Lastic Document Writer

V2.2

User Guide

Introduction

The Lastic Document Writer (LDW) allows formal reports to be produced to printers, PDF files or PostScript files. It is intended to operate on a Windows™ server where it can be shared by various applications.

It permits a number of streams (or routes) to be maintained with each route identifying a specific printer or a directory to receive the resultant files.

Where PDF and PostScript (PS) files are produced, they are generated with the same root filename as the source file but with a PDF or PS extension. This enables the generating applications to specifically name the required output file without any operator intervention – unlike many PDF generating tools.

Reports may contain element types of:

Text
- Each element:
  - is assigned font details (name, size, style, colour)
  - may be single or multi-line (which is word-wrapped by the LDW automatically)
  - aligned, within the defined area, Left, Right or Centre
  - may have a frame of specified thickness and colour
  - may have a background colour

Table
- Each element:
  - is assigned font details (name, size, style, colour)
  - has individual column alignment (Left or Right)
  - may have a frame of specified thickness and colour
  - may have a background colour
  - may have grid lines (vertical and/or horizontal)
  - has individual cell text font style and colour override

Image
- Each Element:
  - may be a Bitmap, WMF or EMF file
  - is automatically re-sized to fit the defined area.
  - may be specific to this document or may be a ‘library’ image.

Lines (vertical or horizontal) may be drawn using the Text control without any presented text and setting the appropriate top, left, right and bottom positions.

The document source file is a well formed XML file containing the drawing specification, texts and references to image files.

Contents

1. Installation

1.1 Loading the routines

Copy ldw.exe and this user guide to a new directory.

Note: ldw.exe will also create ldwctr.ini to hold the set up details and ldw.log which records all submission details and the action results (including any that failed).

1.1.1 PostScript

For generation of postscript and PDF files, Lastic Document Writer utilises the driver of a postscript printer.

If you have no PostScript printer attached to (or accessible from) this server then create a logical printer and load the postscript driver for that printer.

1.1.2 PDF file Generation

Lastic Document Writer uses GhostScript or an equivalent mechanism. It is not “bound” within this product but is invoked via a command thus, with the various parameter definitions (see LDW Set up below) enable alternative mechanisms to be used.
Ghostscript is available as a free download from various web sites including GNU and Sourceforge to create PDF files from postscript files.

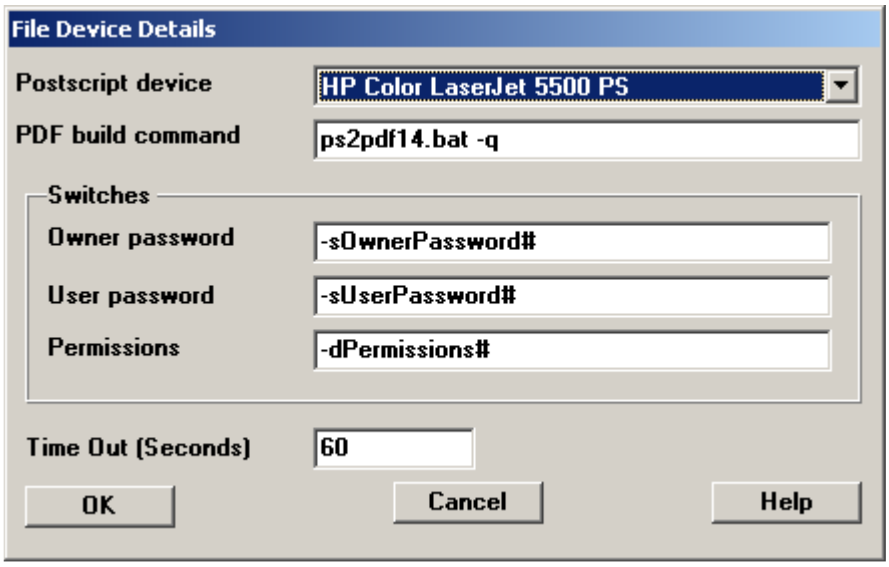
Down load the Windows version and unzip to the default directories.

Amend the system Path variable to include any of the required converter directories (Bin & Lib for Ghostscript).

For Ghostscript, this product has been tested with release gs8.60.

1.2 LDW Set up

Start up ldw.exe (double click on the ldw.exe file or set up a shortcut/menu entry). The main form (as shown in the following section) appears.
Now select File Device Details from the menu. This invokes the form below.



Postscript Device
Select the postscript printer referred to above from the drop down list. The driver setup must select “Print to File” as the port, winprint as the print processor and raw as the data type. This “printer” wil be used for generation of postscript files either as the final output or as an intermediate file for subsequent PDF file generation.

PDF build command
Enter the command file and any basic parameters (switches) for the conversion utility required. The above example shows the GhostScript postscript to PDF converter command. This should be the .bat file.

Switches
These are defined to enable the resultant PDF file to have limited user access (via passwords and permissions). Enter the switch Identifier for each of the above (if required). If no such switch options are to be provided (even if the incoming document requests it) then leave these switch settings blank.
If entered then they must correspond to the converter's exact specification. The examples above relate to the use of Ghostscript.

Time Out (Seconds)
The Postscript printer driver and PS to PDF converter routines operate asynchronously. Ldw.exe checks that the postscript file has been generated before calling converter routine to create a PDF file. It also checks that the PDF file has been created.
For each of these files, if a problem has occurred in their creation it can only be identified by the failure of ldw.exe to have access to that task's completed file. Ldw.exe checks for such conditions. This Time Out period is the maximum time ldw.exe should wait for completion of the job.

Enter the maximum number of seconds to allow the postscript driver or PS to PDF converter routine to generate a file. It is assumed that, if the file has not been completely created in this time then the conversion has failed and that file is to be abandoned. Such an error will be notified on the LDW main form and logged to ldw.log.

Click the OK button to save the changes. If changes have been made LDW will require an auto restart for those changes to take effect.

Once the above has been completed, the remainder of the set up (i.e. the routing) can now be arranged.

1.3 New Features in this version

Measurements
All measurements may now be given in decimal fractions of a millimetre – e.g. 1.0125 millimetres. This applies to all positioning and line thickness values.

Text controls
These now may have multiple embedded font changes and CDATA sections as well as PCDATA. Text of the same font is presented as one or more sequential sub-nodes of the TextLine element (PCDATA or CDATA) and Font elements may occur at the appropriate position to indicate changes of font.
Text wrapping has been upgraded to work directly with the printer drivers.
BackGround colour attribute has been added
Paragraph termination may now be expressed as either <CR><LF> or ASCII(127) characters within the TextLine data.

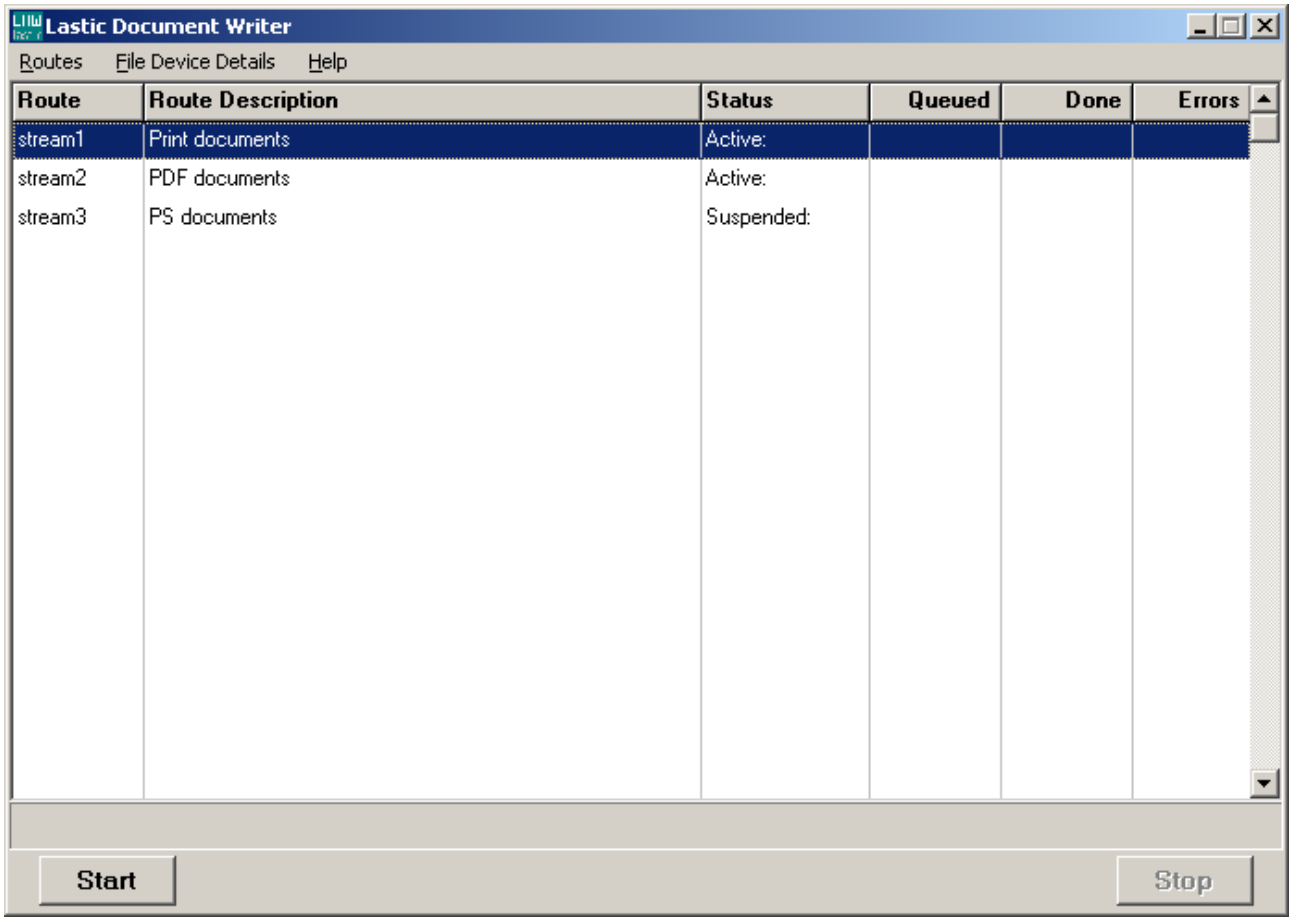
Table controls
These now have an optional Background colour attribute (defaults to White).
Individual cells now have background Colour and FontStyle attributes.

Document Object Model
LDW has now implemented a Document Object Model which loads the source XML file before the productions of the output.

2. Maintaining Routes

Invoke LDW (start ldw.exe by double clicking the routine or shortcut). This invokes the main form but does not start the actual processing.

The Main Form



The Main form is displayed when the application is invoked. It is initially dormant and requires the Start button to be clicked to commence processing. Processing can be halted by clicking the Stop button. See section 3 for running LDW.

Configuration amendments may only be performed when the Writer is dormant. During processing, the menus, grid and start button are disabled.

Before the application will run, at least one route must be created and the file device details completed. The File Device Maintenance has been discussed in section 1 above.

Routing
LDW provides up to 1000 concurrent routes.
A route identifies:

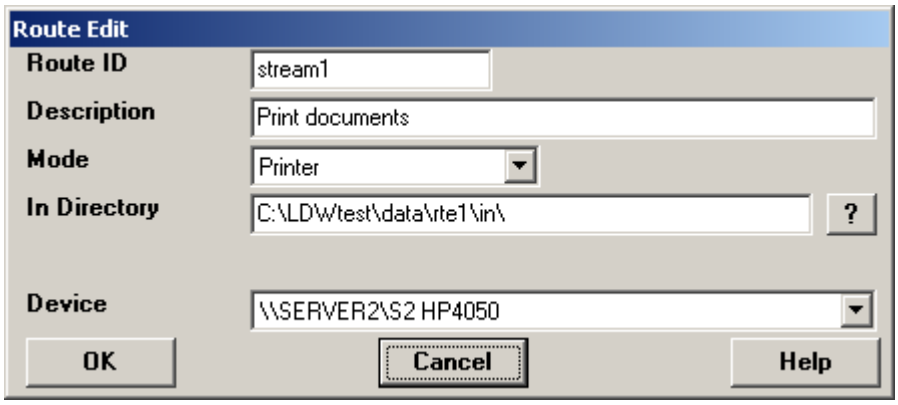
- The inward directory (where that route looks for source XML documents to process)
- The type of output (hard copy, Postscript file or PDF file)
- The physical output device/directory for that route.

This means that dedicated printers or directories can be used for specific document types.

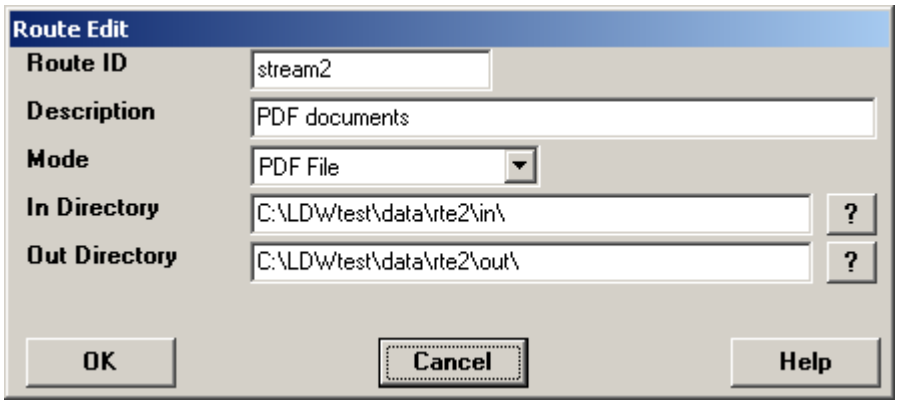
Any application wishing to use the LDW services will create an XML file (document source details) plus any associated image files (BMP, WMF, EMF files) and will save those into an inbound queue directory associated with that particular route. It should also be noted that library images can be used (where the same image is used many times such as logos) and these can be held in alternative directories and pointed to by the source document. Image files must be created BEFORE the referencing source XML document. For more information on this, see section 4.

2.1 Creating a Route

To create a route, select a blank row in the grid and then select Edit from the Routes menu. The following form will be displayed.



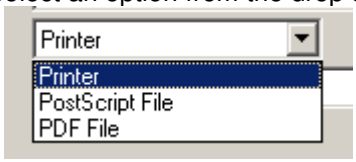
If PDF or PostScript file mode is selected the form will appear as:



Enter a route ID (max 10 characters) and a Description (max 40 characters) to identify this route.

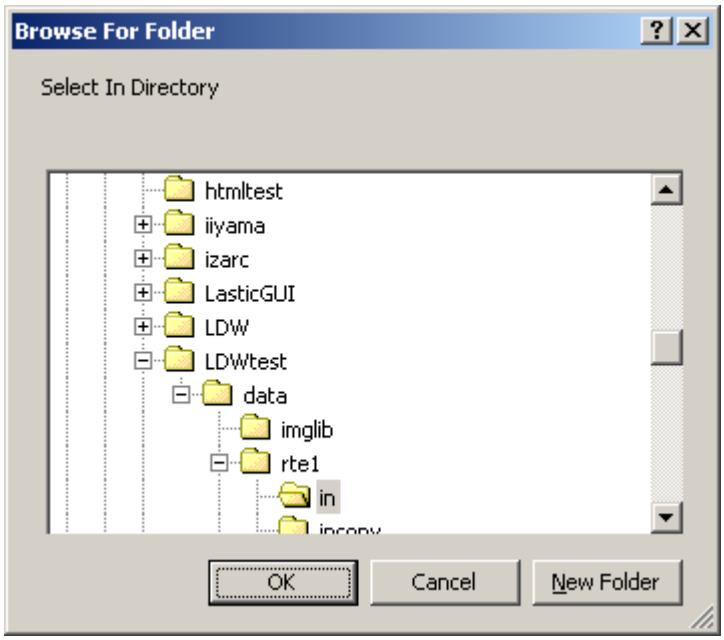
Mode

Mode is the output form for this route. Select an option from the drop down list. (See below).



Changing the mode will cause the form to request either the print device or the outbound directory to receive the generated files.

In Directory
The In Directory identifies the whereabouts of the source files for that route. It is selected by clicking on the '**?**' button in the above forms and will invoke the form shown below.



Select the drive and directory by double clicking the required items. When identified, click the OK button.
Click the New Folder button to add a directory.
This form will close and the selected directory will appear in the edit form.

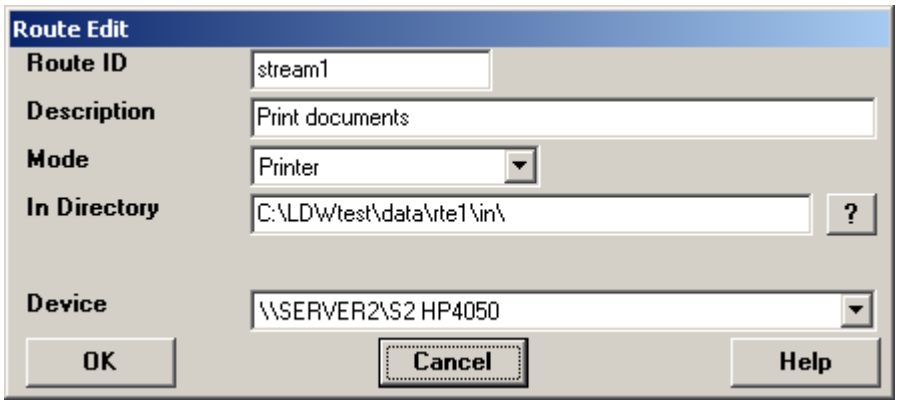
Out Directory
For PDF and PostScript file modes, the Out Directory identification follows that of the In Directory. It is not defined for Print output. Click the New Folder button to add a directory.
It is recommended that the In and Out directories are not the same for a route.

Device
For hardcopy output, select the printer for this route.

Warning: ***Do not select any device which will cause any further prompts since ldw.exe will not respond to them and the processing will, effectively, hang waiting on operator intervention.***

2.2 Amending a Route

To amend a route, select the route within the grid and then click the Routes/Edit menu.



The existing details are displayed and may be amended as described in the previous section.

2.3 Activating a route

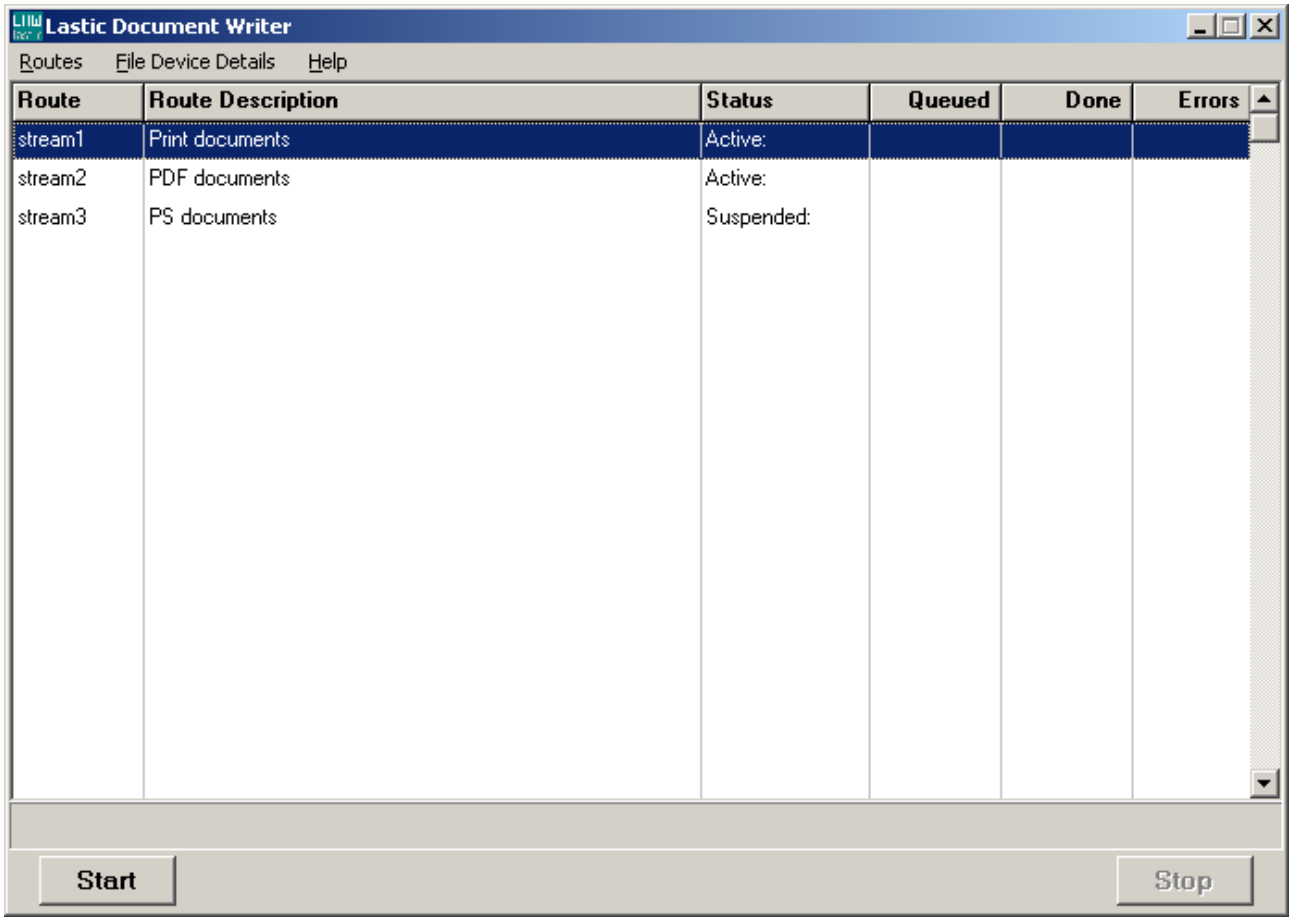
When a route is created it is suspended.
A route can be suspended or activated from the Routes menu.
Select the route in the grid then click the Routes/Active menu. This will reverse the current status of the Active flag – the menu will show a √ if active or no √ if suspended. Once changed, the status column for that route will also be changed.

2.4 Deleting a route

A route may be deleted by selecting that route and clicking the Routes/Delete menu item.
A message box form will be launched to request confirmation of the deletion.

3. Running LDW

Invoke ldw.exe and the main form below is displayed.



3.1 Starting the Processor

Once the file device details and the routes have been defined, start the processor by clicking the Start button.
The menus, grid and Start button will be disabled and the Stop button enabled.

LDW will cycle through each route and will process the oldest source file found in each active route. This mechanism ensures that each route gets an equitable share of the resource. When a file ready for processing is identified, the grid will highlight the route being processed and will also display the source file in the panel below the grid.

As each document is processed, the results are written to the text file ldw.log in the application directory. This will continue to grow until it is deleted (or removed/renamed) whereupon a new file will be created.

3.2 Stopping the Processor

The processor can be stopped by clicking the Stop button. It will continue to process the current route until that file has been completed whereupon the form's controls (above) will become enabled and the Stop button disabled.
Changes can now be made as required or the processor closed down by clicking the system 'X' control (top right of the form).

3.3 Creating source documents and images.

When a document is located in a route's In Directory it is checked to ensure that it has been completely written and closed by the creating application. If not, it is ignored and the next oldest file in that route's In Directory checked. This ensures that LDW is not in conflict with the generating routine. Also, because the processor will locate and load any associated image files those image files MUST be created BEFORE the primary source (XML) document is closed and made available to the LDW.
Within the source XML document image files are referenced by their filename and, if they are in a separate directory to the In Directory, they must also have a full file path.

Images in a directory other than the In Directory are not deleted with the source file thus a separate image library directory can be maintained for creating and storing re-usable images such as logos etc.
The form and creation of source documents is described in detail in section 4 below.

3.4 Processing a document.

When a source document is located and checked that it is available for processing, the document is read and the 'output' form is generated to a logical print device 'canvas'. This includes placement of images, wrapping of multiline text, drawing of frames/lines etc.
The logical print device may be a physical printer (via Windows Print Manager) or to a postscript file in the Out Directory if a postscript or PDF file is required.
Each completed page is written to the logical print device.
Upon completion of the print stage:

- If the output device is a printer that document is complete and the source files are deleted and ldw.log updated.
- If the output device is a postscript file then LDW waits for that file to be completely created and closed. It is then considered that document is complete and the source files are deleted and ldw.log updated.
- If a PDF file is required then LDW waits for the completion of the postscript file creation then the Ghostscript routine is called with the references to the postscript and PDF filenames passed as parameters. LDW then waits for the PDF file to be created whereupon that document is considered complete and the source files and the postscript file are deleted leaving the PDF file only in the Out Directory. Again ldw.log is updated.

3.5 Error situations

Errors may arise including situations of:

- Invalid XML file – missing elements
- Associated image files not found

In such cases, the error is recorded in ldw.log and the error count column updated on the main form.

3.6 Automated start and stop actions

LDW provides automated start and stop facilities for use within command files.
To start LDW from a command file issue the command: "ldw autostart"

To stop LDW remotely, create a null file (content irrelevant) with the name "ldwstop.req" in the LDW application directory. When there are no more XML files to process in any "in" directory LDW will check to see if such a file exists and, if so, will rename the above file to "ldwstop.fin" and then terminate. Command file script authors must be aware that there can be a delay between the creation of the ".req" file and the subsequent renaming of it to a ".fin" extension due to the completion of processing any remaining XML source files.

4. Source Files

4.1 Structure

Document source files are ‘well formed’ XML files.
A DTD is provided (see Appendix 1) as the definitive structure of such a document.

4.2 Generating

It is expected that source files will be generated by applications. If so, those applications must adhere to the DTD format.

Lastic’s L-DPM application for Cache environments provides a mechanism for generating XML file build routines from DTD resources thus ensuring document correctness.

The size limit of an individual document is that which can be loaded as a DOM within run-time memory but typically would be hundreds of pages.

4.3 Making Copies

It may be simpler (and bulletproof) to generate the XML file to a temporary directory and, when complete, copy it to the In Directory. Make sure that any image files required have already been stored in the In Directory.

4.4 Library Images

Image files may either be stored in the In Directory (before the completion of the Source document) or may exist in an ‘image library’ directory to be re-used for various documents.

Appendices

1. DTD

The DTD for the document source file is listed below.
The Primary element is 'Report'. Information relating to specific elements and attributes are given over page.

```
<!ELEMENT Report (Output, Page+)>
<!ELEMENT Output EMPTY>
<!ATTLIST Output
        DeviceScaling CDATA #IMPLIED
        Orientation (L|P) #REQUIRED
        Copies CDATA #REQUIRED
        PageMarginLeft CDATA #IMPLIED
        PageMarginTop CDATA #IMPLIED
        PassWord CDATA #IMPLIED
        Permissions CDATA #IMPLIED
        AltLineEnd CDATA #IMPLIED
        >
<!ELEMENT Page (Text*,Image*,Table*)>
<!ELEMENT Text (Font, TextLine)>
<!ATTLIST Text
        Position CDATA #REQUIRED
        FrameSize CDATA #REQUIRED
        FrameColour CDATA #IMPLIED
        BackColour CDATA #IMPLIED
        Alignment (L|R|C) #REQUIRED
        LineSpacing CDATA #IMPLIED
        TextMarginSide CDATA #IMPLIED
        TextMarginTop CDATA #IMPLIED
        >
<!ELEMENT TextLine (Font* ,#PCDATA)>
<!ELEMENT Image EMPTY>
<!ATTLIST Image
        Position CDATA #REQUIRED
        FileName CDATA #REQUIRED
        FileType (BMP|WMF|EMF) #REQUIRED
        >
<!ELEMENT Table (Font, Column+, Row+)>
<!ATTLIST Table
        Position CDATA #REQUIRED
        FrameSize CDATA #REQUIRED
        FrameColour CDATA #IMPLIED
        BackColour CDATA #IMPLIED
        ColumnLineSize CDATA #IMPLIED
        RowLineSize CDATA #IMPLIED
        LineSpacing CDATA #IMPLIED
        TextMarginSide CDATA #IMPLIED
        TextMarginTop CDATA #IMPLIED
        ColumnOverrun CDATA #IMPLIED
        >
<!ELEMENT Column EMPTY>
<!ATTLIST Column
        Width CDATA #REQUIRED
        Alignment (L|R|C) #REQUIRED
        >
<!ELEMENT Row (Cell+)>
<!ELEMENT Cell EMPTY>
<!ATTLIST Cell
        Colour CDATA #IMPLIED
        BackColour CDATA #IMPLIED
        FontStyle CDATA #IMPLIED
        Text CDATA #REQUIRED
        >
<!ELEMENT Font EMPTY>
<!ATTLIST Font
        FontName CDATA #REQUIRED
```

FontSize CDATA #REQUIRED
FontStyle CDATA #IMPLIED
FontColour CDATA #IMPLIED
>

Element: Output

Occurs once per document and defines attributes of:

Orientation

'L' = Landscape , 'P' = Portrait

Copies (applicable for actual print output only)

numeric and must be greater than 1

DeviceScaling

Applicable to a few print devices where the logical pixels per inch are not available from the system call GetDeviceCaps for the value of LogPixelsX.
If required then the DeviceScaling string is set to: Printer_name,factor*Printer_name,factor …
Where Printer_name is that known by the operating system and factor is the relative ratio of logPixelsX/600 for that device. For example if a printer uses 1200 Pixels per inch then the factor would be 2 (1200/600).

PageMarginLeft

The number of millimetres to leave between the logical left edge of the page and the boundary of the page. Many printers do not start printing at the physical edge of the paper but commence a few millimetres from that edge. In order to have a consistent output for various printers you may wish to leave, say, a margin of 10 Millimetres from the edge of the paper to the logical, left hand edge of the paper. All other horizontal measurements are then related to this margin.

PageMarginTop

As for PageMarginLeft but for the top of page margin.

PassWord

Optional item which, if included and is not a null value, will set that password into the resultant PDF document. The user will need to investigate the implications of such passwords for their own operations.

Permisions

Optional item which, if included and is not a null value, will set those permissions into the resultant PDF document. The use will need to identify the permission values required for such PDF documents.

AltLineEnd

This optional setting provides alternative line ends (other than CR/LF) to be used within the incoming document. This wil be provided as a numeric value which translates to a single character. Where such a character is then located within a text string, it will be converted into CR/LF for inclusion within the PS/PDF file or to a print device.

Element: Page

This element occurs for each page to be printed.
It references the various Text, Table and Image elements to appear on that page. It has no attributes of its own.

Measurements

All measurements (other than Font size) are defined in millimetres and may have decimal fractions. Conversion to the print canvas will be to the nearest point which the print driver can handle and will be stored and used as double precision numbers.

Element: Text

This may occur as frequently as required within the page. It has both attributes and associated elements of Font and TextLine.
Attributes consist of:

Position

A string of values representing Left, Top, Right, Bottom in millimetres from the TopLeft logical page margin. The block of text provided within the TextLine associated element of style defined by the associated Font Element must fit within these bounds. Text will be wordwrapped but lines will be truncated if they will not fit within these bounds.

FrameSize

The thickness (in 10ths of a millimetre) of a frame (box) to be drawn round this text. If no frame is required, the value must be zero.

FrameColour

The colour of the surrounding box of frame. See Colours section below for details.

BackColour

Optional attribute to define the background colour of this text object (see Colours for details).

Alignment

This defines the text alignment within the position 'box'. It must be one of:
L (left), R (right) or C (centre).

LineSpacing

This is the relative spacing between lines – default is 1. It may be set to any value required (e.g. 1.4, 2.035 etc.)

TextMarginSide

This is the left (and right) margin between the Left and Right positions above and the start of the text. It is specified in millimetres. Typically, for a position without a frame this could be 0. For frames, a margin it usually required otherwise the text will 'merge' with the vertical lines.

TextMarginTop

This is the top and bottom margin and applies as for TextMarginSide above.

Element: TextLine

This may contain multiple sub-nodes of PCDATA, CDATA and Font elements.
In order to accommodate new paragraphs, a new paragraph may be indicated by the character ASCII 127 or <CR><LF>. Fonts may be defined only for those elements which are different to the base font for this Text Element thus allowing changes of style only (for example) without having to define all of the font's characteristics.

Element: Font

This is used to define font presentation for each text and table element.
Attributes consist of:

FontName

Mandatory and must be recognized font on the computer

FontSize

Mandatory and must give size in points

FontStyle

Optional (default is regular) and may consist of combinations of B (bold), I (italic) and U (underline). They may appear in uppercase or lowercase and in any order.

FontColour

Optional (default is black) and will be one of the colour options as for FrameColour above.

Element: Image

This element defines a 'picture' to be placed on the page.
It has attributes of:

Position

A string of values representing Left, Top, Right, Bottom in millimetres from the TopLeft logical page margin. The image defined by the filename attribute will be resized to fit within these bounds.

FileName

The name of the image file to be loaded. If it has no file path then it is assumed to be present within the In Directory. If it has a file path different to the In Directory then it will be assumed to be a library image and will not be deleted when the document has been produced.

FileType

This must be one of BMP, WMF or EMF (upper or lower case).
It is recommended that WMF or EMF is used to provide the optimum presentation when resized.

Element: Table

This defines a table/grid of data to be presented. It has associated elements of Font, Column and Row)
Attributes consist of:

Position

A string of values representing Left, Top, Right, Bottom in millimetres from the TopLeft logical page margin. The table data within the Cell element must fit within these bounds. Lines will be truncated if they will not fit within these bounds.

FrameSize

As for the corresponding attribute of the Text element.

FrameColour

As for the corresponding attribute of the Text element. See Colours section below

BackColour

As for the Text Element. See Colours section below

ColumnLineSize

The thickness (in 10ths of a millimetre) of the vertical lines to be drawn between columns. If no vertical lines are required, the value must be zero.

RowLineSize

The thickness (in 10ths of a millimetre) of the horizontal lines to be drawn between rows. If no horizontal lines are required, the value must be zero.

LineSpacing

As for the equivalent attribute within the text element.

TextMarginSide

As for the same attribute within the Text element and applies to each individual cell. It is of more significance where column lines are drawn or a left aligned column follows a right aligned column.

TextMarginTop

As for the same attribute within the Text element and applies to each individual cell.

ColumnOverrun

If present and set to 'N' (or "n") then cell texts will be truncated to fit within the cell boundary. Otherwise the cell text will overrun the cell boundary to the left or right as determined by the alignment of the column and the length of the text.

Element: Column

This defines the column widths and alignment. An element must be defined for each column and the order presented within the XML file is the left to right order across the page.
Attributes consist of:

Width

The width, in millimetres, of this column.

Alignment

This defines the text alignment within the cell. It must be one of:
L (left), R (right) or C (centre).

Element: Row

This element is repeated for each row of the table and contains elements for each cell (i.e. column data) within that row. It has no attributes.

Element: Cell

This element **MUST** appear for every column for every row – including empty cells.
Attributes consist of:

Colour

The colour of the text in that cell. If not defined, black is assumed. If defined it must be a colour as defined for the FrameColour attribute.

Text

The text to be output in that cell. It will not be wrapped but will be truncated if too long for that cell. It will be aligned as per the corresponding column Alignment attribute.

BackColour

Optional alternative background colour for this cell. If not present, the overall table background colour is used.

FontStyle

May be one or more of B(bold), I (Italic) or U (Underline). Changes to the font face or size is not permitted in then individual cell's text.

Colours

Where defined then this item may be in one of the following formats

- Windows colour name (e.g. "teal")
- Web colour name (e.g. "DarkSalmon")
- Hex representation (e.g. #F2A3B5)
- RGB value (e.g. 125,255,50)

Names may be entered in upper or lower case. If not defined or misspelt then Black is assumed for foreground and White for background.