Typesetting Forms with LaTeX

Mark A. Roth, Maj, USAF
Department of Electrical and Computer Engineering, Air Force Institute of Technology, AFIT/ENG,
Wright-Patterson AFB OH, 45433-6583 USA
513-255-9263; FAX: 513-476-4055
Internet: mroth@afit.af.mil

Abstract

The Air Force Forms System (AFFORMS) is combination of a user-friendly fill-in-the-blank front end and a LaTeX-based forms typesetting system. The overall system is described and the procedure to develop a LaTeX style for a form is presented.

Introduction

The United States Air Force (USAF), like any large corporation or government agency, utilizes hundreds of different forms in its day to day business. Some forms are simple to fill out and if a mistake is made the form is changed via erasure or cross out, or is simply reaccomplished. Other forms are either more complex or are such that even the simplest error cannot be tolerated. For example, the USAF Officer Evaluation System (OES) requires that an officer performance report (OPR) be rendered annually for each officer on active duty. Furthermore, before each promotion board, a promotion recommendation form (PRF) must be completed by the officer's commanding general. Although these forms are not exceedingly complex, they must be typographically perfect, and since each completed form undergoes several layers of review and revision, a single form may be accomplished and reaccomplished from ten to twenty times.

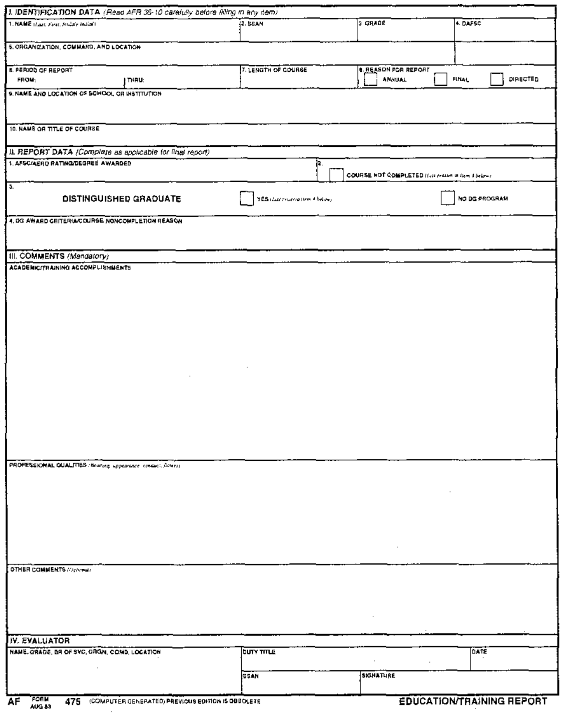
Only a couple of years ago all of this work was accomplished via preprinted forms and an electric (or manual!) typewriter. Now with the widespread availability of computers and laser printers, many offices use word processing programs to lay out the blocks of information for the form and then print directly onto the preprinted forms. This is not entirely satisfactory since the typist needs to maintain the critical spacing requirements within the word processing file—the addition of a line of text means the deletion of a line of space. Furthermore, the Government Printing Office (GPO) rarely supplies consistently printed forms. Each batch is printed on a slightly different position on the page, and, even worse, the forms are sometimes not horizontal. In the days of typewriters these slanted forms could be fed into the typewriter in the same slanted manner so that the typed text would line up with the form boxes. This is quite impossible with laser printers and thus many forms are made useless.

In 1989, several colleagues and myself developed a competely automated forms preparation system for the purpose of preparing and printing OPRs and PRFs. This system avoids all of the above problems by printing the entire form and the user's text onto a plain piece of paper. This system provides a friendly window/menu-oriented interface for the user to compose or edit form entries, and a LaTeX-based typesetting system to produce, preview, and print the complete form.

In this paper, I describe the Air Force Forms Systems (AFFORMS) which evolved from these initial requirements, with specific emphasis on the how the forms were produced using the LaTeX picture environment, and how we standardized the form input parameters. Using these techniques other forms can easily be generated and put into use in your organization. The form I will be presenting is the AF Form 475, Education/Training Report (Figure 1). This form is used to document the progress of Air Force officers in a variety of long term education or training programs. I will present a brief overview of how the current system works from a user viewpoint, and then present the technical issues of designing the LaTeX style for producing the form.

The AFFORMS Package

The AFFORMS package can be functionally broken into two main areas: the user interface and the LaTeX processing files. The user interface provides fill-in-the-blank screens for each of the forms, and a pull-down menu system to perform various functions required to edit, view, or print the forms. The user does not need to know anything about LaTeX, except for the usual rules about the different dashes and



I. IDENTIFICATION DATA *(Read AFR 35-10 carefully before filling in any item)*

1. NAME *(Last, First, Middle Initial)*	2. SSAN	3. GRADE	4. DAFSC
5. ORGANIZATION, COMMAND, AND LOCATION			
6. PERIOD OF REPORT FROM: THRU:	7. LENGTH OF COURSE	8. REASON FOR REPORT □ ANNUAL □ FINAL □ DIRECTED	
9. NAME AND LOCATION OF SCHOOL OR INSTITUTION			
10. NAME OR TITLE OF COURSE			

II. REPORT DATA *(Complete as applicable for final report)*

1. AFSC/AERO RATING/DEGREE AWARDED	2. □ COURSE NOT COMPLETED *(List reason in item 4 below)*
3. DISTINGUISHED GRADUATE	□ YES *(List criteria item 4 below)* □ NO DG PROGRAM
4. DG AWARD CRITERIA/COURSE NONCOMPLETION REASON	

III. COMMENTS *(Mandatory)*

ACADEMIC/TRAINING ACCOMPLISHMENTS

PROFESSIONAL QUALITIES *(Bearing, appearance, conduct, fitness)*

OTHER COMMENTS *(Optional)*

IV. EVALUATOR

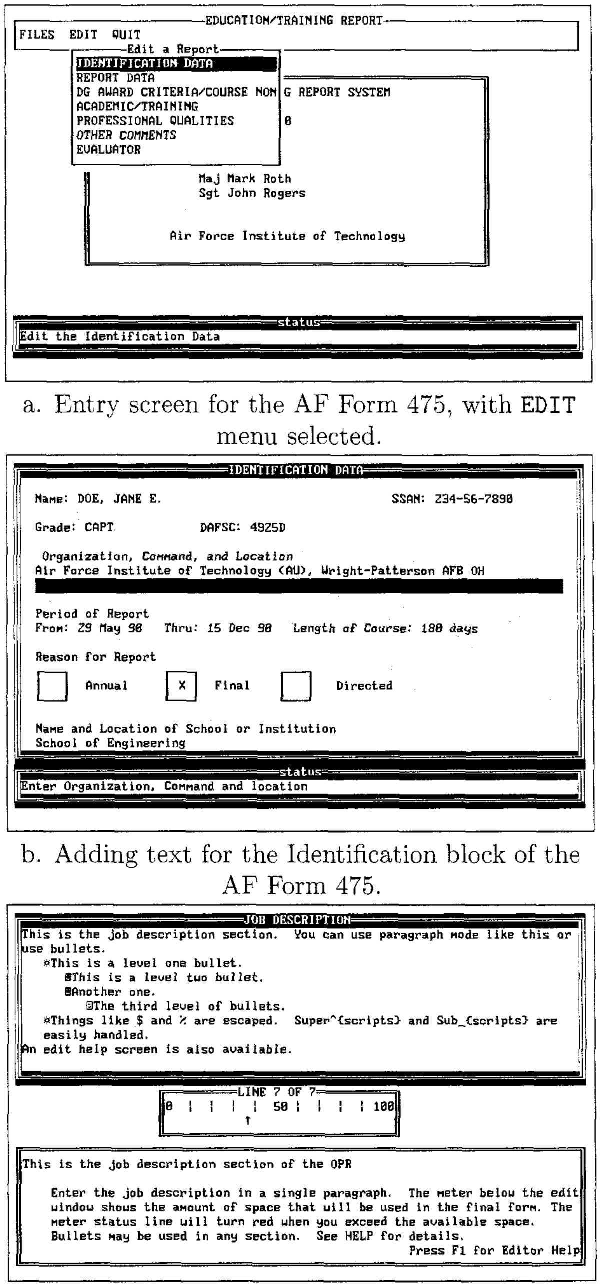
NAME, GRADE, BR OF SVC, ORGN, COMD, LOCATION	DUTY TITLE	DATE
	SSAN	SIGNATURE

AF FORM AUG 83 475 (COMPUTER GENERATED) PREVIOUS EDITION IS OBSOLETE EDUCATION/TRAINING REPORT

Figure 1: Computer generated AF Form 475.

how to get the quote symbols right. Special symbols and actions like dollar signs, percent signs and superscripts and subscripts are automatically captured by the software and translated to the proper codes when the LaTeX input file is prepared. Special keys are used to indicate that bullets, sub-bullets, or sub-sub-bullets are required. Some forms use a visual meter to indicate, for large text blocks, approximately how much actual space LaTeX will need to typeset the input. Figure 2 shows some sample screens from the interface.

The Vitamin C graphics/window library of C functions was used to write the user interface. This turned out to be a fortuitous decision as versions of the Vitamin C package are available for many systems including MSDOS, UNIX, and VMS based computers. We were thus able to port our software to many different architectures, although it is primarily used on MSDOS machines. Any implementation of TeX can be used to run the software. On the MSDOS computers, we used public domain software including SBTeX, DVIVGA and DVIEW screen previewers, and Nelson Beebe's printer driver family. Each interface contains about 1500 lines of C code, most of which is duplicated in each interface. Once



```
                      EDUCATION/TRAINING REPORT
FILES  EDIT  QUIT
          Edit a Report
       IDENTIFICATION DATA
       REPORT DATA
       DG AWARD CRITERIA/COURSE NON G REPORT SYSTEM
       ACADEMIC/TRAINING
       PROFESSIONAL QUALITIES       0
       OTHER COMMENTS
       EVALUATOR

                     Maj Mark Roth
                     Sgt John Rogers


                 Air Force Institute of Technology

                              status
Edit the Identification Data
```

a. Entry screen for the AF Form 475, with EDIT menu selected.

```
                         IDENTIFICATION DATA
 Name: DOE, JANE E.                            SSAN: 234-56-7890

 Grade: CAPT           DAFSC: 4925D

  Organization, Command, and Location
 Air Force Institute of Technology (AU), Wright-Patterson AFB OH

 Period of Report
 From: 29 May 90   Thru: 15 Dec 90   Length of Course: 180 days

 Reason for Report
 □  Annual    ☒  Final    □  Directed

 Name and Location of School or Institution
 School of Engineering
                              status
Enter Organization, Command and location
```

b. Adding text for the Identification block of the AF Form 475.

```
                          JOB DESCRIPTION
This is the job description section.  You can use paragraph mode like this or
use bullets.
   *This is a level one bullet.
      ■This is a level two bullet.
      ■Another one.
         ■The third level of bullets.
   *Things like $ and % are escaped.  Super^{scripts} and Sub_{scripts} are
   easily handled.
An edit help screen is also available.

                 LINE 7 OF 7
             0 | | | | 50 | | | | 100
                     ↑

This is the job description section of the OPR

    Enter the job description in a single paragraph.  The meter below the edit
    window shows the amount of space that will be used in the final form. The
    meter status line will turn red when you exceed the available space.
    Bullets may be used in any section.  See HELP for details.
                                                        Press F1 for Editor Help
```

c. Entering text using bullets in the Job Description block of an OPR.

Figure 2: Samples screens from the AF Form 475 and OPR user interface.

a developer is familiar with the system it takes only 5-10 days to create and debug a new interface.

The AFFORMS package can currently typeset eight[1] forms which have been approved by the Secretary of the Air Force for use by Air Force agencies, and the Education/Training Report described in this paper which has been submitted for approval.

Creating a Form in LaTeX

Designing a form style with LaTeX is relatively straightforward. Most forms can be done with simple application of `picture` environment commands. Slanted lines that do not conform to the available slopes of the `line` fonts that are available require a more sophisticated package to be added such as the `epic` macros. In our applications slanted lines were only needed to "check" boxes. In forms where it was permitted to allow the computer to check the form, square boxes avoided the slanted line problem. Let me take you through the development of the AF Form 475 style (hereafter referred to as the 475).

Analyze the Fields. Each box of the form which can be filled in needs to be identified and given a macro name. In addition fields which can be checked need a macro flag to indicate whether or not the field should be checked.

First, I created default internal names for each of the entries. I used the name of field with the second character an @ to avoid conflicts with user defined macros. Some of the text fields for the 475 are:

```
% Defaults for entries
\def\N@ME{}   %student name
\def\S@AN{}   %student ssan
\def\G@ADE{}  %student rank/grade
\def\D@FSC{}  %student duty specialty code
\def\O@GANIZATIONONE{}  %1st line of org
\def\O@GANIZATIONTWO{}  %2nd line of org
etc.
```

The REASON FOR REPORT block on the 475 is a set of three boxes, one of which is checked. The macro `\R@ASONFLAG` will be compared to one of three constant value macros and depending on which one is matched the appropriate box will be checked.

```
\def\R@ASONFLAG{}   %will have one of the
\def\A@NUAL{ANNUAL} % following values
\def\F@NAL{FINAL}
\def\D@RECTED{DIRECTED}
```

Each of the fields then has a macro which will receive the user's input for the field. Alternate names can be specified (e.g., ssn and ssan in the following) although this is not really necessary in our system, as the user interface generates the LaTeX input file. The system can be used without the user interface so this capability could be useful. We compare each user macro with an empty field, in case non-empty defaults are desired for the entries. Then, if a field is not specified or is specified by the user but with a blank entry, then the default is used. Since all users of our system go through the user interface, all of defaults are blank.

```
\def\@e{}  % empty field for comparison
\def\name#1{\ifx\@e#1\else\def\N@ME{#1}\fi}
\def\ssn#1{\ifx\@e#1\else\def\S@AN{#1}\fi}
\def\ssan#1{\ifx\@e#1\else\def\S@AN{#1}\fi}
\def\grade#1{\ifx\@e#1\else\def\G@ADE{#1}\fi}
\def\dafsc#1{\ifx\@e#1\else\def\D@FSC{#1}\fi}
\def\organizationone#1{
  \ifx\@e#1\else\def\O@GANIZATIONONE{#1}\fi}
\def\organizationtwo#1{
  \ifx\@e#1\else\def\O@GANIZATIONTWO{#1}\fi}
\def\reasonflag#1{
  \ifx\@e#1\else\def\R@ASONFLAG{#1}\fi}
etc.
```

Determine Fonts. The next decision was to identify what fonts were required. We use 12pt roman[2] (cmr12) for all text unless a smaller font is required to let the required text fit in the field. The closest computer modern font is used to match the actual form text. Usually, the forms use sans serif and bold and italic sans serif fonts. I always list all fonts directly used in the style even if some are preloaded, so that a maintenance programmer knows what the font macros mean. For the 475 the fonts are:

```
%commented fonts are preloaded
%\font\tenrm=cmr10                %10pt roman
\font\elhrm=cmr10 scaled1150      %11.5pt roman
%\font\twlrm=cmr12                %12pt roman
\font\sixsf=cmss8 scaled750       %6pt sans serif
\font\sixsfb=cmssbx10 scaled600   %6pt bold ss
\font\egtsfb=cmssbx10 scaled800   %8pt bold ss
\font\ninsfb=cmssbx10 scaled900   %9pt bold ss
%\font\tensfb=cmssbx10            %10pt bold ss
\font\egtsfi=cmssi8               %8pt ss italic
\font\ninsl=cmsl9                 %9pt slant
\font\sixit=cmti8 scaled750       %6pt italic
%\font\egtit=cmti8                %8pt italic
```

We also preload the standard 12, 10, and 8pt fonts for typesetting text and math in a paragraph.

It would be better if we had fonts created at the design size rather than scaled, but usually it doesn't make much difference since we can control the exact placement of individual words and letters, if necessary, in the `picture` environment.

[1] OPRs, EPRs, Travel Orders, and Staff Summary Sheets

[2] To be totally honest, I used PostScript fonts for the form examples shown in this document so I could get better reductions.

Layout of the Form. The picture environment is used to layout the form. Here is where a ruler and little patience pays off. Most boxes and line separations on a form are usually of a standard size, so once you measure a few distances, simple addition gets you the others. We usually like to do our measurements from the top down and left to right, so we set the lower left corner coordinates of the picture to (0,-z), where z is the height of the form. Then for each measurement (x, y) from the top left, the appropriate \put command is \put(x,-y){...}. We used 1cm as the unit length. Here is the first part of the commands to typeset the 475 form:

```
\newcommand{\front}{
\clearpage
\begin{picture}(20.1,25.1)(0,-25.1)
   %second pair is the low. left corner coord.
\linethickness{.06cm}
\put(0,-25.1){\framebox(20.1,25.1)}%outer frame
\linethickness{.03cm}
\put(0.1,-.25){\egtsfb I. IDENTIFICATION DATA
  \egtsfi (Read AFR 36-10 carefully before
  filling in any item)}
\put(0,-.40){\line(1,0){20.10}} %top of box I
\put(0.1,-0.65){\sixsfb 1. NAME\ \sixit
  (Last, First, Middle Initial)}
\put(8.60,-.40){\line(0,-1){.84}} %vert line
\put(8.65,-0.65){\sixsfb 2. SSAN}
\put(12.90,-.40){\line(0,-1){.84}}
\put(12.95,-0.65){\sixsfb 3. GRADE}
\put(16.45,-.40){\line(0,-1){1.68}}
\put(16.50,-0.65){\sixsfb 4. DAFSC}
\put(0,-1.25){\line(1,0){20.10}} %bot of box I
\put(0.1,-1.50){\sixsfb 5. ORGANIZATION,
  COMMAND, AND LOCATION}
etc.
\end{picture}
} %end definition of front
```

Next, within the above picture environment, the user input text entries are positioned at the appropriate places within the form. This is better done after a new blank form is printed, since the computer modern fonts used for the form text usually take up less room than the text on the original form. A \parbox is used to set paragraphs, using \centering to center items as appropriate. We encountered a couple of interesting problems in trying to fit entries into the blocks. The first problem was a size constraint. Since we didn't want the user to have to specify a size function to put in the text, we had to automatically determine which size font to print certain critical items. Also, some entries could be entered as a single line of text, or multiple lines. The following code shows how we handled a two versus one line organization field with multiple sizes for the 475 form:

```
\ifx\O@GANIZATIONTWO\@e %then 1 line org
  \newbox\org
  \setbox\org=\hbox{\twlrm\O@GANIZATIONONE}
  \ifdim\wd\org>19.9cm %then too big at 12pt
    \setbox\org=\hbox{\elhrm\O@GANIZATIONONE}
    \ifdim\wd\org>19.9cm
       %then too big at 11.5pt set at 10pt
       \put(0.17,-1.95){\tenrm\O@GANIZATIONONE}
    \else %OK at 11.5pt
       \put(0.17,-1.95){\elhrm\O@GANIZATIONONE}
    \fi
  \else %OK at 12pt
    \put(0.17,-1.95){\twlrm\O@GANIZATIONONE}
  \fi
\else  %two lines output at 12pt
  \put(5.4,-1.60){\twlrm\O@GANIZATIONONE}
  \put(0.17,-2.00){\twlrm\O@GANIZATIONTWO}
\fi
```

By using a box, we can test the width of the entered text at various sizes and choose the maximum size that allows the entries to fit.

An example of checking boxes is shown below:

```
\ifx\R@ASONFLAG\A@NUAL %check the ANNUAL box
  \put(13.15,-2.85){\line(1,1){.42}}
  \put(13.15,-2.43){\line(1,-1){.42}}
\else\ifx\R@ASONFLAG\F@NAL %check FINAL box
  \put(15.72,-2.85){\line(1,1){.42}}
  \put(15.72,-2.43){\line(1,-1){.42}}
\else\ifx\R@ASONFLAG\D@RECTED %check DIRECTED
  \put(17.78,-2.85){\line(1,1){.42}}
  \put(17.78,-2.43){\line(1,-1){.42}}
\fi\fi\fi
```

The last typesetting problem we had was in the evaluators duty title block. The duty title needs to be centered in this block. When a two line duty title is used, it is possible that the first line, if centered, will overlap with the form text. However, if the first line is shifted right to avoid this overlap, then it looks strange if the second line is not shifted correspondingly. Of course, we can't shift the second line so far that it doesn't stay within the block. Thus some interesting calculations are needed to make this block format aesthetically. The code for this is shown next:

```
\ifx\E@ALUATORDUTYONE\@e\else
 \ifx\E@ALUATORDUTYTWO\@e% then 1 line dutytitle
  \put(8.5,-24.00){\makebox[8.35cm]
    {\centering\twlrm\E@ALUATORDUTYONE}}
 \else% two line dutytitle
  \put(8.5,-24.15){\parbox[b]{8.35cm}{
   \twlrm\twlbase
   %tighten up the baselines a little
   \addtolength{\baselineskip}{-.12cm}
   \newbox\dutyone\newbox\dutytwo\newbox\duty
   \setbox\dutyone=\hbox{\E@ALUATORDUTYONE}
   \setbox\dutytwo=\hbox{\E@ALUATORDUTYTWO}
   \setbox\duty=\vbox{
    \hbox to 8.20cm
     {\hfill\E@ALUATORDUTYONE\hfill}
    \parskip=0pt\par
    \hbox to 8.2cm
     {\hfill\E@ALUATORDUTYTWO\hfill}
    }
   \ifdim\wd\dutyone>5.2cm %5.2=8.2-1.5*2
     %then first line of duty to big to center
     %figure out how much whole thing needs to
     %be moved over to avoid conflict with
     %DUTY TITLE on form
     \newdimen\dutyin \dutyin=\wd\dutyone
     \advance\dutyin by -8.20cm
     \divide\dutyin by2 \advance\dutyin by1.5cm
     \ifdim\wd\dutyone<\wd\dutytwo
      %then see if 2nd line will overflow right
       \newdimen\dutyline \dutyline=\wd\dutytwo
       \advance\dutyline by\dutyin
       \ifdim\dutyline>8.20cm
        %then split lines,right justify 2nd line
         \hspace*{1.5cm}\box\dutyone\newline
         \hbox to 8.20cm{\hfill\box\dutytwo}
       \else%center both lines of duty title
        %left justifying first line
         \hspace*{\dutyin}\box\duty
       \fi
      \else
       %indent to spot right after DUTY TITLE
       \hspace*{\dutyin}\box\duty
      \fi
   \else% center duty title for both lines
    \box\duty
   \fi \vskip-\lastskip}}
 \fi
\fi
```

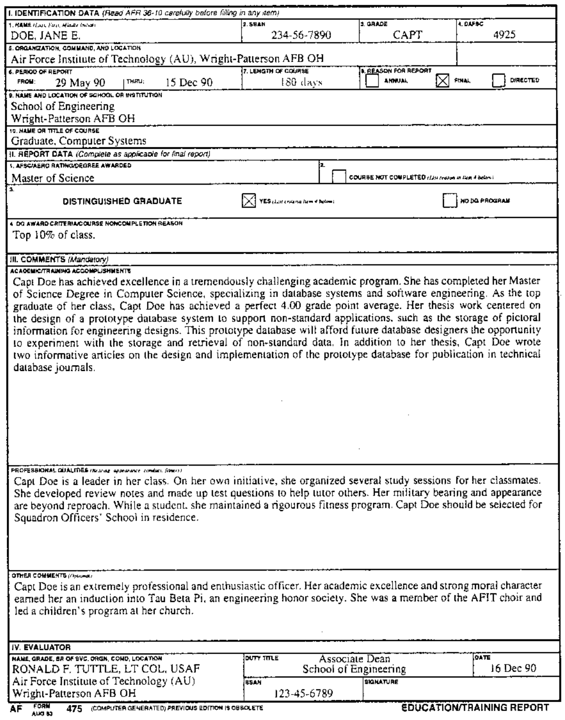
The LaTeX Input

The LaTeX input file simply consists of the definition of each of fields followed by the `\front` command. To get a blank form the commands are:

```
\documentstyle{af475}
\begin{document}
\front
\end{document}
```

Figure 1 showed an example of this. The input for a filled in form, automatically generated by our user interface, looks like the following:

```
\documentstyle{af475}
\begin{document}
\name{DOE, JANE E.} \ssan{234-56-7890}
\grade{CAPT} \dafsc{4925}
\organizationone{Air Force Institute of
Technology (AU), Wright-Patterson AFB OH}
\organizationtwo{}
\from{29 May 90} \thru{15 Dec 90}
\length{180 days} \reasonflag{FINAL}
\schoolone{School of Engineering}
\schooltwo{Wright-Patterson AFB OH}
\course{Graduate, Computer Systems}
\awarded{Master of Science}
```



I. IDENTIFICATION DATA (Read AFR 36-10 carefully before filling in any item)

1. NAME (Last, First, Middle Initial) DOE, JANE E.
2. SSAN 234-56-7890
3. GRADE CAPT
4. DAFSC 4925

5. ORGANIZATION, COMMAND, AND LOCATION Air Force Institute of Technology (AU), Wright-Patterson AFB OH

6. PERIOD OF REPORT FROM: 29 May 90 THRU: 15 Dec 90
7. LENGTH OF COURSE 180 days
8. REASON FOR REPORT □ ANNUAL ☒ FINAL □ DIRECTED

9. NAME AND LOCATION OF SCHOOL OR INSTITUTION
School of Engineering
Wright-Patterson AFB OH

10. NAME OR TITLE OF COURSE Graduate, Computer Systems

II. REPORT DATA (Complete as applicable for final report)

1. AFSC/AERO RATING/DEGREE AWARDED Master of Science
2. □ COURSE NOT COMPLETED (List reason in item 4 below)
3. DISTINGUISHED GRADUATE ☒ YES (List criteria in item 4 below) □ NO DG PROGRAM
4. DG AWARD CRITERIA/COURSE NONCOMPLETION REASON Top 10% of class.

III. COMMENTS (Mandatory)

ACADEMIC/TRAINING ACCOMPLISHMENTS

Capt Doe has achieved excellence in a tremendously challenging academic program. She has completed her Master of Science Degree in Computer Science, specializing in database systems and software engineering. As the top graduate of her class, Capt Doe has achieved a perfect 4.00 grade point average. Her thesis work centered on the design of a prototype database system to support non-standard applications, such as the storage of pictoral information for engineering designs. This prototype database will afford future database designers the opportunity to experiment with the storage and retrieval of non-standard data. In addition to her thesis, Capt Doe wrote two informative articles on the design and implementation of the prototype database for publication in technical database journals.

PROFESSIONAL QUALITIES (Bearing, appearance, conduct, fitness)

Capt Doe is a leader in her class. On her own initiative, she organized several study sessions for her classmates. She developed review notes and made up test questions to help tutor others. Her military bearing and appearance are beyond reproach. While a student, she maintained a rigourous fitness program. Capt Doe should be selected for Squadron Officers' School in residence.

OTHER COMMENTS (Optional)

Capt Doe is an extremely professional and enthusiastic officer. Her academic excellence and strong moral character earned her an induction into Tau Beta Pi, an engineering honor society. She was a member of the AFIT choir and led a children's program at her church.

IV. EVALUATOR

NAME, GRADE, BR OF SVC, ORGN, COMD, LOCATION
RONALD F. TUTTLE, LT COL, USAF
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH

DUTY TITLE Associate Dean
School of Engineering

DATE 16 Dec 90

SSAN 123-45-6789

SIGNATURE

AF FORM AUG 83 475 (COMPUTER GENERATED) PREVIOUS EDITION IS OBSOLETE — EDUCATION/TRAINING REPORT

Figure 3: Computer generated AF Form 475 with entries.

```
\notcompflag{} \distgradflag{YES}
\dgnoncomp{Top 10\% of class.}
\accomplishments{
Capt Doe has achieved excellence ...  }
\qualities{
Capt Doe is a leader ...  }
\comments{
Capt Doe is an extremely ...  }
\evaluatorsigone{RONALD F. TUTTLE, LT COL,
USAF}
\evaluatorsigtwo{Air Force Institute of
Technology (AU)}
\evaluatorsigthree{Wright-Patterson AFB OH}
\evaluatorsigfour{}
\evaluatordutyone{Associate Dean}
\evaluatordutytwo{School of Engineering}
\date{16 Dec 90}
\evaluatorssan{123-45-6789}
\front
\end{document}
```

This completed form is shown in Figure 3.

Problems Along the Way

Our initial hurdle was political in nature. Getting Air Force approval for using this system was a long and involved process. Publishing is so little understood that the initial approval letter required us to use a particular computer system (the Zenith 248), but made no restrictions on the laser printer!

That vast variety of computers, versions of MS-DOS, and laser printers (especially those that were "100% compatible", but really weren't) caused us a lot of headaches. One of major physical problems is that the forms use most of an 8 1/2 × 11 page, stretching the limits of most laser printer engines. Also, LaTeX takes up a lot of memory to run. In order to get the most complex forms to run, I created a stripped down version of `latex.tex` and `lfonts.tex` to reduce the memory requirements. Many things such as sectioning, table of contents, etc. will never be used in a form, and so could be eliminated.

Conclusion

The Air Force has been very happy with the system. Although faster commercial forms packages are available, they are not free, not portable, and don't do nearly as nice a job as LaTeX on formatting the text for the blocks.

Acknowledgements

Many thanks go to all of the Air Force personnel who contributed to the AFFORMS package: Col Stan Lewantowicz and Lt Col Charlie Bisbee who with myself created the initial package, and Maj Bob Rebo, SrA Dave Weissfeld and Sgt John Rogers who added many features and distributed the program worldwide.

I also thank all of the people who create public domain TeX software. Without this we could not have been so successful, or produced such beautiful documents.