DVI-Driver for *OSF/Motif*

DVIMOTIF

Porting Report
Installation Guide
User's Guide

June 27, 1990

Christian Markus
Fahrenkrön 124
D-2000 Hamburg 71
Fed.Rep.of Germany
Tel.: (040)-643 21 73

FB Informatik
Universität Hamburg
Email: markus@rz.informatik.uni-hamburg.dbp.de

Contents

1 *DVIMOTIF* release notes

This is the first version of *DVIDECW* destined to run under the new *OSF/Motif* Graphical User Interface (GUI). The program itself was renamed to *DVIMOTIF*, the command file TEXVIEW.COM was renamed to TEXVIEWM.COM, the command file DVIDECW.COM was renamed to DVIMOTIF.COM and the command file DVILINK.COM was renamed to DVILINKM.COM. Installation and Operation should be very similar to the *DECwindows* Version of the TEX-Preview software, more information can be retrieved from the Installation and User's Guide section of this document. However, be aware that there might be still changes in the *OSF/Motif* system software, so this program is provided on an 'as is' basis. Everyone is encouraged to work with this version, any bug reports are greatly appreciated. *DVIMOTIF* is based on the field test (DXM_BL1) version of the DEC *OSF/Motif*-Toolkit, I will try to adapt it to newer versions when I can get hold of them.

The provided version of *DVIMOTIF* has all required libraries linked with it instead of using shareable libraries, so the executable program size is very large.

2 Porting *DVIDECW* to *OSF/Motif*

2.1 Introduction

The basic idea behind this program port was to find out how easy and straitforward it can be to port existing software written under the *DECwindows* GUI to *OSF/Motif*. The goal was to perform as little work and modifications as neccesary to port the program source code successfully to the *OSF/Motif* environment. This is an important goal bearing in mind that not all software which has been written under the *DECwindows* GUI is so small and easy to comprehend as DVIDECW. So an approach using the automatic conversion tools provided by DEC was the way to proceed.

Before I was able to start the conversion project, these things were essentially required:

- DIGITAL's *OSF/Motif*-Toolkit software (field test) which consists of libraries, bindings, an UIL-Compiler, the *OSF/Motif* window manager, sample programs and porting tools
- DIGITAL's *OSF/Motif*-Toolkit documentation

Before the actual work can start, the command procedure DXM_LOGICALS.COM has to be executed. It defines all neccesary symbols and logical names required to work with the *OSF/Motif*-Toolkit. As a side-effect, some previous defined symbols and logical names required for the *DECwindows*-Tookit are superseeded, they have to be re-initialized in order to continue work on *DECwindows* projects.

2.2 Using the porting tools

Within the *OSF/Motif*-Toolkit software is a collection of porting tools and a command procedure DXM_PORT.COM to call each of these tools. I converted both my .UIL and my .C file by executing `dxm_port <source path> <dest path>`. After this, nearly all logical names, types and procedure entries were automatically transferred to the corresponding *OSF/Motif*-calls. As described by the documentation, some portions of the original coding where flagged with `'>>>> Developer decision'` or `'>>>> Motif transition'` texts indicating that some manual work had to be done there.

2.3 Adapting the UIL-File

First, it turned out to be critical to use names like 'OK', 'Cancel' or 'Quit' because they are partly converted by default to a wrong type or a type cast is set on these names. This occured in my UIL-File within the GotoPageDlg dialog box with its 'OK' button. This resulted in an UIL-Compiler error, the message was `'nested instructions issued'` with no line number where this error occured. This can be very nasty with larger UIL files, so better check for special names.

In some objects, the `X` argument in the `arguments` section was not automatically converted to `XmNx`.

All `XmInformationDialog` widgets as created by the porting utilities had to be renamed to `XmMessageDialog` widgets, the `XmLabelString` argument had to be renamed to `XmNmessageString`. In all `XmBulletinBoardDialog` definitions, the `XmNtitleString` had to be converted to `XmNdialogTitle` and the `take focus` argument had to be ommited. Any occurence of `XmNlabelString` within a `XmPushButton` widget is acceptable, the `'>>>> Developer decision'` flag could be cancelled. All marked `XmNactivateCallback` functions within the menu bar definition were right, the `'>>>> Developer decision'` flag was removed here too.

In the GotoPageDlg dialog box, the `accept_focus` argument of *DECwindows* had to be removed, some of the size dimensions of this dialog box had to be adjusted. The `XmNdialogStyle` argument was added to the definition.

Some modifications were neccessary in the MainWindow definition. It was impossible to take over the scrollbar definitions used in the *DECwindows* version of the program. Instead, the required width and height had to be supplied by means of the `XmNheight` and `XmNwidth` arguments and the `XmNscrollingPolicy` argument had to be set to zero. All scrollbar definitions and references were deleted and the width and height arguments had to be defined in the `XmDrawingArea` widget too, an extra `XmNcreateCallback` definition had to be added there. Generally speaking, *OSF/Motif* has a more automatic approach towards scrollbar handling.

2.4 Adapting the C-File

Due to the changes in the MainWindow definition (namely scrollbars), all related procedures in the .C file had to be modified. The *scrollbar* callback procedure was

fully deleted, in the *output-exposed* callback procedure all code related to the resizing of the main window and the updating of the scrollbars was removed, instead the `XmDrawingArea` size is adjusted according to the zoom range. *OSF/Motif* itselfs takes care of window content's scrolling if the actual window is smaller than the size of the drawing area widget. Unfortunately, there is no chance of controlling the scrollbar's positions and size, so moving the TEX page with the Move menu commands wont be reflected in the scrollbars!

Additional code was written in the `devinit()` procedure to set the drawing area size, all string creation routines where converted in the right way but were flagged by the conversion tools, these flags were deleted.

All coding regarding the Copy operation of the Edit menu had to be revised to be compatible with *OSF/Motif*.

2.5 General problems

There were some general problems which turned up while working with the *OSF/Motif* window manager. If the Close box of any window is clicked, the program will crash. This is extremly bad if the crashed program is the *OSF/Motif* window manager. A program crash also occurs if the Eve editor's window is maximized using either the System command menu or the maximize button.

In the preliminary documentation, no hints are given which actual `#include` files are required for a specified UIL-object. The same is true with all the other include files supplied with the toolkit, a general guideline or instructions which file to include would be greatly appreciated as the conversion utilities do not provide any hints.

There was no description of available formats for the copy to clipboard functions. The only information is a reference to another document, which I did not get so far. Currently, the format is named `XYBitmap`, but it was impossible to paste any clipboard contents to another application.

3 Installation of *DVIMOTIF*

DVIMOTIF itself can be installed after the command file TEXVIEWM.COM is modified to suit the system paths of the local TEX installation. Please perform the following steps to do so:

- in *FileView*, select the Customize/Verbs and Menus option
- enter `TeX Preview` below the 'verb names' box as a new name and click on the 'Enter' button to accept it
- enter the required DCL command for the TEX-Preview entry, e.g. `<path>texviewm.com`, then click on the 'Enter' button

- select 'Applications' from the 'Menu Names' box and click on 'Add' below the 'Verbs in menu' box
- click on the 'Apply' and 'Ok' buttons to install the entry into the menu bar.
- in *FileView*, select the Customize/File Types option and install the `.DVI` file type

4 Operation of *DVIMOTIF*

4.1 Invocation of *DVIMOTIF*

4.1.1 Starting from *DECTERM*

Enter command name, options and filename as described in [2] within the *DECTERM* window.

4.1.2 Starting from *FileView/DCL Command Window*

Create a *DCL Command Window* by clicking on Utilities/DCL Command Window in *FileView*. Enter command name, options and filename as described in [2].

4.1.3 Starting from *FileView* menu bar

If the installation of *DVIMOTIF* was done properly, *DVIMOTIF* can be started by choosing the Applications/TeX Preview option from the *FileView* menu bar. A dialog box will appear and ask for the filename to be processed, after clicking on the 'OK' button a second dialog box will appear and ask for processing options.

4.1.4 Starting by double-clicking within *FileView* window

If the installation of *DVIMOTIF* was done properly, *DVIMOTIF* can be started by double-clicking on any DVI-file (Filename has the extension `.DVI`). A dialog box will appear and ask for processing options.

4.1.5 Remarks

DVIMOTIF offers the same options as the Beebe *BBN Bitgraph*-driver. However, it ignores the `-o` option (order pages) and will only display the first page (Default: 1) given in the `-o` option . All other options conforms to the description of N.F.Beebe's driver family [2].

DVIMOTIF initially operates with 69 dpi fonts which permit a full page preview in all formats. The option `-m1` (as well as zooming from the menu bar) uses 83 dpi fonts

which are quite readable and allow a full page preview in most cases (if offsets are disabled).

During page creation *DVIMOTIF* writes status reports to the *OSF/Motif* standard output window. These status reports (and the creation of a standard output window) can be suppressed by entering the `quiet` option (`-q`) in the processing options dialog box. This will switch to quiet mode which only writes error messages to the standard output window.

After invocation *DVIMOTIF* creates a window with a menu bar (for interactive commands) and two scrollbars. The window title bar contains information about the current page and resolution.

If the 'TeX Preview' is active (it has the *Input Focus*), some of the menu bar options can be operated with keyboard shortcuts which correspond to those defined by Nelson F. Beebe in [2]. To give the *Input Focus* to the 'TeX Preview' window, just click with the mouse somewhere within the window area. All shortcut key combinations are also displayed in the menu bar next to the option names.

4.2 *DVIMOTIF* menus

4.2.1 File-Menu

- About *DVIMOTIF...* will display a dialog box which contains information about the program version and copyrights.
- Quit closes all *DVIMOTIF* windows and terminates the application.

4.2.2 Edit-Menu

- Cut
- Copy copies the window contents to the Clipboard [1]
- Paste
- Clear
- Select All

The Edit Menu is implemented according to the standards defined in [5], but only the `Copy` option is currently supported.

[1] The Copy operation does not work in the current version of *DVIMOTIF*

4.2.3 Page-Menu

- Previous switches to the previous page
- Next switches to the next page
- Goto... opens a dialog box, where a page number can be entered. After clicking on 'OK' or 'GO', the selected page will be displayed.
- Zoom in magnifies the current page (by one TEX magstep)
- Zoom out reduces the current page (by one TEX magstep)
- Top of Page moves window contents to top of page
- Bot of Page moves window contents to bottom of page

4.2.4 Move-Menu

- Up moves window contents up
- Down moves window contents down
- Left moves window contents left
- Right moves window contents right

4.2.5 Options-Menu

- No offset/Use offset deactivates/activates any offset specified in the command-line
- Fast move/Slow move switches between big/small move distances
- Reset position resets window position

4.3 Dialog boxes

DVIMOTIF provides six dialog boxes:

- AlertNYI pops up, if an unimplemented program operation was selected by the user
- About*DVIMOTIF* shows information about the *DVIMOTIF* program version and copyrights
- LastpageInfo is displayed, if a page beyond the last page in the DVI file is selected
- FirstpageInfo is displayed, if a page beyond the first page in the DVI file is selected

- ZoomrangeInfo pops up, if the zoom range is exceeded
- Gotopage_box provides the possibility to enter a page number, which is to be displayed next. This dialog box is of the '*modeless*' type and remains on the screen if the 'Go' button is clicked. If the 'Cancel' or 'Ok' buttons are clicked, this dialog box will disappear. The 'Enter' key can be used instead of clicking the 'Ok' button.

References

[1] Knuth, Donald E. *The TEXbook*, Addison-Wesley, Menlo Park (California) 1989

[2] Beebe, Nelson H.F. *DVIxxx - Display TEX DVI Files on Assorted Output Devices*, University of Utah, Salt Lake City (Utah)

[3] Beebe, Nelson H.F. *A TEXDVI Driver Family*, University of Utah, Salt Lake City (Utah) 15.April 1987, Rev. 2.07

[4] *Overview of VMS DECwindows*, Digital Equipment Corporation, Maynard (Massachusetts) 1988

[5] *XUI Style Guide*, Digital Equipment Corporation, Maynard (Massachusetts) 1989

A DCL-Command files

A.1 DVIMOTIF.COM

Command file to compile/link *DVIMOTIF*

```
$ !---------------------------------------------------------------------!
$ ! Filename: DVIMOTIF.COM                                              !
$ ! Function: Compiles and links the DVIMOTIF application               !
$ ! (c) 19-06-90 by Ch. Markus                                          !
$ !---------------------------------------------------------------------!
$ !
$ ! Define symbols for OSF/Motif toolkit
$ !
$ @$disk1:[1050101]dxm_logicals.com
$ !
$ ! Defines which have to be adapted to system installation before
$ ! attempting to compile (will need modifications)
$ !
$ define/nolog dvi$decw $disk1:[1050101.includes] !modified .H files
$ define/nolog dvi$beebe $disk2:[dvi-beebe.dvi]   !standard .H files
$                                                 !from BEEBE packg.
$ !
$ ! Defines which rely on system symbols set previously (partly
$ ! defined in DXM_LOGICALS)
$ !
$ define/nolog C$INCLUDE -
"DEC$KIT$XT:",-
"DEC$KIT$XM:",-
"DEC$KIT$DXM:",-
"DEC$KIT$UIL:",-
"DEC$KIT$MRM:",-
"SYS$LIBRARY:",-
"SYS$COMMON:[DECW$INCLUDE]",-
"dvi$decw:",-
"dvi$beebe:"
$ define/nolog vaxc$include c$include
$ define/nolog X11 decw$include
$ define/nolog sys sys$library
$ define/nolog uil$include dec$kit$uil
$ !
$ ! General checkup of old errors
$ !
$ on warning then goto error
$ !
```

```
$ !
$ ! Compile the 'C'-Source to 'OBJ'
$ !
$ if f$search("DVIMOTIF.C") .eqs. "" then goto compile_uil
$ !
$ vue$set_task_label "CC DVIMOTIF"
$ vue$popup_progress_box
$ cc /define="_MOTIF" /define="DEC_MOTIF_EXTENSION" -
  /define=("OS_VAXVMS"=1) /define=("ANSI_LIBRARY"=1) dvimotif
$ !
$ !
$compile_uil:
$ !
$ if f$search("DVIMOTIF.UIL") .eqs. "" then goto do_link
$ !
$ ! Compile the 'UIL'-Source to 'UID'
$ !
$ vue$set_task_label "UIL DVIMOTIF"
$ vue$popup_progress_box
$ uil dvimotif.uil
$ !
$do_link:
$ !
$ ! Link the DEC-Windows program
$ !
$ vue$set_task_label "LINK DVIMOTIF"
$ vue$popup_progress_box
$ !
$ link /exe=$disk2:[1050101]dvimotif.exe -
 sys$input:/options
 dvimotif.obj
 unixclib/lib
 dec$kit$mrmlib/lib
 dec$kit$dxmlib/lib
 dec$kit$xmlib/lib
 dec$kit$xtlib/lib
 sys$library:decw$xlibshr/share
 sys$library:vaxcrtl/share
 dec$kit$xm:clib/lib
$ !
$ !
$ ! Change APPLY button to UPDATE button on FileView
$ !
$ vue$highlight_update
$end:
```

```
$ exit
$ !
$error:
$ vue$set_error_status
$ exit
```

A.2 DVILINKM.COM

Command file to link *DVIMOTIF*

```
$ !---------------------------------------------------------------------!
$ ! Filename: DVILINKM.COM                                              !
$ ! Function: Links the DVIMotif OSF/Motif application (neccesary if  !
$ !           DVIMotif will run under a VMS version  other than 5.3)  !
$ ! Important:There will be a linker warning message for multiply     !
$ !           defined Symbols QSORT and SYSTEM, this can be ignored   !
$ ! (c) 08-06-90 by Ch. Markus                                         !
$ !---------------------------------------------------------------------!
$ !
$ @$disk1:[1050101]dxm_logicals.com
$ !
$ vue$set_task_label "LINK DVIMOTIF"
$ vue$popup_progress_box
$ !
$ link /exe=$disk2:[1050101]dvimotif.exe -
 sys$input:/options
 dvimotif.obj
 unixclib/lib
 dec$kit$mrmlib/lib
 dec$kit$dxmlib/lib
 dec$kit$xmlib/lib
 dec$kit$xtlib/lib
 sys$library:decw$xlibshr/share
 sys$library:vaxcrtl/share
 dec$kit$xm:clib/lib
$ !
$ !
$ ! Change APPLY button to UPDATE button on FileView
$ !
$ vue$highlight_update
$ exit
```

A.3 TEXVIEWM.COM

Integration of *DVIMOTIF* in *FileView* menu bar

```
$ !---------------------------------------------------------------------!
$ ! Filename: TEXVIEWM.COM                                              !
$ ! Function: This is the interface between OSF/Motif FILEVIEW and the!
$ !           DVIMOTIF Tex Preview driver which runs in the OSF/Motif !
$ !           environment. Fileview selections are scanned and passed !
$ !           in the command line.                                      !
$ ! (c) 19-06-89 by Christian MARKUS                                    !
$ !---------------------------------------------------------------------!
$ !
$ ! These definitions supersede global definitions and have to be
$ ! corrected in other environments
$ !
$ define/nolog TEX_FONTS $DISK0:[TEX29.DVI.CM.]     !path for fonts
$ define/nolog TEX_INPUTS $disk1:[1050101.dvimotif] !path for .UID file
$ dvimotif := $$disk2:[1050101]dvimotif             !path for .EXE file
$ !
$ ! Check if files are selected in FileView
$ !
$ vue$get_selection_count
$ vue$read count
$ !
$ ! If files are selected then go on processing
$ !
$ if count .ne. 0 then goto get_file
$ !
$ ! Ask for filename
$ !
$ vue$inquire_symbol "VUE$TEXVIEW_DVI Filename:"
$ vue$read selection
$ !
$ ! Check for valid filename
$ !
$ if "''selection'".eqs. "" then goto end
$ goto do_dvi
$ !
$ ! Get Filename from FileView selection
$ !
$get_file:
$ vue$get_next_selection
$ vue$read selection
$ !
```

```
$ ! Start DVIDECW with filename argument
$ !
$do_dvi:
$ !
$ ! Extract the filename from selection
$ !
$ filename= "’’f$parse(selection,,,"NAME")’"
$ !
$ ! Check if file exists
$ !
$ if f$search("’’filename’"+".dvi") .eqs. "" then goto end
$ !
$ ! Ask for processing parameter
$ !
$ vue$inquire_symbol "VUE$TEXVIEW Parameter:"
$ vue$read parameter
$ !
$ ! Write info box
$ !
$ vue$set_task_label "DVIMOTIF ’’filename’"
$ vue$popup_progress_box
$ !
$ dvimotif ’filename’ ’parameter’
$ !
$end:
$ exit
```