```
// PWG inline feedback for Google Cloud Description Document protobuf definitions.
//
// PWG suggestions are highlighted like this.  We have only provided feedback on the
// printer description thus far.
//
// Text with line-through means to delete the corresponding line in the original
// CDD protobuf definition, e.g.:
//
//    ~~required string foo;~~
//
// Otherwise hightlighted lines are definitions to be added...

// Description of a cloud-enabled device's capabilities and properties. Also
// known as CDD.
message CloudDeviceDescription {
  // Read-only property that can be used by vendors to further describe the
  // device.
  message VendorProperty {
    required string id = 1;
    required string value = 2;
  }

  // Version of the CDD in the form of "X.Y" where changes to Y are backwards
  // compatible, and changes to X are not.
  required string version = 1;

  // Version of the device's firmware.
  optional string device_firmware_version = 2;

  // URL to direct a user in need of technical support.
  optional string support_url = 3;

  // URL to direct a user to setup your device.
  optional string setup_url = 4;

  // Read-only data that can be used by vendors.
  repeated VendorProperty vendor_property = 100;

  // Special vendor-specific capabilities that are not available in any of the
  // semantic sections of the CDD.
  repeated VendorCapability vendor_capability = 101;

  // Section of the CDD that specifically describes printers.
  optional PrinterDescriptionSection printer = 102;

  // Section of the CDD that specifically describes scanners.
  optional ScannerDescriptionSection scanner = 103;
}

// Section of a CDD that describes the capabilities of a cloud-connected
// printer.
message PrinterDescriptionSection {
  // Content types (sometimes referred to as MIME types) that are supported by
  // the printer.
  optional SupportedContentType supported_content_type = 1;

  // Printing speeds that the printer can operate at.
  optional PrintingSpeed printing_speed = 2;

  // PWG raster configuration of the printer. Only set this if the printer
  // supports image/pwg-raster content type. This allows a cloud service to
  // understand how to rasterize a document for the printer.
  optional PwgRasterConfig pwg_raster_config = 3;

  // What about PwgRasterDocumentResolution and PwgRasterDocumentType capabilities?
  // PWG Raster type strings: "black-1", "srgb-8", "cmyk-8", etc.
  optional repeated string pwg_raster_type = 112;
  // PWG Raster resolutions that are supported; this is separate from the devie
  optional repeated Dpi pwg_raster_resolution = 113;
  // Color printing capabilities of the printer.
  optional Color color = 100;

  // Duplexing capabilities of the printer.
  optional Duplex duplex = 101;

  // Manual duplex is a separate capability; note that using manual duplex
  // requires additional capability information to know what order pages
  // must be printed and to provide instructions to the user for printing
  // the second time.  Might want a string or dedicated type instead to
  // describe how manual duplex printing works.
  optional bool manual_duplex = 114;

  // Page/paper orientation capabilities of the printer.
  optional PageOrientation page_orientation = 102;

  // Multiple copy capability of the printer.
  optional Copies copies = 103;

  // Page margins capability of the printer.
  optional Margins margins = 104;

  // Printing quality or dots-per-inch (DPI) capabilities of the printer.
  optional Dpi dpi = 105;

  // Page fitting capabilities of the printer.
  optional FitToPage fit_to_page = 106;

  // Page range selection capability of the printer.
  // This is a boolean in PWG Semantic Model
  optional PageRange page_range = 107;

  // Page or media size capabilities of the printer.
  optional MediaSize media_size = 108;

  // Paper collation capability of the printer.
  optional Collate collate = 109;

  // Reverse order printing capability of the printer.
  optional ReverseOrder reverse_order = 110;

  // Automatic page rotation capability of the printer.
  // This is a function of the orientation element in the PWG Semantic Model
  optional RotateToPage rotate_to_page = 111;
}

// Property that defines what content types the printer can print natively.
message SupportedContentType {
  message Option {
    // Content type (e.g. "image/png" or "application/pdf"). Use “*/*”, if your
    // printer supports all formats.
    required string content_type = 1;

    // Numeric rank used to order content types. Content types with lower rank
    // rank will be used before those with higher rank. Minimum rank should be
    // 1.
    required int32 rank = 2;

    // Minimum supported version of the content type if applicable (e.g. "1.5").
    optional string min_version = 3;

    // Maximum supported version of the content type if applicable (e.g. "1.5").
    optional string max_version = 3;
  }

  repeated Option option = 1;
}

// Property that defines what speeds (in pages per minute) the printer can
// operate at.
// PWG Semantic Model, Printer MIB, etc. only define two values - maximum speed in
// grayscale and color
message PrintingSpeed {
  // Available speed of the printer.
  //
  // Specify settings that are associated with the given speed. If a setting
  // is left unset, then it will be assumed that the speed is independent of
  // that setting. For example, the following Option
  //
  //   {
  //     “speed_ppm”: 5.5,
  //     “color_type”: [1 /*STANDARD_MONOCHROME*/],
  //     “dpi_type”: [],
  //     “media_size_type”: [8 /*LETTER*/, 6 /*A4*/]
  //   }
  //
  // indicates that the printer prints at 5.5 pages per minute when printing in
  // MONOCHROME in either LETTER or A4 paper sizes, but does not depend on any
  // particular print quality.
  message Option {
    // Speed measured in pages per minute.
    required float speed_ppm = 1;

    // Types of color settings that operates at this speed.
    repeated Color.Type color_type = 2;

    // Types of print quality settings that operates at this speed.
    repeated Dpi.Type dpi_type = 3;

    // Types of color settings that operates at this speed.
    repeated MediaSize.Type media_size_type = 4;
  }

  // Speeds that the printer can operate at.
  repeated Option option = 1;
}

// Configuration of how printer should receive PWG raster images.
// This is not consistent with the PWG definition of pwg-raster-document-sheet-back;
// the type can represent 48 different values, but the PWG value this is derived from
// only has 4 possible values.  Similarly, CUPS has the same 4 values for CUPS Raster,
// which is the parent format of PWG Raster:
enum PwgRasterConfig {
  NORMAL = 0;
  MANUAL_TUMBLE = 1;
  ROTATE = 2;
  FLIP = 3;
}
```

~~message PwgRasterConfig {~~
~~// Type of flip to perform on a page.~~
~~enum FlipType {~~
~~NONE = 0;~~
~~LONG_EDGE = 1;~~
~~SHORT_EDGE = 2;~~
~~}~~

~~// Whether this printer accepts all of its pages rotated by 180 degrees.~~
~~// Ex. 1' 2' 3' 4' where ' means rotated.~~
~~optional bool rotate_all_pages_180 = 1 [default = false];~~

~~// Whether this printer accepts even pages rotated by 180 degrees when~~
~~// printing in duplex. Ex. 1 2' 3 4' where ' means rotated.~~
~~optional bool rotate_even_pages_180_for_duplex = 2 [default = false];~~

~~// Whether this printer accepts even pages flipped when printing in duplex.~~
~~// Ex. 1 2^ 3 4^ where ^ means flipped.~~
~~optional FlipType flip_even_pages_for_duplex = 3 [default = NONE];~~

~~// Whether this printer accepts printing even pages before odd pages when~~

```
  ~~// printing in duplex. Ex. 2 1 4 3.~~
  ~~optional bool print_even_page_first_for_duplex = 4 [default = false];~~

  ~~// Whether this printer needs pages flipped before being rotated when~~
  ~~// printing in duplex. This only applies if both flip_even_pages_for_duplex~~
  ~~// and rotate_even_pages_180_for_duplex are both true.~~
  ~~optional bool flip_first_then_rotate_for_duplex = 5 [default = false];~~
~~}~~

// Capability that defines a set of duplexing options available on a device.
message Duplex {
  enum Type {
    NO_DUPLEX = 0;
    LONG_EDGE = 1;
    SHORT_EDGE = 2;
    // This needs to be a separate value:
    //
    // 1. Manual duplex is a printer capability - i.e. you ask to print duplex, not
    //    auto vs. manual
    // 2. Manual duplex can be short or long edge (rotation of back side image)
    // 3. Manual duplex is not part of the PWG Semantic Model and needs more info
    //    to rasterize pages in the right order and supply instructions to the user...
    ~~MANUAL_DUPLEX = 3;~~
  }

  message Option {
    required Type type = 1;
    optional bool is_default = 2 [default = false];
  }

  repeated Option option = 1;
}

// Capability that defines a set of page-orientations options available on a
// device.
message PageOrientation {
  // PWG Semantic Model has a 'none' value; or if orientation is unspecified in the
  // job ticket then the printer can automatically rotate as needed.
  enum Type {
    PORTRAIT = 0;
    LANDSCAPE = 1;
    REVERSE_PORTRAIT = 2;
    REVERSE_LANDSCAPE = 3;
    NONE = 4;
  }

  message Option {
    required Type type = 1;
    optional bool is_default = 2 [default = false];
  }

  repeated Option option = 1;
}

// Capability that defines a default and maximum value for multiple copies on a
// device.
// How does a printer indicate they only support copies for specific formats?
message Copies {
  optional int32 default = 1;
  optional int32 max = 2;
}

// Capability that defines a set of margins available on a device (including a
// custom one). Margins are measured in microns.
// PWG Semantic Model ties a set of margin values to the size, type, source, etc.
// associated with the margin values.
message Margins {
  // PWG Semantic Model has no notion of a margin type; rather, margins are associated
  // with their corresponding sizes since margins often vary based on the size or
  // source/input tray...
  enum Type {
    BORDERLESS = 0;
    STANDARD = 1;
    // DUPLEX???
    CUSTOM = 2;
  }

  message Option {
    required Type type = 1;
    required int32 top_microns = 2;
    required int32 right_microns = 3;
    required int32 bottom_microns = 4;
    required int32 left_microns = 5;
    optional bool is_default = 6 [default = false];
  }

  repeated Option option = 1;
  optional int32 min_top_microns = 2;
  optional int32 min_right_microns = 3;
  optional int32 min_bottom_microns = 4;
  optional int32 min_left_microns = 5;
}

// Capability that defines a set of page fitting options available on a device.
// What about scale-to-fill for borderless printing?
message FitToPage {
  enum Type {
    NO_FITTING = 0;
    FIT_TO_PAGE = 1;
    GROW_TO_PAGE = 2;
    SHRINK_TO_PAGE = 3;
  }

  message Option {
    required Type type = 1;
    optional bool is_default = 2 [default = false];
  }

  repeated Option option = 1;
}

// Whether to rotate pages to best fit a media size. This is useful for
// documents that have pages in different orientations.
// See page orientation - this should be a 'none' or 'auto'/unspecified value instead
// of a separate property.
message RotateToPage {
  // Whether to automatically rotate pages by default.
  required bool default = 1;
}

// Capability that defines a default page-range selection on a device.
message PageRange {
  // Interval of pages in the document to print.
  message Interval {
    // Beginning of the interval (inclusive).
    required int32 start = 1;

    // End of the interval (inclusive). If not set, then the interval will
    // include all available pages after start.
    optional int32 end = 2;
  }

  repeated Interval default = 1;
}

// Capability that defines the default collation setting on a device.
message Collate {
  required bool default = 1;
}
```

```
// Capability that defines the default reverse-printing-order setting on a device.
message ReverseOrder {
  required bool default = 1;
}

// Section of a CJT of how a print job should be handled by a cloud-connected
// printer.
// What about print quality? More useful/descriptive than just DPI...
message PrinterTicketSection {
  optional ColorTicketItem color = 1;
  optional DuplexTicketItem duplex = 2;
  optional PageOrientationTicketItem page_orientation = 3;
  optional CopiesTicketItem copies = 4;
  optional MarginsTicketItem margins = 5;
  optional QualityTicketItem quality = 13;
  optional DpiTicketItem dpi = 6;
  optional FitToPageTicketItem fit_to_page = 7;
  optional PageRangeTicketItem page_range = 8;
  optional MediaSizeTicketItem media_size = 9;
  optional CollateTicketItem collate = 10;
  optional ReverseOrderTicketItem reverse_order = 11;
  optional RotateToPageTicketItem rotate_to_page = 12;
}

// Ticket item indicating the desired output quality
message QualityTicketItem {
  required Dpi.Type quality = 1;
}

// Ticket item indicating which duplexing option to use.
message DuplexTicketItem {
  required Duplex.Type type = 1;
}

// Ticket item indicating which page orientation option to use.
message PageOrientationTicketItem {
  required PageOrientation.Type type = 1;
}

// Ticket item indicating how many copies to produce.
message CopiesTicketItem {
  required int32 copies = 1;
}

// Ticket item indicating what margins to use (in microns).
message MarginsTicketItem {
  required int32 top_microns = 1;
  required int32 right_microns = 2;
  required int32 bottom_microns = 3;
  required int32 left_microns = 4;
}

// Ticket item indicating what page-fitting algorithm to use.
message FitToPageTicketItem {
  required FitToPage.Type type = 1;
}

// Ticket item indicating whether to automatically rotate pages.
// As noted previously, this is an orientation value in the PWG Semantic Model
message RotateToPageTicketItem {
  required bool rotate_to_page = 1;
}

// Ticket item indicating what pages to use.
message PageRangeTicketItem {
  repeated PageRange.Interval interval = 1;
}

// Ticket item indicating whether to collate pages.
message CollateTicketItem {
  required bool collate = 1;
}

// Ticket item indicating whether to print in reverse.
message ReverseOrderTicketItem {
  required bool reverse_order = 1;
}

// Capability that defines a set of color options available on a device.
message Color {
  // What about "AUTO"?
  enum Type {
    STANDARD_COLOR = 0;
    STANDARD_MONOCHROME = 1;
    CUSTOM_COLOR = 2;
    CUSTOM_MONOCHROME = 3;
    AUTO = 4;
  }

  message Option {
    // ID to help vendor identify the color option.
    required string vendor_id = 1;

    // Type of color option used in UI to differentiate color and non-color
    // options. Note there should only be at most one STANDARD_COLOR option, at
    // most one STANDARD_MONOCHROME, and any number of the CUSTOM_* options.
    required Type type = 2;

    // User-friendly string that represents this option. Options marked as
    // STANDARD_COLOR or STANDARD_MONOCHROME will have their display-name
    // localized, this field should be used for CUSTOM_* options.
    optional string custom_display_name = 3;

    // Whether this option should be selected by default. Only one option
    // should be set as default.
    optional bool is_default = 4 [default = false];
  }

  repeated Option option = 1;
}

// Capability that defines a set of 2D image quality levels available on a
// device.
message Dpi {
  // PWG Semantic Model Print Quality values are: Draft, Normal, Best
  enum Type {
    CUSTOM = 0;
    DRAFT = 1;
    NORMAL = 2;
    ~~PHOTO = 3;~~
    BEST = 4;
  }

  message Option {
    required Type type = 1;
    required int32 horizontal_dpi = 2;
    required int32 vertical_dpi = 3;
    optional bool is_default = 4 [default = false];
  }

  repeated Option option = 1;
  optional int32 min_horizontal_dpi = 2;
  optional int32 max_horizontal_dpi = 3;
  optional int32 min_vertical_dpi = 4;
  optional int32 max_vertical_dpi = 5;
}

// Capability that defines a set of media sizes available on a device.
message MediaSize {
  // PWG Semantic Model, Printer MIB, IPP, etc. all use a string for the name.
  // Consider adopting PWG 5101.1 (Media Standardized Names)
  // Also, "Type" here has a different meaning than "MediaType" which normally
  // refers to media coating/characteristics like "Glossy", "Matte", "Plain", etc.
  // and not identification of a particular size.
  enum Type {
    CUSTOM = 0;
    A0 = 1;
    A1 = 2;
    A2 = 3;
    A3 = 4;
    A3_PLUS = 5;
    A4 = 6;
    A5 = 7;
    LETTER = 8;
    LEGAL = 9;
    LEDGER = 10;
  }

  message Option {
    required Type type = 1;
    required string name = 6;
    optional int32 width_microns = 2;
    optional int32 height_microns = 3;
    optional bool is_continuous_feed = 4;
    optional bool is_default = 5 [default = false];
  }

  repeated Option option = 1;
  // Also need minimum width/height
  optional int32 max_width_microns = 2;
  optional int32 max_height_microns = 3;
  optional int32 min_width_microns = 4;
  optional int32 min_height_microns = 5;
}

// Ticket item indicating which color option to use.
message ColorTicketItem {
  required string vendor_id = 1;
  required Color.Type type = 2;
}

// Ticket item indicating what image resolution to use.
message DpiTicketItem {
  required int32 horizontal_dpi = 1;
  required int32 vertical_dpi = 2;
}

// Ticket item indicating what media size to use.
message TicketItem {
  required int32 width_microns = 1;
  required int32 height_microns = 2;
}
```