surface type

```
typedef enum {
        PRATIONAL,
        PNON_RATIONAL
} Psurf_type;
```

Psurf_point -- 3 or 4d point used for surface control points

```
typedef union {
        Ppoint3         point3d;        /* 3d point                     */
        Ppoint4         point4d;        /* 4d point                     */
} Psurf_point;
```

Ptrim -- trimming curve data

```
typedef struct {
        Pedge_flag      visibility;     /* edge visibility flag         */
        Pint            order;          /* order of the curve           */
        Pfloat_list     knots;          /* list of knots                */
        Pcurve_type     type;           /* curve type                   */
```

```
        Pint            num_points;     /* number of points             */
        Pcurve_point    *points;        /* list of points               */
        Pfloat          tmin, tmax;     /* range of points          */
} Ptrim;
```

Ptrim_list -- list of trimming curves

```
typedef struct {
        Pint            num;            /* number of trimming curves    */
        Ptrim           *trim_curve;    /* array of trim curves         */
} Ptrim_list;
```

Ptx_bundl_plus -- text bundle

```
typedef struct {
        Pint            font;           /* text font                    */
        Ptx_precprecision;          /* text precision               */
        Pfloat          exp_factor;     /* character expansion factor   */
        Pfloat          spacing;        /* character spacing            */
        Pgcolr          colr;           /* text color index             */
} Ptx_bundl_plus;
```

Puniformity -- uniformity

```
typedef enum {
        PUNIFORM,
        PNON_UNIFORM
} Puniformity;
```

Pvertex -- vertex

```
typedef struct {
        Ppoint3         point;          /* 3D point                     */
        Popt_data       data;           /* optional data                */
} Pvertex;
```

Pvertex_data -- vertex information

```
typedef struct {
        Pint            vflag;          /* data per vertex flag         */
        Pint            colr_type;      /* color type                   */
} Pvertex_data;
```

Pvertex_edge_list -- vertex and edge list

```
typedef struct {
        Pint            length;         /* length of the vertex array   */
        Pvertex         *vertices;      /* Array of vertices            */
        Pedge_flag      *edges;         /* Array of edge flags          */
} Pvertex_edge_list;
```

Pvertex_list -- vertex list

```
typedef struct {
        Pint            length;         /* length of the vertex array   */
        Pvertex         *vertices;      /* Array of vertices            */
} Pvertex_list;
```

Pvertex_list_ptr -- list of list of vertices

```
typedef struct {
        Pint            num_lists;      /* number of lists of vertices  */
        Pvertex_list    *list_ptr;      /* list of list of vertices     */
} Pvertex_list_ptr;
```

Pws_tables_plus -- length of extended workstation tables

```
typedef struct {
        Pint            depth;          /* depth cue tables             */
        Pint            light;          /* light source tables          */
        Pint            colr_map;       /* color mapping tables         */
} Pws_tables_plus;
```

POLYLINE SET 3 WITH DATA
void ppolyline_set3_data (num_sets, data, vertices)

```
Pint      number of sets of vertices  num_sets;     */
Pvertex_data                          data;         */
Plist_of_vertex_list                  *vertices;    */
```

FILL AREA 3 WITH DATA
void pfill_area3_data (num_vertices, vertices, data)

```
Pint      number of vertices          num_sets;     */
Plist of vertices                     *vertices;    */
Pfacet_data color type                *data;        */
```

FILL AREA SET 3 WITH DATA
void pfill_area_set3_data (bounds, list, data)

```
Pint      number of lists of vertices bounds;       */
Plist_of_list_of_vertices             *vertices;    */
Pfacet_data, color type, etc          *data;        */
```

TRIANGLE STRIP 3 WITH DATA
void ptriangle3_data (num_vertices, vertices, data)

```
Pint      number of vertices          num_vertices  */
Plist of vertices                     *vertices;    */
Pfacet_data color type                *data;        */
```

QUADRILATERAL MESH 3 WITH DATA
void pquad_mesh3_data (dim, vertices, data)

```
Pint2     dimension of the mesh (X by Y) *dim;      */
Plist of vertices                     *vertices;    */
Pfacet_data color_type, etc           *data;        */
```

SET OF FILL AREA SET 3 WITH DATA
void pset_fill_area_set3_data (vertices, data, num_indices, indices)

```
Plist_of_list_of_vertices             *vertices;    */
Pfacet_data color_type, etc.          data;         */
```

```
/* number of indices        num_indices;      */
/* index_list_list  list of indices*indices;  */
```

NON-UNIFORM B-SPLINE CURVE

void pnurbs_curve (order, num_knots, knots, type, num_points, points, range)

```
/* order of the curve  order;                 */
/* number of knots in the curve  num_knots;   */
/* list of knots       *knots;                */
/* RATIONAL or type;   NON-RATIONAL           */
/* number of control points  num_points;      */
/* list of control points  *points;           */
/* tmin, tmax          range[2];              */
```

NON-UNIFORM B-SPLINE SURFACE

void pnurbs_surface (udata, vdata, type, dim, points, range, num_lists, trim)

```
/* nurbs_data and knots  *udata;              */
/* nurbs_data and knots  *vdata;              */
/* RATIONAL or type;   NON-RATIONAL           */
/* dimensions of control point list  *dim;    */
/* list of list of control points  *points;   */
/* umin, umax, vmin, vmax  range[4];          */
/* number of trimming curve lists  num_lists;  */
/* list of list of trim curves  *trim;        */
```

EXTENDED CELL ARRAY 3

void pcell_array3_plus (parallelogram, dim, color_type, color_array)

```
/* parallelogram of cells  parallelogram[3];  */
/* dimension of cell DX and DY  dim;          */
/* color type being used  color_type;         */
/* array of color values  *color_array;       */
```

---

3. Utility Functions To Support Output Primitives

COMPUTE FILL AREA SET GEOMETRIC NORMAL

void pcomp_fill_area_set_norm (num_bounds, points, err, normal)

```
/* number of boundaries */ num_bounds;
/* list of list3 */ *points;
/* OUT error indicator */ *err;
/* OUT unit normal vector */ *normal;
```

---

4. Individual Attribute Selection

SET DEPTH CUE INDEX
void pset_depth_ind (index)

```
Pint    depth cue index index;          */
```

SET COLOR MAPPING INDEX
void pset_colr_map_ind (index)

```
Pint    color mapping index index;      */
```

SET RENDERING COLOR MODEL
void pset_rend_colr_model (model)

```
Pint    rendering color model model;    */
```

SET AREA PROPERTIES
void pset_area_prop (prop)

```
Parea_properties *prop;                 */
```

SET BACK AREA PROPERTIES
void pset_back_area_prop (prop)

```
Pback_area_properties *prop;            */
```

SET POLYLINE SHADING METHOD
void pset_line_shade_method (method)

```
Pint    polyline shading method method  */
```

SET BACK INTERIOR STYLE
void pset_back_int_style (style)

```
/* int_style interior style; */
```

SET BACK INTERIOR STYLE INDEX
void pset_back_int_style_ind (index)

```
/* int index; */
```

SET INTERIOR SHADING METHOD
void pset_int_shade_method (method)

```
/* int method; */
```

SET BACK INTERIOR SHADING METHOD
void pset_back_int_shade_method (method)

```
/* int method; */
```

SET INTERIOR REFLECTANCE EQUATION
void pset_int_refl_eq (equation)

```
/* int equation; */
```

SET BACK INTERIOR REFLECTANCE EQUATION
void pset_back_int_refl_eq (equation)

```
/* int equation; */
```

SET LIGHT SOURCE STATE
void pset_light_st (act_list, deact_list)

```
/* Int_list *act_list; */
/* Int_list *deact_list; */
```

SET FACE DISTINGUISHING MODE
void pset_face_dist_mode (mode)

```
/* face_distinguishing_mode */
/* dist_mode mode;
```

SET FACE CULLING MODE
void pset_face_cull_mode (mode)

```
/* cull_mode mode; */
/* cull_mode mode;
```

SET POLYLINE COLOR
void pset_line_color (color)

```
/* polyline_color */
/* polyline_color;
```

SET POLYMARKER COLOR
void pset_marker_color (color)

```
/* polymarker_color */
/* polymarker;color
```

SET TEXT COLOR
void pset_text_color (color)

```
/* text_color */
/* text_color;
```

SET INTERIOR COLOR
void pset_int_color (color)

```
/* interior_color */
/* interior_color;
```

SET BACK INTERIOR COLOR
void pset_back_int_color (color)

```
/* back_interior_color */
/* back_interior_color
```

SET EDGE COLOR
void pset_edge_color (color)

```
Pint edge color;                    */
```

SET CURVE APPROXIMATION CRITERIA
void pset_curve_approx_crit (type, value)

```
Pint approx type;                   */
Pfloat approx value;                */
```

SET TRIMMING CURVE APPROXIMATION CRITERIA
void pset_trim_curve_approx_crit (type, value)

```
Pint approx type;                   */
Pfloat approx value;                */
```

SET SURFACE APPROXIMATION CRITERIA
void pset_surface_approx_crit (type, values)

```
Pint approx type;                   */
Pfloat approx values[2];            */
```

SET PARAMETRIC SURFACE CHARACTERISTICS
void pset_param_surf_char (type, data)

```
Pint type of parametric surface     */
Pparam_surf_data *data;     */
```

SET BACK PARAMETRIC SURFACE CHARACTERISTICS
void pset_back_param_surf_char (type, data)

```
Pint type of parametric surface     */
Pparam_surf_data *data;     */
```

5. Aspect Source Flag Setting

SET INDIVIDUAL ASF
void pset_indiv_asf (asf_id, asf_source)

```
/* attribute_source identifier */
/* attribute_source;           */
```

---

6. Workstation Attribute Table Definition

SET POLYLINE REPRESENTATION PLUS
void pset_line_rep_plus (ws_id, index, rep)

```
Pin workstation id  ws_id;          */
Pin polyline index  index;          */
Pin extended polyline bundle entry  */
```

SET POLYMARKER REPRESENTATION PLUS
void pset_marker_rep_plus (ws_id, index, rep)

```
Pin workstation id  ws_id;          */
Pin polymarker index  index;        */
Pin polymarker bundle entry         */
```

SET TEXT REPRESENTATION PLUS
void pset_text_rep_plus (ws_id, index, rep)

```
Pin workstation id  ws_id;          */
Pin text index      index;          */
Pin extended text bundle entry      */
```

SET INTERIOR REPRESENTATION PLUS
void pset_int_rep_plus (ws_id, index, rep)

```
Pin workstation id  ws_id;          */
Pin interior index  index;          */
Pin extended interior bundle entry  */
```

SET EDGE REPRESENTATION PLUS
void pset_edge_rep_plus (ws_id, index, rep)

```
Pin workstation id  ws_id;          */
Pin edge index      index;          */
Pin edge bundle entry               */
```

SET PATTERN REPRESENTATION PLUS

void pset_pattern_rep_plus (ws_id, index, rep)

```
/* workstation id   ws_id;          */
/* pattern index    index;          */
/* extended pattern bundle entry    */
```

SET DEPTH CUE REPRESENTATION

void pset_depth_rep (ws_id, index, rep)

```
/* workstation id   ws_id;          */
/* depth cue index  index;          */
/* depth cue bundle entry           */
```

SET LIGHT SOURCE REPRESENTATION

void pset_light_rep (ws_id, index, rep)

```
/* workstation id   ws_id;          */
/* light source index  index;       */
/* light source bundle entry        */
```

SET COLOR MAPPING REPRESENTATION

void pset_colr_map_rep (ws_id, index, rep)

```
/* workstation id   ws_id;          */
/* color mapping index  index;      */
/* color mapping bundle entry       */
```

---

7. Inquiry Functions For Workstation State List

INQUIRE POLYLINE REPRESENTATION PLUS
void pinq_line_rep_plus (ws_id, index, type, error_ind, line_rep)

```
/* workstation id  ws_id;                  */
/* polyline index  index;                  */
/* returned value type;-- PSET or PREALIZED */
/* OUT error indicator *error_ind;         */
/* OUT extended line_rep;*line_rep;        */
```

INQUIRE POLYMARKER REPRESENTATION PLUS
void pinq_marker_rep_plus (ws_id, index, type, error_ind, marker_rep)

```
/* workstation id  ws_id;                  */
/* polymarker index index;                 */
/* returned value type;-- PSET or PREALIZED */
/* OUT error indicator *error_ind;         */
/* OUT marker plus *marker_rep;            */
```

INQUIRE TEXT REPRESENTATION PLUS
void pinq_text_rep_plus (ws_id, index, return_type, error_ind, text_rep)

```
/* workstation id  ws_id;                  */
/* text index      index;                  */
/* returned value type;-- PSET or PREALIZED */
/* OUT error indicator *error_ind;         */
/* OUT extended plus *text_rep;            */
```

INQUIRE INTERIOR REPRESENTATION PLUS
void pinq_int_rep_plus (ws_id, index, type, error_ind, int_rep)

```
/* workstation identifier ws_id;           */
/* interior index  index;                  */
/* type of returned value type;            */
/* OUT error indicator *error_ind;         */
/* OUT interior plus *int_rep;representation */
```

INQUIRE EDGE REPRESENTATION PLUS
void pinq_edge_rep_plus (ws_id, index, type, error_ind, rep)

```
Pint    ws_id;          /* workstation id */
Pint    index;          /* edge index */
Pinq_type type;         /* inquired value type -- PSET or PREALIZED */
Pint    *error_ind;     /* OUT error indicator */
Pedge_bundle_plus *edge_bundle_rep;  /* OUT bundle table entry */
```

INQUIRE PATTERN REPRESENTATION PLUS

void pinq_pat_rep_plus (ws_id, index, return_type, size, error_ind, buffer, pattern_rep, total_size)

```
Pint    ws_id;          /* workstation id */
Pint    index;          /* pattern index */
Pinq_type return_type;  /* inquired value type -- PSET or PREALIZED */
size_t  size;           /* size of application buffer */
Pint    *error_ind;     /* OUT error indicator */
void    *buffer;        /* OUT pointer to buffer */
Ppattern_plus *pattern_rep;  /* OUT bundle table entry */
size_t  *total_size;    /* OUT actual size of buffer */
```

INQUIRE LIST OF DEPTH CUE INDICES

void pinq_list_depth_ind (ws_id, start, length, error_ind, indices, total_length)

```
Pint    ws_id;          /* workstation id */
Pint    start;          /* starting position */
Pint    length;         /* length of application list */
Pint    *error_ind;     /* OUT error indicator */
Pint_list *indices;     /* OUT list of depth cue table indices */
Pint    *total_length;  /* OUT length of list */
```

INQUIRE DEPTH CUE REPRESENTATION

void pinq_depth_rep (ws_id, index, type, error_ind, depth_rep)

```
Pint    ws_id;          /* workstation id */
Pint    index;          /* depth source index */
Pinq_type type;         /* inquired value type -- PSET or PREALIZED */
Pint    *error_ind;     /* OUT error indicator */
Pdepth_cue_plus *depth_rep;  /* OUT depth cue table entry */
```

INQUIRE LIST OF LIGHT SOURCE INDICES

void pinq_list_light_ind (ws_id, start, length, error_ind, indices, total_length)

```
Pint    ws_id;          /* workstation id */
```

```
*int starting position start;                    */
*int length of application length;  list         */
*int OUT error indicator *error_ind;             */
*int OUT *indices; depth cue table indices       */
*int OUT length of list *total_length;           */
```

INQUIRE LIGHT SOURCE REPRESENTATION

void pinq_light_rep (ws_id, index, type, error_ind, light_rep)

```
*int workstation id ws_id;                       */
*int light source index index;                   */
*int returned value type type;                   */
*int OUT error indicator *error_ind;             */
*light_bundl OUT light source *light_rep;        */
```

INQUIRE COLOR MAPPING STATE

void pinq_colr_map_st (ws_id, method, error_ind, rec)

```
*int workstation id ws_id;                       */
*int color mapping method method;                */
*int OUT error indicator *error_ind;             */
*colr_map_method_rec *rec;                       */
```

INQUIRE LIST OF COLOR MAPPING INDICES

void pinq_list_colr_map_ind (ws_id, length, start, error_ind, indices, total_length)

```
*int workstation identifier ws_id;               */
*int starting position start;                    */
*int length of application length;  list         */
*int OUT error indicator *error_ind;             */
*int OUT *indices; color indices                 */
*int OUT size of list *total_length;             */
```

INQUIRE COLOR MAPPING REPRESENTATION

void pinq_colr_map_rep (ws_id, index, return_type, size, error_ind, buffer, bundl, total_size)

```
*int workstation identifier ws_id;               */
*int color table index index;                    */
*int type of returned value type;  value         */
*size size of application buffer size;           */
*int OUT error indicator *error_ind;             */
```

```
/* OUT pointer to buffer;                    */
/* OUT map_bundle_handle entry               */
/* OUT actual size of buffer;                */
```

8. Inquiry Functions For Workstation Description Table

INQUIRE DIRECT COLOR MODEL FACILITIES

void pinq_dir_colr_model_facil (type, start, length, error_ind, models, total_length)

```
/* IN workstation type type;                    */
/* IN starting position start;                  */
/* IN length of application length;list         */
/* OUT error indicator *error_ind;              */
/* OUT list of color models *models             */
/* OUT length of list in PHIGS *total_length;   */
```

INQUIRE RENDERING COLOR MODEL FACILITIES

void pinq_rend_colr_model_facil (type, start, length, error_ind, models, total_length)

```
/* IN workstation type type;                    */
/* IN starting position start;                  */
/* IN length of application length;list         */
/* OUT error indicator *error_ind;              */
/* OUT list of color models *models             */
/* OUT length of list in PHIGS *total_length;   */
```

INQUIRE DYNAMICS OF WORKSTATION PLUS

void pinq_dyn_ws_plus (type, error_ind, attr)

```
/* IN workstation type type                     */
/* OUT error indicator *error_ind;              */
/* OUT attributes dynamics *attr;               */
```

INQUIRE POLYLINE FACILITIES PLUS

void pinq_line_facil_plus (type, line_start, line_length, shade_start, shade_length, error_ind, facilities, tot_shade_len,tot_line_len)

```
/* IN workstation type type;                    */
/* IN starting position line_start;types        */
/* IN length of app list for line types         */
/* IN starting position shade_start;types       */
/* IN length of app list for shade types        */
/* OUT error indicator *error_ind;              */
/* OUT extended polyline facilities *facilities */
/* OUT size of shade list in PHIGS *tot_shade_len;  */
/* OUT size of line list in PHIGS *tot_line_len;  */
```

INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS
void pinq_pred_line_rep_plus (type, index, error_ind, bundle)

```
Pint *workstation type type;                    */
Pint *predefined index index;                   */
Pint *OUT error indicator error_ind;            */
Pint *OUT predefined polyline rep               */
```

INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS
void pinq_pred_marker_rep_plus (type, index, error_ind, bundle)

```
Pint *workstation type type;                    */
Pint *predefined index index;                   */
Pint *OUT error indicator error_ind;            */
Pint *OUT predefined bundle marker rep   */
```

INQUIRE PREDEFINED TEXT REPRESENTATION PLUS
void pinq_pred_text_rep_plus (type, index, error_ind, bundle)

```
Pint *workstation type type;                    */
Pint *predefined index index;                   */
Pint *OUT error indicator error_ind;            */
Ptx_bundle_plus *bundle text rep                */
```

INQUIRE PREDEFINED EDGE REPRESENTATION PLUS
void pinq_pred_edge_rep_plus (type, index, error_ind, bundle)

```
Pint *workstation type type;                    */
Pint *predefined index index;                   */
Pint *OUT error indicator error_ind;            */
Pedge_bundle_plus *bundle edge rep              */
```

INQUIRE INTERIOR FACILITIES PLUS
void pinq_int_facil_plus (type, start, length, error_ind, facilities,total_length)

```
Pint *workstation type type;                    */
Pint *list of starting positions start;         */
Pint *style, hatch, refl, shade list sz         */
Pint *OUT error indicator error_ind;            */
```

```
/* OUT extended interior facilities */
/* OUT size of list in PHIGS */
```

INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS

void pinq_pred_int_rep_plus (type, index, error_ind, bundle)

```
/* IN workstation type; */
/* IN predefined index; */
/* OUT error indicator; */
/* OUT predefined bundle; */
```

INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS

void pinq_pred_pat_rep_plus (type, index, size, error_ind, buffer, bundle, total_size)

```
/* IN workstation type; */
/* IN predefined pattern index; */
/* IN size of application buffer; */
/* OUT error indicator; */
/* OUT application buffer; */
/* OUT extended pattern bundle entry */
/* OUT actual size of buffer; */
```

INQUIRE DEPTH CUE FACILITIES

void pinq_depth_facil (type, error_ind, indices)

```
/* IN workstation type; */
/* OUT error indicator; */
/* OUT number of depth cue indices */
```

INQUIRE PREDEFINED DEPTH CUE REPRESENTATION

void pinq_pred_depth_rep (type, index, error_ind, bundle)

```
/* IN workstation type; */
/* IN predefined index; */
/* OUT error indicator; */
/* OUT predefined depth rep */
```

INQUIRE LIGHT SOURCE FACILITIES

void pinq_light_facil (type, start, length, error_ind, facilities, total_length)

```
/* IN workstation type type;                    */
/* IN starting position start;                  */
/* IN length of application list length;        */
/* OUT error indicator *error_ind;              */
/* OUT light source facilities *facilities;     */
/* OUT size of list in PHIGS *total_length;     */
```

INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION

void pinq_pred_light_rep (type, index, error_ind, bundle)

```
/* IN workstation type type;                    */
/* IN predefined index index;                   */
/* OUT error indicator *error_ind;              */
/* OUT predefined bundle *bundle;  source rep   */
```

INQUIRE COLOR MAPPING FACILITIES

void pinq_colr_map_facil (type, length, start, error_ind,total_length,facilities)

```
/* IN workstation type type;                    */
/* IN length of application list length;        */
/* IN starting position start;                  */
/* OUT error indicator *error_ind;              */
/* OUT length of list in PHIGS *total_length;   */
/* OUT color mapping facilities *facilities;    */
```

INQUIRE COLOR MAPPING METHOD FACILITIES

void pinq_colr_map_method_facil (type, method, error_ind, rec)

```
/* IN workstation type type                     */
/* IN color mapping method method               */
/* OUT error indicator *error_ind;              */
/* OUT data record *rec;                        */
```

INQUIRE PREDEFINED COLOR MAPPING REPRESENTATION

void pinq_pred_colr_map_rep (type, index, size, error_ind, buffer, bundl, total_size)

```
/* IN workstation type type                     */
/* IN predefined color mapping index index;     */
/* IN size of application buffer size;          */
```

```
/* OUT error indicator                    */
/* OUT pointer to buffer                  */
/* OUT mapping handle entry               */
/* OUT actual size of buffer              */
```

INQUIRE CURVE AND SURFACE FACILITIES

void pinq_curve_surface_facil (type, start, length, error_ind, facilities, total_length)

```
/* IN workstation type                    */
/* IN starting position                   */
/* IN size of application lengths         */
/* OUT error indicator                    */
/* OUT curve and surface facilities       */
/* OUT size of list total_length          */
```

INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS

void pinq_ws_st_table_plus (type, error_ind, lengths)

```
/* IN workstation type                    */
/* OUT error indicator                    */
/* OUT lengths of workstation tables      */
```

---

9. Inquiry Functions For Structure Content

INQUIRE DIRECT COLOR MODEL FACILITIES

void pinq_dir_colr_model_facil (type, start, length, error_ind, models, total_length)

```
/* IN workstation type type;                    */
/* IN starting position start;                  */
/* IN length of application length;  */
/* IN OUT error indicator error_ind;            */
/* IN OUT list of color models                  */
/* IN OUT length of list total_length;          */
```

INQIURE CURRENT ELEMENT TYPE AND SIZE

void pinq_cur_elem_type_size (error_ind, type, size)

```
/* IN OUT error indicator error_ind;            */
/* IN OUT element type                          */
/* IN OUT element size size;                    */
```

INQIURE ELEMENT TYPE AND SIZE

void pinq_elem_type_size (struct_id, element, error_ind, type, size)

```
/* IN structure identifier struct_id;           */
/* IN element number element;                   */
/* IN OUT error indicator error_ind;            */
/* IN OUT element type                          */
/* IN OUT element size size;                    */
```

PHIGS PLUS FUNCION NAME INDEX

POLYLINE SET 3 WITH DATA 21
FILL AREA 3 WITH DATA 21
FILL AREA SET 3 WITH DATA 21
TRIANGLE STRIP 3 WITH DATA 21
QUADRILATERAL MESH 3 WITH DATA 21
SET OF FILL AREA SET 3 WITH DATA 21
NON-UNIFORM B-SPLINE CURVE 22
NON-UNIFORM B-SPLINE SURFACE 22
EXTENDED CELL ARRAY 3 22
COMPUTE FILL AREA SET GEOMETRIC NORMAL 23
SET DEPTH CUE INDEX 24
SET COLOR MAPPING INDEX 24
SET RENDERING COLOR MODEL 24
SET AREA PROPERTIES 24
SET BACK AREA PROPERTIES 24
SET POLYLINE SHADING METHOD 24
SET BACK INTERIOR STYLE 24
SET BACK INTERIOR STYLE INDEX 25
SET INTERIOR SHADING METHOD 25
SET BACK INTERIOR SHADING METHOD 25
SET INTERIOR REFLECTANCE EQUATION 25
SET BACK INTERIOR REFLECTANCE EQUATION 25
SET LIGHT SOURCE STATE 25
SET FACE DISTINGUISHING MODE 26
SET FACE CULLING MODE 26
SET POLYLINE COLOR 26
SET POLYMARKER COLOR 26
SET TEXT COLOR 26
SET INTERIOR COLOR 26
SET BACK INTERIOR COLOR 26
SET EDGE COLOR 27
SET CURVE APPROXIMATION CRITERIA 27
SET TRIMMING CURVE APPROXIMATION CRITERIA 27
SET SURFACE APPROXIMATION CRITERIA 27
SET PARAMETRIC SURFACE CHARACTERISTICS 27
SET BACK PARAMETRIC SURFACE CHARACTERISTICS 27
SET INDIVIDUAL ASF 29
SET POLYLINE REPRESENTATION PLUS 30
SET POLYMARKER REPRESENTATION PLUS 30
SET TEXT REPRESENTATION PLUS 30
SET INTERIOR REPRESENTATION PLUS 30
SET EDGE REPRESENTATION PLUS 30
SET PATTERN REPRESENTATION PLUS 31
SET DEPTH CUE REPRESENTATION 31
SET LIGHT SOURCE REPRESENTATION 31

SET COLOR MAPPING REPRESENTATION 31
INQUIRE POLYLINE REPRESENTATION PLUS 32
INQUIRE POLYMARKER REPRESENTATION PLUS 32
INQUIRE TEXT REPRESENTATION PLUS 32
INQUIRE INTERIOR REPRESENTATION PLUS 32
INQUIRE EDGE REPRESENTATION PLUS 32
INQUIRE PATTERN REPRESENTATION PLUS 33
INQUIRE LIST OF DEPTH CUE INDICES 33
INQUIRE DEPTH CUE REPRESENTATION 33
INQUIRE LIST OF LIGHT SOURCE INDICES 33
INQUIRE LIGHT SOURCE REPRESENTATION 34
INQUIRE COLOR MAPPING STATE 34
INQUIRE LIST OF COLOR MAPPING INDICES 34
INQUIRE COLOR MAPPING REPRESENTATION 34
INQUIRE DIRECT COLOR MODEL FACILITIES 36
INQUIRE RENDERING COLOR MODEL FACILITIES 36
INQUIRE DYNAMICS OF WORKSTATION PLUS 36
INQUIRE POLYLINE FACILITIES PLUS 36
INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS 37
INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS 37
INQUIRE PREDEFINED TEXT REPRESENTATION PLUS 37
INQUIRE PREDEFINED EDGE REPRESENTATION PLUS 37
INQUIRE INTERIOR FACILITIES PLUS 37
INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS 38
INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS 38
INQUIRE DEPTH CUE FACILITIES 38
INQUIRE PREDEFINED DEPTH CUE REPRESENTATION 38
INQUIRE LIGHT SOURCE FACILITIES 39
INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION 39
INQUIRE COLOR MAPPING FACILITIES 39
INQUIRE COLOR MAPPING METHOD FACILITIES 39
INQUIRE PREDEFINED COLOR MAPPING REPRESENTATION 39
INQUIRE CURVE AND SURFACE FACILITIES 40
INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS 40
INQUIRE DIRECT COLOR MODEL FACILITIES 41
INQIURE CURRENT ELEMENT TYPE AND SIZE 41
INQIURE ELEMENT TYPE AND SIZE 41

PHIGS PLUS C LANGUAGE BINDING INDEX

void ppolyline_set3_data 21
void pfill_area3_data 21
void pfill_area_set3_data 21
void ptriangle3_data 21
void pquad_mesh3_data 21
void pset_fill_area_set3_data 21
void pnurbs_curve 22
void pnurbs_surface 22
void pcell_array3_plus 22
void pcomp_fill_area_set_norm 23
void pset_depth_ind 24
void pset_colr_map_ind 24
void pset_rend_colr_model 24
void pset_area_prop 24
void pset_back_area_prop 24
void pset_line_shade_method 24
void pset_back_int_style 24
void pset_back_int_style_ind 25
void pset_int_shade_method 25
void pset_back_int_shade_method 25
void pset_int_refl_eq 25
void pset_back_int_refl_eq 25
void pset_light_st 25
void pset_face_dist_mode 26
void pset_face_cull_mode 26
void pset_line_color 26
void pset_marker_color 26
void pset_text_color 26
void pset_int_color 26
void pset_back_int_color 26
void pset_edge_color 27
void pset_curve_approx_crit 27
void pset_trim_curve_approx_crit 27
void pset_surface_approx_crit 27
void pset_param_surf_char 27
void pset_back_param_surf_char 27
void pset_indiv_asf 29
void pset_line_rep_plus 30
void pset_marker_rep_plus 30
void pset_text_rep_plus 30
void pset_int_rep_plus 30
void pset_edge_rep_plus 30
void pset_pattern_rep_plus 31
void pset_depth_rep 31
void pset_light_rep 31

void pset_colr_map_rep 31
void pinq_line_rep_plus 32
void pinq_marker_rep_plus 32
void pinq_text_rep_plus 32
void pinq_int_rep_plus 32
void pinq_edge_rep_plus 32
void pinq_pat_rep_plus 33
void pinq_list_depth_ind 33
void pinq_depth_rep 33
void pinq_list_light_ind 33
void pinq_light_rep 34
void pinq_colr_map_st 34
void pinq_list_colr_map_ind 34
void pinq_colr_map_rep 34
void pinq_dir_colr_model_facil 36
void pinq_rend_colr_model_facil 36
void pinq_dyn_ws_plus 36
void pinq_line_facil_plus 36
void pinq_pred_line_rep_plus 37
void pinq_pred_marker_rep_plus 37
void pinq_pred_text_rep_plus 37
void pinq_pred_edge_rep_plus 37
void pinq_int_facil_plus 37
void pinq_pred_int_rep_plus 38
void pinq_pred_pat_rep_plus 38
void pinq_depth_facil 38
void pinq_pred_depth_rep 38
void pinq_light_facil 39
void pinq_pred_light_rep 39
void pinq_colr_map_facil 39
void pinq_colr_map_method_facil 39
void pinq_pred_colr_map_rep 39
void pinq_curve_surface_facil 40
void pinq_ws_st_table_plus 40
void pinq_dir_colr_model_facil 41
void pinq_cur_elem_type_size 41
void pinq_elem_type_size 41

PREFACE

This report presents a C binding for PHIGS PLUS. The PHIGS PLUS functions are taken from the ISO/IEC 9592-4:19xx specification dated July 25, 1989. At the present time, that document is in the draft stage and is not an accepted standard. Thus, we can expect that some of the functions will be changed in the future, and that new functions may be added. At the time of this printing, this binding is current with that document and will probably change as the specification progresses.

This C binding was written to be as similar as the American National Standard PHIGS C binding dated June 20, 1988. The data structures and function parameters in this paper have been defined to be as consistent with that document as possible.

TABLE OF CONTENTS

1. PHIGS PLUS Data Structures 1

2. Output Primitive Structure Elements 21

3. Utility Functions To Support Output Primitives 23

4. Individual Attribute Selection 24

5. Aspect Source Flag Setting 29

6. Workstation Attribute Table Definition 30

7. Inquiry Functions For Workstation State List 32

8. Inquiry Functions For Workstation Description Table 36

9. Inquiry Functions For Structure Content 41