Contents

Figures

Tables

Preface

Welcome to the DEC Documentation Rebuild project. This project, driven by the community, aims to recreate as much of the documentation for vintage Digital Equipment Corporation (DEC) computer equipment and software as possible.

The reason for this is twofold:

1. Much of the documentation is in a poor state. It can be hard to make out some parts of it due to bad scanning, or generally poor source material.
2. The existing documentation can be very hard to work with. Scanned PDFs, even with advanced OCR, are hard to search and navigate. Rebuilding the documentation gives us the chance to correct that with embedded hyperlinks, clean text, and a generally more usable experience.

Chapter 1
How we are doing it

It is highly likely (backed up by some evidence from the documentation itself[1]) the original documentation is written in LaTeX. In order to try and maintain as much of the original layout and pagination as closely as possible to the original LaTeX has again been chosen as the method to re-create the documentation. This also lets us separate out the style from the content making it much faster, once suitable document classes have been created, to recreate documents with the minimum of effort.

[1] For example using a space separated console prompt such as > > > to combat LaTeX conversion of >> into »

Chapter 2
Contributing

You'd like to contribute? Fantastic! We are always looking for more volunteers to help recreate more documents. Just fork this repository and get writing. Most of the groundwork has been done for you in the form of some handy document classes (dec.cls and decsectional.cls) that implement reasonably accurately[1]. By all means take a look at one of the existing LaTeX files for an idea of how to go about implementing the document.

We only have a few stipulations when it comes to style, both of content and general working:

- The hyperref package is automatically included in the base dec.cls file. Please use hyperlinks and hyperrefs within the document to link to sections, figures and tables where they are mentioned in the text (see below for helper functions for these). Also please use the `pdf{...}` command to wrap any references to other DEC documents. This just creates a href to a PDF document in the same directory at the moment though that may be subject to change in the future.
- Your document should be named by the order number of the document (for example EK-VAXAC-OM-003.tex) with any sub-parts being named the same but with a hyphenated suffix (for example EK-VAXAC-OM-003-ch1.tex).
- In DEC documentation all figures have a reference number associated with them which denotes the author, their image sequence number, and the year of production. When you cut out an image from the original scanned PDF please include this reference number. Name the image file with this reference number and place it in the **fig** folder, then use the `fig{ref}{caption}` command to reference it within your document.
- Title page images should be stored in the **titles** folder and named after the order number of the document.

When transcribing you should attempt to match the layout and pagination of the original document as possible. This is chiefly so that someone who is referencing the original scanned PDF and someone who is referencing the rebuilt PDF both get the same page numbers for the same information and can collaborate more seamlessly. Some bleed of paragraphs from page to page is fine, but tables, figures, and sections should be on the same pages as the original where possible.

[1] Apart from the fonts which we have tried to find reasonable matches for in the standard LaTeX font library, but has proved almost impossibe - and we don't want to have to use third party fonts to complicate matters.

Chapter 3
Helper Functions

We have a number of handy helper functions to aid in keeping the layout of the document as close to the original as possible without you having to think too hard about how to do it.

They are included as (currently) two class files, `dec.cls` and `decsectional.cls`. The former is the master class which is geared towards simpler non-numbered (single chapter) documents. The latter extends the master class to allow creation of longer chapter based documents.

3.1 Figures

There are two figure helper functions, `fig` and `ttfig`. The first of these is used to include a figure into the document at the current location.

```
\fig[Scale]{ImageRefCode}{Caption For This Figure}
```

The `Scale` parameter is optional and sets the width of the image as a percentage (0.0 - 1.0) of the page width. The `ImageRefCode` is the ID code (XX-NNNN-YY) of an image within the fig directory, and the caption is placed above the image and included in the list of figures in the contents section.

The `ttfig` is a little different in that it defines a new environment which is used for creating text-based (ASCII art, console display, etc) figures.

```
\begin{ttfig}{This is the caption}
 ______ _
|  ____(_) __ _ _   _ _ __ ____
| |_  | |/ _` | | | | '__/ _ \
|  _| | | (_| | |_| | | |  __/
|_|   |_|\__, |\__,_|_|  \___|
         |___/
\end{ttfig}
```

Result:

Figure 3-1: This is the caption

A DEC-style label is automatically created for every figure (figure:F or figure:C-F) for hyperlinks to jump to.

3.2 Tables

Tables are internally handled by the `tabularx` package, but are wrapped in extra code to handle DEC style labels and captions. The main table environment is:

```
\begin{tbl}{Caption Here}{Spec}
... content ...
\end{tbl}
```

The `Spec` is a normal tabularx column set specification describing the columns in the table. A top and bottom horizontal line are automatically added, so just add the headings, another hline, and then the table body. For example:

```
\begin{tbl}{A Sample Table}{c c}
\textbf{First column} & \textbf{Second column} \\
\hline
This is something & This is something else \\
This is more & This is even more \\
\end{tbl}
```

The result:

Table 3-1: A Sample Table

First column	Second column
This is something	This is something else
This is more	This is even more

If a table is too long to fit on one page you can finish the table early, then re-start it on the next page using the `tblcont` environment. This is exactly the same as the

`tbl` environment except the word (Cont.) is added to the caption numbering, and the table is not included in the list of tables in the TOC.

```
\begin{tblcont}{A Sample Table}{c c}
\textbf{First column} & \textbf{Second column} \\
\hline
This is exta & This bit wouldn't fit in the previous table.\\
\end{tblcont}
```

Table 3-1 (Cont.): A Sample Table

First column	Second column
This is exta	This bit wouldn't fit in the previous table.

3.3 Chapters and sections

As well as the normal chapter and section (both starred and unstarred variant) commands we have u-prefixed variants which serve as a half-way house between the starred and unstarred variants. Like the starred variants they are unnumbered, but like the unstarred variants they are included in the TOC. This allows for unnumbered documents to be created yet still have a functional TOC with minimum fuss.

3.4 References

Creating links within the document is made easier with the use of a few reference helper functions: `figref` and `tableref`. Both just take a DEC-style figure or table reference number (for example 2-5) and format the name of the link for you automatically.

There is also a `pdf` helper function which just takes a DEC order number and links to the PDF externally.

Chapter 4
License