

/*****

Program: pdf_inc.i

Written By: Gordon Campbell - PRO-SYS Consultants Ltd.
Written On: June 2002

Description: Contains function and variable definitions for
generating a PDF document from within Progress

Note: This can only be included once per program

----- Revision History -----

Date:	Author	Change Description
07/12/02	G Campbell	Initial Release
09/05/02	G Campbell	Fixed issue regarding the first call to pdf_set_font - wasn't actually setting the font, had to be called twice before accepting changes Fixed annoying 'rebuild' message - caused by inappropriate offset values when producing XREF table
09/10/02	G Campbell	Due to previous XREF changes, the pdf_load_image and pdf_load_images functions had to change also
10/14/02	G Campbell	Changed the element setting functions to procedures - older versions of Progress were reaching the 64K Segment issue.
10/18/02	G Campbell	Added routine called pdf_replace_text and call from appropriate text procedures. Used to handle special characters in text strings.
10/22/02	G Campbell	As per Steven L. Jellin (sjellin@elcb.co.za) Added two procedure pdf_reset_all and pdf_reset_stream.
11/04/02	G Campbell	As per Julian Lyndon-Smith (jmls@tessera.co.uk) Modified the Font/Image import procedures to use the OS-APPEND command.
02/28/03	G Campbell	As per SourceForge Bug 694886 - Adam Davies (adam.davies@travellink.com.au) The basel4 Font "Helvetica" is missing the width of a character. This means all lookups into the array are out by one (after about "D" in the ASCII chars).

This causes calls to pdf_text_width to be quite a long way out.

02/28/03 G Campbell

As per SourceForge Bug 694888 - Adam Davies
(adam.davies@travellink.com.au)

Function pdf_text_width in pdf_inc.i is supposed to return the width of the passed in text parameter based on the current font.

It does this by summing each characters width as defined in an Adobe AFM file for the font (or hard-coded for base14 fonts), then dividing by 1000 and multiplying by the point size.

However, rounding is done via INTEGER conversion after the first division instead of after the multiplication resulting in poor accuracy.

02/28/03 G Campbell

As per SourceForge Bug 695454 - Steve Southwell
(ses@bravepointdallas.com)

In pdf_set_Orientation, the pdf stream name was hard-coded to "Spdf.

02/28/03 G Campbell

As per Steve Southwell
(ses@bravepointdallas.com)

Added procedure pdf_text_center. This centers text on a certain X,Y point

03/02/03 G Campbell

As per Adam Davies
(ses@bravepointdallas.com)

Added function pdf_get_NumFittingChars. This function returns the index of the last character that will fit into the specified width. It does this by summing each characters AFM width (as specified in the tt_pdf_font.font_width array) and comparing with the required width (converted into these same units).

03/12/03 G Campbell

Added procedure pdf_link to allow for creation of document links.

03/19/03 G Campbell

As per Steve Southwell
(ses@bravepointdallas.com)

Added procedure pdf_wrap_text to wrap text within a specific column range. Returns a parameter that tells you the last Y position used.

Added function pdf_get_wrap_length.
You can use the function to see how long a piece of
text WOULD be if you were to wrap it.

03/29/03 G Campbell

As per Kim Allbritain
(kimatrr@tampabay.rr.com)

pdf_text_width was always returning 0 (zero) when a
loaded (non Base) font was being used. To fix this
issue I now run the pdf_ParseAFMFile routine as soon
as pdf_load_font is called. This also required
additional field requirements to TT_pdf_font.

06/23/03 G Campbell

Added ability to define a Rotation Angle. Angles
available are 0,90,180 and 270. Also updated code
to ensure that TextX and TextY were being set
correctly whenever pdf_Text was being created (based
on current angle). Reworked placement of routines.

07/02/03 G Campbell

Updated pdf_new_page to include calls to reset
Angle and Text X/Y coordinates.

08/18/03 G Campbell

Added pdf_set_BottomMargin and pdf_BottomMargin.
This allows us to define a bottom most point when
creating a document. If any text goes below the
Bottom Margin value (eg: 50) a new page is
automatically created. This is useful for
converting text documents that don't have any page
markers (eg: ASCII chr(12)).

Also, modified pdf_set_TextY to use the BottomMargin
to determine if any text elements have (or are)
going below the Bottom Margin value. If so, create
a new page.

08/18/03 G Campbell

As per Mike Frampton
(MikeF@sequoiains.com)

Added pdf_watermark procedure. Allows you to create
a text watermark (eg: Invoice) that appears on the
first layer of the PDF --- below any rectangles,
images etc. Use with caution as you may not get the
complete watermark if you overlay with images.

Sample Call:

```
RUN pdf_watermark ("Spdf",  
                  "SAMPLE REPORT",  
                  "Courier-Bold",  
                  34,  
                  .5,.5,.14,  
                  300,500).
```

08/18/03	G Campbell	<p>As per Herbert Bayer (I think) Herbert Bayer [Herbert.Bayer@bundrinno.de]</p> <p>Added pdf_skipn. This allows you to skip 'n' number of lines. Saves calling pdf_skip numerous times.</p>
08/25/03	G Campbell	<p>Added pdf_TotalPages function.</p> <p>This allows you to include the Total Number of Pages for the document directly into your report. For instance, if you wanted to have 'Page n of x' as your Page footer then you can implement this with pdf_TotalPages(streamname).</p>
08/26/03	G Campbell	<p>Added PageHeader functionality.</p> <p>Similar to the PageFooter functionality previously outlined but for the top of the page.</p>
08/26/03	G Campbell	<p>Updated pdf_xref to use PUT CONTROL and CHR(13) and CHR(10).</p> <p>This is due to compatability issues between UNIX and Windows implementations. Each Xref line must be a certain number of characters and if we use SKIP it will differ based on OS.</p>
08/27/03	G Campbell	<p>Updated pdf_line to include a decimal value for the line thickness.</p>
09/02/03	G Campbell	<p>Updated text drawing components to use pdf_TextY to determine the appropriate line number. Removed use of pdf_CurrentLine. This was causing issues when working with PageFooter.</p>
09/02/03	G Campbell	<p>Started playing with proportional fonts and text placement today. Fixed some issues that were found hoping to accommodate the ability to place proportional fonts correctly. Fixed pdf_text_to to place appropriately but you may notice that the placement seems 'off' compared to Fixed fonts. This is due to the fact that the placement is calculated as -- (Column Position * Width of CHR(32)) minus the pdf_text_width("of appropriate text"). This gives us a placemnt to work to. Also, fixed pdf_text_width to use the entry + 1 to determine the font width -- this is because there is a zero entry in the AFM File which doesn't coincide with ASC value of a character.</p>
09/02/03	G Campbell	<p>Added function pdf_ImageDim to retrieve appropriate dimension values. Valid Dimension input values are</p>

"HEIGHT" and "WIDTH". Return 0 if can't find Image name. This is helpful when placing a rectangular box around an image.

09/03/03	G Campbell	As per John Stonecipher (jstone@inpac.com) Added 'Line Weight' parameter to pdf_rect. This allows you to modify the line drawing weight of a rectangle.
09/03/03	G Campbell	Ensured that all calls to pdf_error had the correct funtion or procedure name. This is so that debugging is easier.
09/03/03	G Campbell	Updated pdf_wrap_text to see if any CHR(13) were included in the character string. If so, then automatically do a pdf_skip in the wrap procedure.
09/09/03	G Campbell	Added an extra parameter to pdf_load_font procedure. The additional parameter is used to define character differences. That is, it allows you to re-map specific characters in the AFM file to something else. The format of the DIF file should be: <character number><space>/<PS Character Name><skip> eg: 218 /SF010000 This would change character 218 (Uacute in a cour.afm) file to the Postscript character SF010000 (which is "box drawings light down and right"). These codes can be found at: http://partners.adobe.com/asn/tech/type/opentype/appendices/wgl4.jsp Also, created pdf_font_diff procedure so that you can dynamically re-map characters.
09/09/03	G Campbell	Added pdfSKIP preprocessor. This sets the SKIP to include the appropriate characters when compiled on a given OPSYS. This is to help overcome the OS compatability issues when generating PDFs.
10/14/03	G Campbell	Updated pdf_replace_text to handle back slashes in text fields.
10/28/03	G Campbell	Added logic to determine whether a text line could possibly include the TotalPages function. Hopefully this will help increase the processing speed.
11/10/03	G Campbell	Copied pdf_inc.i to pdf_inc.p and reworked pdf_inc.i

to call pdf_inc.p Persistently

pdf_inc.i now also has single argument

{1} - can be nothing, SUPER, or THIS-PROCEDURE

If SUPER or THIS-PROCEDURE then pdf_inc.p is added as a Super Procedure appropriately.

If nothing is specified then pdf_inc.p is run persistently. All procedural calls must now contain IN h_PDFinc. eg:

RUN pdf_text IN h_PDFinc ("Spdf","Sample").

*****/

DEF VAR h_PDFinc AS HANDLE NO-UNDO.

/* Call pdf_inc.p Persistently */

RUN pdf_inc.p PERSISTENT
SET h_PDFinc
(INPUT THIS-PROCEDURE:HANDLE) .

&IF "{1}" = "SESSION" &THEN

IF VALID-HANDLE(SESSION) THEN SESSION:ADD-SUPER-PROCEDURE(h_PDFinc).

&ELSEIF "{1}" = "THIS-PROCEDURE" &THEN

IF VALID-HANDLE(THIS-PROCEDURE) THEN THIS-PROCEDURE:ADD-SUPER-PROCEDURE(h_PDFinc).

&ENDIF

/* ----- Pre-Define Functions ----- */

FUNCTION pdf_Font RETURN CHARACTER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_FontType RETURN CHARACTER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_ImageDim RETURN INTEGER (INPUT pdfStream AS CHARACTER,
INPUT pdfImage AS CHARACTER,
INPUT pdfDim AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TextX RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TextY RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_VerticalSpace RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_PointSize RETURN DECIMAL (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_text_width RETURNS INTEGER (INPUT pdfStream AS CHARACTER,
INPUT pdfText AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TextRed RETURN DECIMAL (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TextGreen RETURN DECIMAL (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TextBlue RETURN DECIMAL (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_Page RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_PageWidth RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_Pageheight RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_Angle RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_TopMargin RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_BottomMargin RETURN INTEGER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.

```

FUNCTION pdf_GraphicX RETURN INTEGER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_GraphicY RETURN INTEGER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_get_info RETURNS CHARACTER ( INPUT pdfStream      AS CHARACTER,
                                           INPUT pdfAttribute AS CHARACTER) IN h_PDFinc.

FUNCTION pdf_LeftMargin RETURN INTEGER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_get_numFittingChars RETURNS INTEGER
      ( INPUT pdfStream      AS CHARACTER,
        INPUT pdfText        AS CHARACTER,
        INPUT pdfRequiredWidth AS INTEGER ) IN h_PDFinc.

FUNCTION pdf_Orientation RETURN CHARACTER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_PaperType RETURN CHARACTER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_Render RETURN INTEGER ( INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_get_wrap_length RETURNS INTEGER ( INPUT pdfStream      AS CHARACTER,
                                              INPUT pdfText AS CHARACTER,
                                              INPUT pdfWidth AS INTEGER ) IN h_PDFinc.

FUNCTION pdf_TotalPages RETURN CHARACTER (INPUT pdfStream AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_PageFooter RETURN LOGICAL (INPUT pdfFooterProc AS CHARACTER) IN h_PDFinc.
FUNCTION pdf_PageHeader RETURN LOGICAL (INPUT pdfHeaderProc AS CHARACTER) IN h_PDFinc.

```

```

/* ----- End of Pre-Define Functions ----- */

```