

# 如何在 Linux 上建立 Motif Tools Library libXmt.a

William June 25, 2022

## 摘要

Motif Tools 是建立在 Motif Library libXm.a 基础上的一个软件工具，它使实现X-Window的应用更快捷，更方便。在网上可以下载它的 C源文件。用户必须在你自己的机器上重新编译，最后产生所需要的libXmt.a和其他utility programs. 因为这套软件开发的时间久远(1994)所以在编译的时候会产生种种错误。本文就解决方案进行必要的描述。解决方案在以下两个系统中测试过，结果都良好。

- A) VM VirtualBox上的 Linux OpenSuSE.
- B) Raspberry Pi 4 上的 Linux ubuntu MATE.

## I. Motif Tools 主要的参考书籍

Motif Tools 主要的参考书籍是 “Motif tools streamlined GUI design and programming with the Xmt library”. 该书带有 CD-ROM, 包含了 Motif Tools 的全部 C源文件. CD-ROM 上最重要的文件就是一个tar压缩文件, XMT200.TAR. 解开该文件以后, 一切需要的文件, 包括C源程序, 编译说明, 一些应用(utility)程序和测试样例程序都有了.

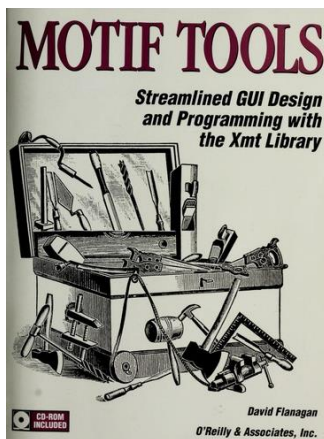


图 1. Book “Motif Tools”

详细的软件安装说明可参考该书的最后一章, Appendix A. (p. 341). 该章描写了如何解开XMT200. TAR文件以及如何在你的机器上重新编译 Motif Tools软件. XMT200. TAR. 文件解开以后, 有一个重要的文件INSTALLATION, 它描述了安装的全过程.

## II. 安装Motif Tools

### 2.1 解开 XMT200.TAR.文件

2.1.1 CD-ROM上有一个README文件告诉你怎么解开XMT200. TAR文件:

```
mkdir /usr/local/source/Xmt-2.0.0
cd /usr/local/source/Xmt-2.0.0
cp /cdrom/xmt200.tar .
tar xvf xmt200.tar
    (...tar will print lots of file names...)
rm xmt200.tar
```

2.1.2 解开 XMT200. TAR以后就看到以下各个文件和子目录:

**This distribution contains the following files and directories:**

|              |                                                     |
|--------------|-----------------------------------------------------|
| BUGREPORT    | a form and instructions for submitting bug reports. |
| CHANGES      | a list of changes since the last release            |
| COPYRIGHT    | copyright information for Xmt                       |
| INSTALLATION | how to compile and install Xmt                      |
| KNOWNBUGS    | a (short) list of bugs that I know about            |
| LICENSE      | the terms and conditions for using Xmt              |
| NO_WARRANTY  | a standard "No Warranty" disclaimer.                |
| OVERVIEW     | a list of features in Xmt                           |
| README       | this file                                           |
| SHAREWARE    | how to obtain a license for Xmt, and why you should |
| VERSION      | specifies what version of Xmt you have              |
| Imakefile    | a top-level Imakefile for building the distribution |
| Xmt.tmpl     | an imake template for use with the Imakefile        |
| Xmt/         | source code for the Xmt library                     |

|           |                                                     |
|-----------|-----------------------------------------------------|
| clients/  | Xmt client applications                             |
| contrib/  | stuff that is not part of Xmt, but is useful anyway |
| examples/ | some examples demonstrating important Xmt features  |
| man/      | ASCII formatted man pages for Xmt.                  |

## 2.2 编译 Motif Tools 库 libXmt

XMT200. TAR解开以后有一个重要文件是INSTALLATION. 该文件描述如何编译和安装Xmt. 下面是文件中描述的主要步骤:

### 2.2.1 检查和阅读Xmt. tml文件. 该文件主要定义以下各项:

| Variable        | Purpose                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------|
| DoNormalLib     | whether to build a normal, non-shared library. YES or NO.                                     |
| DoSharedLib     | whether to build ashared library. YES or NO.                                                  |
| DoDebugLib      | whether to build a debugging (-g) library. YES or NO.                                         |
| DoProfileLib    | whether to build a profiled library. YES or NO.                                               |
| XMTLIBDIR       | Where to install the Xmt library.                                                             |
| XMTINCDIR       | Where to install the Xmt header files.                                                        |
| XMTXTLIB        | Where your X libraries are installed.                                                         |
| XMTXTINC        | Where your X header files are installed.                                                      |
| XMTXMLIB        | Where your Motif library is installed.                                                        |
| XMTXMINC        | Where your Motif header files are installed.                                                  |
| XMTEXTRALIBS    | Any extra libraries needed by Motif on your system.                                           |
| OLDMOTIFDEFINES | Uncomment this line if you use Motif 1.1.0 or another<br>oldversion of Motif 1.1              |
| HPDEFINES       | Uncomment this line if you use HP's version of Motif 1.1.                                     |
| DECDEFINES      | Uncomment this line, or portions of it, if you are using<br>DECWindows or VMS.                |
| IBMDDEFINES     | Uncomment this line to work around a bug in IBM's AIX 3.2<br>version of the X11R4 Xt library. |

一般来说可以不作修改, 用原来的Xmt. tml文件即可.

### 2.2.2 用以下命令进行编译

```
cd /usr/local/source/Xmt-2.0.0 <CR>
xmkmf
make World >& makelog &
tail -f makelog
```

实际操作中，在我的机器上只用了以下两条命令：

```
xmkmf <CR>
make World <CR>
```

原理上如果一切正常，第2条命令结束以后所有需要的 Xmt 库和应用程序都已经建立完成。但现实总是不如理想，因 Motif Tools 软件的建立在几十年以前，如今的 C编译系统已经有了较大的改动。所以这些原始的C文件会产生很多错误和警告，必须一一纠正才能建成需要的服务和应用程序。

## 2.3 纠正编译 Motif Tools 库 libXmt 产生的错误

用 make World <CR> 命令编译 Motif Tools 库 libXmt 产生的错误主要有以下三类：

|   | Error Message                                          | C Files      | Line |
|---|--------------------------------------------------------|--------------|------|
| 1 | Char * sys_errlis[] Conflicting types for sys_errlis[] | AskForFile.c | 40   |
|   |                                                        | HelpNode.c   | 309  |
|   |                                                        | Include.c    | 32   |
|   |                                                        | XpmParse.c   | 321  |
| 2 | Undefined _XmTextMarginsProc                           | cli.c        | 243  |
|   |                                                        | InputField.c | 122  |
|   |                                                        | MsgLine.c    | 115  |
| 3 | Undefined MAXINT                                       | InputField.c | 122  |
|   |                                                        |              | 1075 |

表 1. 编译产生的错误主要有三类

经努力，以上错误都找到了解决方案。现在把它列成下表：

|   | Error Message                                                | Solutions                                                                                                                                            |
|---|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Char * sys_errlis[]<br>Conflicting types for<br>sys_errlis[] | Add Line:<br><code>extern __const char *__const sys_errlist[];</code><br><br>实际上这个sys_errlist[];是定义在GNU C Library<br>/usr/include/bits/sys_errlist.h |

|   |                                 |                                                                                                                                                                           |
|---|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 | Undefined<br>_XmTextMarginsProc | Add Lines:<br><code>extern void _XmTextMarginsProc(<br/>    Widget w,<br/>    XmBaselineMargins *margins_rec) ;</code>                                                    |
| 3 | Undefined MAXINT                | Add Lines:<br><code>#include &lt;limits.h&gt;<br/>const int MAXINT = INT_MAX;</code><br><br>说明：找不到MAXINT： 可以定义：<br>const int MAXINT = INT_MAX;<br>INT_MAX 定义在文件<limits.h> |

表 2. 纠正编译产生的三类错误

## 2.4 安装 Xmt 软件

2.4.1 把表1中各个出错的文件按表2中的解决方案纠错以后再重新编译.

`make World <CR>`

这时应该还能看到一些 Warning Message. 现在暂时不去管它.

2.4.2 安装 Xmt 软件

`make install <CR>`

这条命令把生成的Xmt库, 应用程序, C源程序以及\*.h文件都拷贝到预定的地方. 比如:

```
libXmt.a -> /usr/lib
*.h -> /usr/include/Xmt
mockup -> /usr/bin    等.
```

## III. 安装Motif Tools完成后的简单测试

完成 “2.4 安装 Xmt 软件” 以后就可以进行简单的测试.

### 3.1 用 mockup 进行快速测试

```
cd /usr/local/source/Xmt-2.0.0/examples/22
```

在此目录下有一个文件dialog.ad. 用 mockup 命令测试:

```
mockup dialog <CR>
```

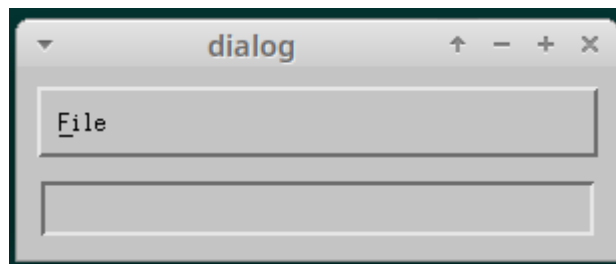


图 2. Run mockup dialog <CR>

### 3.2 用 cc 用C编译源程序测试

```
cd /usr/local/source/Xmt-2.0.0/examples/22
gcc -o dialog dialog.c -lXmt -lXm -lXt -lX11 -lXext
setenv XENVIRONMENT dialog.ad
./dialog
```

## 后记

未决问题: mockup 进行快速测试某一个程序产生 Core Dump.

在 /usr/local/source/Xmt-2.0.0/examples/20 目录下  
有一个文件 menus.ad. 用 mockup 命令测试:

```
mockup menus <CR>
*** buffer overflow detected ***: terminated
```

在此目录下还有另外一个C文件， menu. c. 用3.2节描述的命令编译后运行得到同样的错误信息：

```
*** buffer overflow detected ***: terminated
```

现将menu. c文件 main()显示如下(全部程序见篇末)：

```
void main(argc, argv)
int argc;
char **argv;
{
    XtAppContext app;
    ... ..

    menubar = XmCreateMenubar(layout, "menubar", args, ac);
    XtManageChild(menubar);

    workarea = XmCreateDrawingArea(layout, "workarea", NULL, 0);
    XtVaSetValues(workarea,
                  XmtNlayoutWidth, 400,
                  XmtNlayoutHeight, 300,
                  NULL);
    XtManageChild(workarea);

    ac = 0;
    XtSetArg(args[ac], XmtNitems, popup_items); ac++;
    XtSetArg(args[ac], XmtNnumItems, XtNumber(popup_items)); ac++;

    Dump Here ==>    popup = XmCreatePopupMenu(workarea, "popup", args, ac);

    XtRealizeWidget(toplevel);
    XtAppMainLoop(app);
}
```

经过分析，实际上问题是产生在 Xmt下层，Xmt function XmCreatePopupMenu() calls libXt.a 中的 fuction XtCreateWidget() 时出错。与 libXmt.a 没有关系。关于 XtCreateWidget() Core Dump，网上已经有多人讨论。看来毛病是出在 X-Window, Motif library 等与 Linux 操作系统的C编译器不合配的原因。需要重新编译 X Windows下面的程序库等等。这个工作量就比较大了。

<https://motif.ics.com/forum/developers/segmentation-fault>

Submitted by Anonymous (not verified) on Fri, 06/15/2012 - 12:07

I am developing a big application on RH7.3 Linux using g++ 2.93 compiler and also using open motif.

Sometimes, when I try to create a error dialog window, ui says

**it core dumps at**

(In the reverse order)

XmCreateErrorDialog

**XtCreateWidget**

....

....

....

XtFree

free

chunk\_free

I have chekced all the parameters which are passed to XmCreateError Dialog, they are valid and correct, and no memory corruption is happening there.

But everytime, this happens, Please suggest how to detect, where the problem is

Thanks in advance

B. Prabakaran

Hyderabad

< Tooltips and repaints and exposures... some explanations? How to create your own motif-resources (not the predefined.) >

Mon, 11/03/2003 - 15:30#2

Segmentation Fault

**There might also be a Motif/Glibc versioning problem here.** The compiler you are using suggests that you are using an older Linux systems (perhap RH 6.1?). There was a glibc incompatibility introduced in the RH 7 series. Basically, mixing libraries compiled against different glibc would segfault. Try doing a Hello world button that just exits the application. If that works, you OK. If that also segfaults, you got a version problem.

If you want to stay on that release of Linux, you will need to yank the source code and do your own build. There are no recent builds of OM on Linux systems that old.

Mark

## menu.c 全部程序

```
/* menu.c */
```

```
#include <Xmt/Xmt.h>
```

```
#include <Xmt/Layout.h>
```

```
#include <Xmt/Menu.h>
```

```
#include <Xm/DrawingA.h>
```

```
/* dummy procedures to call */
```

```
static void NewFile(/*Widget w, XtPointer tag, XtPointer data*/) {}
```

```

static void OpenFile(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void SaveFile(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void SaveAs(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void Quit(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void About(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void ContextHelp(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void move(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void rotate(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void scale(/*Widget w, XtPointer tag, XtPointer data*/) {}
static void toggle_grid(/*Widget w, XtPointer tag, XtPointer data*/) {}

static XmtMenuItem file_menu_items[] = {
    {XmtMenuItemPushButton, "New", 'N', "Meta<Key>N", "Meta+N", NewFile },
    {XmtMenuItemPushButton, "Open...", 'O', "Meta<Key>O", "Meta+O", OpenFile},
    {XmtMenuItemPushButton, "Save", 'S', "Meta<Key>S", "Meta+S", SaveFile},
    {XmtMenuItemPushButton, "Save As...", 'A', "Meta<Key>A", "Meta+A", SaveAs},
    {XmtMenuItemSeparator},
    {XmtMenuItemPushButton, "Exit", 'x', "Ctrl<Key>C", "Ctrl-C", Quit},
    {XmtMenuItemEnd}
};

static XmtMenuItem view_menu_items[] = {
    {XmtMenuItemToggleButton, "By Icon", 'B', NULL, NULL, NULL, NULL, NULL,
        "view_by_name", "By Name", 'B'},
    {XmtMenuItemToggleButton, "Sort by Size", 'S', "Meta<Key>Z", "Meta+Z",
        NULL, NULL, NULL, "sort_by_size"},
    {XmtMenuItemEnd}
};

static XmtMenuItem help_menu_items[] = {
    {XmtMenuItemPushButton, "About", 'A', NULL, NULL, About},
    {XmtMenuItemPushButton, "On Context", 'C', "Meta<Key>H", "Meta+H",
        ContextHelp},
    {XmtMenuItemEnd}
};

static XmtMenuItem menubar_items[] = {
    {XmtMenuItemCascadeButton + XmtMenuItemTearoff,
        "File", 'F', NULL, NULL, NULL, NULL, file_menu_items},
    {XmtMenuItemCascadeButton + XmtMenuItemTearoff,
        "View", 'V', NULL, NULL, NULL, NULL, view_menu_items},
    {XmtMenuItemCascadeButton+XmtMenuItemHelp+XmtMenuItemTearoff,
        "Help", 'H', NULL, NULL, NULL, NULL, help_menu_items},
};

static XmtMenuItem popup_items[] = {
    {XmtMenuItemLabel, "@fBOperations"},
    {XmtMenuItemSeparator},
    {XmtMenuItemPushButton, "Move", 'M', "Ctrl<Key>M", "Ctrl-M", move},
    {XmtMenuItemPushButton, "Rotate", 'R', "Ctrl<Key>R", "Ctrl-R", rotate},
    {XmtMenuItemPushButton, "Scale", 'S', "Ctrl<Key>S", "Ctrl-S", scale},
    {XmtMenuItemSeparator},
    {XmtMenuItemLabel, "@fBOptions"},
    {XmtMenuItemSeparator},
    {XmtMenuItemToggleButton+XmtMenuItemOn, "Show Grid", 'G',

```

```

    "Ctrl<Key>G", "Ctrl-G", toggle_grid, NULL, NULL, NULL, "Hide Grid", 'G'}
};

void main(argc, argv)
int argc;
char **argv;
{
    XtAppContext app;
    Widget toplevel;
    Widget layout, menubar, workarea, popup;
    static Boolean name, sort;
    Arg args[5];
    int ac;

    toplevel = XtAppInitialize (&app, "Test", NULL, 0,
#ifdef X11R5
                                (Cardinal *)&argc,
#else
                                &argc,
#endif
                                argv, NULL, NULL, 0);

    XmtVaRegisterSymbols("view_by_name", XtRBoolean, sizeof(Boolean), &name,
                        "sort_by_size", XtRBoolean, sizeof(Boolean), &sort,
                        NULL);

    layout = XmCreateLayout(toplevel, "layout", NULL, 0);
    XtManageChild(layout);

    ac = 0;
    XtSetArg(args[ac], XmtNitems, menubar_items); ac++;
    XtSetArg(args[ac], XmtNnumItems, XtNumber(menubar_items)); ac++;
    menubar = XmCreateMenubar(layout, "menubar", args, ac);
    XtManageChild(menubar);

    workarea = XmCreateDrawingArea(layout, "workarea", NULL, 0);
    XtVaSetValues(workarea,
                  XmtNlayoutWidth, 400,
                  XmtNlayoutHeight, 300,
                  NULL);
    XtManageChild(workarea);

    ac = 0;
    XtSetArg(args[ac], XmtNitems, popup_items); ac++;
    XtSetArg(args[ac], XmtNnumItems, XtNumber(popup_items)); ac++;
    popup = XmCreatePopupMenu(workarea, "popup", args, ac);

    XtRealizeWidget(toplevel);
    XtAppMainLoop(app);
}

```