

```

%!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% NeWS Forth HyperTIES PostScript stuff.
% PostScript function definitions for fmt.cps interface.
% Don Hopkins
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

systemdict /FmtDict known not {
  systemdict /FmtDict 200 dict put
} if
FmtDict begin

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% PostScript interface to fmt.cps
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% do_find_font(name, size)
/FF { % name size => -
%(do_find_font % %\n) [3 index 3 index] dbgprintf
  /FontSize exch store
  /FontName exch store
  {FontName findfont} errored {
    pop
    /Font null store
    2 % NO_FONT_TAG
    tagprint
  } {
    FontSize scalefont
    /Font exch store
    Font setfont
    3 % FOUND_FONT_TAG
    tagprint
  } ifelse
  polite
} def

% do_use_font(name, size)
/UF { % name size
%(do_use_font % %\n) [3 index 3 index] dbgprintf
  /FontSize exch store
  /FontName exch store
  /Font FontName findfont FontSize scalefont store
  % Optimize out useless setfonts!
  count 0 ne { dup /setfont load eq {pop pop} if } if
  Font /setfont load
} def

% do_get_metrics(height, ascent, descent)
/GMDict 10 dict def
/GM {
%(do_get_metrics\n) [] dbgprintf
  Font setfont
  1 % FONT_METRICS_TAG
  tagprint

```

```

Font fontheight cvi typedprint
Font fontascent cvi typedprint
Font fontdescent cvi typedprint
//GMDict begin
  0 1 255 { /i exch store
    (-) dup 0 i put stringwidth pop
    cvi typedprint
  } for
end
polite
} def

% do_get_rect(width, height)
/GR { % name id => -
  gsave
    fboverlay setcanvas
    [ getwholerect waitprocess aload pop points2rect ]
  grestore
  4 % GET_RECT_TAG
  tagprint {typedprint} forall
  polite
} def

% do_make_pile(class, id, x, y, w, h)
/MP { % x y w h class id => -
  systemdict /CP currentprocess put % XXX
  systemdict /CF currentfile put % XXX
  % Make new pile's parent the current pile, or itself if no current pile
  currentdict /PileID known { PileID end } { dup } ifelse
  /Pile 128 dict store
  Pile begin
    /ParentPileID exch def
    /PileID exch def
    PileDict PileID Pile put
    /DefinedTID null def
    /Page null def
    cvx exec
    dup type /integertype eq { % Emacs frame ID?
      get-emacs-dsp
      dup null eq { /NoEmacs dbgbreak } if
    } {
      framebuffer /new 3 -1 roll send
    } ifelse
    /Win exch def
    PileID
    { /WinPileID exch def
      reshape
      map
      ClientCanvas
    } Win send
    /Can exch def
    Can setcanvas
    clippath pathbbox
    /PageHeight exch def /PageWidth exch def
    pop pop
    % Maybe we should register the background ClientCanvas here?
    newprocessgroup
    polite
  end
} def

```

```

} def

% do_use_pile(id)
/UP { % id => -
    currentdict /PileID known { end } if
    PileDict exch get /Pile exch store
    Pile begin
    Can type /canvastype eq {Can setcanvas} if
} def

% do_set_pile_name(name)
/SN { % name => -
    /set-name Win send
} def

% do_def_target(string class, string name, int id)
/DT { % args init, class name id => -
    FilePos /new TargetStamp send % obj
    StampDict exch dup /ID get exch % dict id obj
    put %
    polite
} def

% do_put_target(id, string ref, x, y, width, height)
/PT { % x y width height ref id => -
    StampDict exch get
    /compile exch send
    polite
} def

% do_def_file_pos(string dir, string file, offset, len)
/DFP { % [dir file offset len] => -
    /FilePos exch def
} def

% do_def_picture(class, name, id)
/DP { % name id class => -
    cvn load FilePos exch /new exch send % obj
    StampDict exch dup /ID get exch % dict id obj
    put %
    polite
} def

% do_put_picture(id, x, y)
/PP { % x y id => -
    %(do_put_picture %\n) [4 index] dbgprint
    StampDict exch get
    /compile exch send
    polite
} def

% do_measure_stamp(id, width, height)
/MS { % id => -
    StampDict exch get
    { /size exch send } errored {0 0} if
    6 % MEASURE_STAMP_TAG
    tagprint exch typedprint typedprint
} def

```

```

% do_put_string(s, x, y)
/PS { % string x y => -
%(do_put_string % % %\n) [3 index 3 index 6 index] dbgprintf
  /moveto cvx
  4 -1 roll
  /show load
  pause
} def

% do_get_page_size(width, height)
/GPS {
  5 % PAGE_SIZE_TAG
  tagprint
  PageHeight typedprint PageWidth typedprint
} def

% do_start_page()
/SP {
%(do_start_page\n) [] dbgprintf
  polite
  /Can /new-canvas Win send def
  Can setcanvas
  /ClientFillColor Win send fillcanvas
  clippath pathbbox points2rect
  /PageWidth exch def /PageHeight exch def
  pop pop
  /Page 20 dict def
  Page /Targets 200 dict put
  Win /PageDict get Can Page put
  [ /ClientFillColor cvx /fillcanvas cvx
    /textcolor cvx /setcolor load
  ] def
} def

% do_end_page()
/EP {
%(do_end_page\n) [] dbgprintf
  ] cvx
  activate-page
  userdict /Can null put
  userdict /Page null put
  Win /ClientCanvas get setcanvas
} def

% do_zap_pages()
/ZP {
  /zap-pages Win send
} def

% do_start_line()
/SL {
  [
} def

% do_end_line(x, y, w, h, name, pos, len)
/EL { % ... x y w h name pos len => ...
  pop pop pop % later: make line selectable, w/ pointer back to storyboard
  % (extend left and right edges of region past margins)

```

```

    dup 0 eq {pop pop pop pop} {rectpath stroke} ifelse
  ] cvx
  dup length 0 eq {pop} {/exec load} ifelse
} def

% do_free_stamps()
/FS {
  [ StampDict {pop} forall ] { StampDict exch undef } forall
} def

% do_setup_definition_pile()
/SDP {
  /DefinitionPileID PileID def
  PileDict ParentPileID get /DefinitionPileID PileID put
} def

% do_setup_controls_pile()
/SCP {
  /ControlsPileID PileID def
  PileDict ParentPileID get /ControlsPileID PileID put
} def

% do_setup_contents_pile()
/SAP {
  /ContentsPileID PileID def
  PileDict DefinitionPileID get /ContentsPileID PileID put
  PileDict DefinitionPileID get /ControlsPileID ControlsPileID put
  PileDict ControlsPileID get /ContentsPileID PileID put
  PileDict ControlsPileID get /DefinitionPileID DefinitionPileID put
} def

% do_name_pile(name)
/NP { % name => -
  cvn userdict exch PileID put
} def

% do_use_linked_pile(name, id)
/ULP { % name => -
  cvn load dup UP
  7 % USE_LINKED_PILE_TAG
  tagprint
  typedprint
} def

% do_link_parent_pile(name)
/LPP { % name => -
  PileDict ParentPileID get
  exch cvn PileID put
} def

% do_use_parent_pile(id)
/UPP {
  ParentPileID dup UP
  8 % USE_PARENT_PILE_TAG
  tagprint
  typedprint
} def

```

```

% do_def_local(name)
/DFL { % name => -
  File exch cvn LastTID put
} def

% do_def_global(name)
/DFG { % name => -
  userdict exch cvn LastTID put
} def

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Utilities
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

UserProfile /DoubleClickTime known {
  UserProfile /DoubleClickTime get
} { 8 60 div } ifelse
/DoubleClickTime exch def

/activate-page { % DL => -
  Page /Can Can put
  Page /DL 3 -1 roll put
  systemdict /DL Page /DL get put % XXX
  % Fork an event manager for the target items
  [ Page /Targets get {pop TargetDict exch get} forall ] iteminterests
  [ exch aload pop
    /Damaged
      {CurrentEvent /Canvas get /damage-paint-canvas Win send}
      null Can eventmgrinterest
  ] forkeventmgr
  Page /ItemMgr 3 -1 roll put
  newprocessgroup
  Can setcanvas
  clippath extenddamage
  polite
} def

/send-to-hookfile {
  hookfile exch writestring hookfile flushfile
} def

systemdict /sendtoemacs known {
  /ties-key-prefix ($[T) def
  ties-key-prefix 0 27 put

  /send-to-ties {
    ties-key-prefix exch append sendtoemacs
  } def

  /get-emacs-dsp { % n => dsp
    {
      createevent begin
        /Name [/ReplyEmacsFrame /TimeoutEmacsFrame] def
        currentdict end expressinterest
        createevent begin
          /Name /RequestEmacsFrame def

```

```

    /ClientData exch def
    currentdict end sendevent
    createevent begin
        /Name /TimeoutEmacsFrame def
        /TimeStamp currenttime 5 60 div add def
        /ClientData null def
        currentdict end sendevent
        awaitevent /ClientData get
        {/frameDsp get} errored {null} if
    } fork waitprocess exch pop
} def

} {
    /send-to-ties {
        send-to-hookfile
    } def
} ifelse

/callback {
    { send-to-ties } errored pop
} def

/rr {
    -2 ItemHeight .3 mul
    0 0 ItemWidth ItemHeight insetrrect rrectpath
} def

/rrp { % x y w h => -
    { dup .3 mul 5 1 roll rrectpath } unit-scale
} def

/findraster { % name => -
    dup 0 get dup 47 ne exch 33 ne and { % does not start with / or !
        TiesRootDirectory exch append
    } if
    RasterDict 1 index cvn known {
        RasterDict exch cvn get
    } {
        dup { readcanvas } errored {
            pop pop null
        } {
            RasterDict 3 -1 roll % can RD name
            2 index put % can
        } ifelse
    } ifelse
} def

/here { % file => ./file
    TiesRootDirectory FilePos 0 get append exch append
} def

systemdict begin
/shit! {
    globalinterestlist {dup /Action get /UpTransition ne {pop} {
        /Process get killprocess
    } ifelse } forall
} def
end % systemdict

```

```
end % FmtDict
```