

China-pub.com

下载

## 附录B 安装X Window窗口系统

### B.1 安装X Window的提示

安装X Window(以下简称X)的注意事项如下：

- 尽快将所有文件复制下来。理想的状况是在开始安装系统之前，你可以将光盘上所包含的文件打印出来。
- 首先，对已有的配置文件尽量少做更动。如果你是第一次安装 X，我们推荐你使用缺省的目录来安装那些系统的各个不同的部分，只有在有确实需要更动目录名称时，才考虑重新安装。（如果你不这样做，当有些问题发生时，你将无法知道，它真的是一个问题还是你安装时犯了一个小错误。）
- 如果你是用这次发行的版本来建立一个标准的系统，你几乎不需要做任何更动便可完成安装——配置文件已正确地设定好了。
- 当开始安装时，对大多数的系统而言，不需明确指定系统的形态，它的建立是根据 C 语言的预处理器来定义系统名称，从而自动找到的。
- 在你熟悉core版之前不要安装contrib 软件，因为它安装费时，且易产生问题，当熟悉系统之后，你可能总会修改一些配置参数。
- 如果真的需要使用不同缺省值的配置，请阅读文件 \$TOP/util/imake.includes/ README ，你可得到关于如何改动的信息。

### B.2 建立此版本

本次发行的软件分为两类：一部分核心软件由 X协会提供，另一部分则由其他厂商提供。要使核心软件这一部分能在大多数的地方重新配置、编译和安装存在一定的难度，而那些其他厂商开发的部分，也就是说，未经 X协会标准编译和测试的部分则只能自行逐一建立。在 util/scripts中的ximake.sh可能对Imakefiles和Makefiles 非常有用。

几乎所有在核心软件中的 makefiles 都是由名叫 imake 的实用程序所产生的。所有的 Makefiles 初始版本均包含了在哪些地方不能使用 imake（无疑地，在特定的机器上需要修正）。无论如何，它可以使用目录 util/imake.includes/中的文件site.def和\*.macros中的参数，来重新配置核心软件。

### B.3 安装摘要

欲加载和安装此次发行的X Window System，将需要：

- 1) 读完本次发行的备忘录。
- 2) 建一个你读取软件的目录（通常名称类似 /usr/local/src/X或/src/R31）。核心软件将需要大约30 MB的空间，用户软件需要超过80 MB的空间。
- 3) 既然用户软件必须自行建立，你可以等到以后再加载。
- 4) 阅读文件util/imake.includes/README以了解如何配置特定的X系统。同样地，依照目

录服务程序/ddx/ 中README文件来计划安装服务程序。

5) 如果你计划在多部机器上编译这个版本并且要有一个分类文件系统, 你可能希望每一个目的机器上依照util/scripts/ln\_dir.sh 的说明来建立符号连接树, 这样就允许所有的机器使用 X 的同一组原始程序。

6) 如果你是在 Macintosh 机器上安装, 确定你是从目录服务程序 /ddx/macII 执行 R3 setup.sh 程序。它用来在 util/cpp 中的公共源代码中建立一个 C 的预处理器, 和修正一些放错位置的系统文件。如果你是在一部 SUN 计算机上, 确定你在文件 util/imake.includes/Sun.macros 的上面设定了四个 OS 的参数。这样作可防止在不同的 SunOS 编译器下产生的错误。如果你是在 Apollo 上安装, 确定你是使用 9.7.1 或更后面版本的 C 编译器, 否则, 服务程序将无法正确编译。

7) 确定你已遵循所有和机器特性有关的指示, 并且 imake 也是依你的机器的特性而配置的。

8) 当你完成了配置, 你可以准备安装 core 版。注意那些在 util/imake.includes/ 中的 .macros 文件。它们在接近顶端有一行设定一个 make 的变量名称为 BOOTSTRAPCFLAGS。如果这个变量值是空白, 你可以用下列的命令开始安装:

```
% make World > & make.world &
```

如果不是空白, 你必须将此定义附加到命令行上。这可用 imake 对所有的编译器设定特定的 cpp 符号。(如果你在不同的系统安装, 见 util/imake/imake.c) 在所有的 core 版提供的 macro 文件中, 只有 A/UX 需要这个标志:

```
% make BOOTSTRAPCFLAGS=-DmacII World >& make.world &
```

9) 当 make 完成后, 检查日志文件是否有问题。如果没有严重的问题, A/UX 的用户可以忽略有关枚举类型冲突的编译警告, Apollo 的用户可以忽略最优化的警告。

10) 如果一切正常且建立了正确程序, 测试一些不同的程序 (如 server、xterm、xinit、etc)。你也许需要在根目录执行 server 或 xterm, 如果你执行上有问题, 第二部的工作站或终端机将很有用。

11) 对你旧版的 X 头文件、二进制文件、字体、库等作备份。

12) 到 build tree 的顶端并键入:

```
% make install >& make.install
```

在大多数的系统上, xterm 程序必需把 setuid 安装到根目录, xload 程序必需把 setgid 安装到那些属于 /dev/kmem 的文件中。

13) 如需要安装手册, 在 build tree 顶端键入:

```
% make install.man
```

14) 如果你要建立和安装 lint libraries, 在 build tree 的顶端输入:

```
% make install.ln
```

如果是第一次安装 X, 可能也需要以下的步骤。检查一下在 server/ddx 目录中不同的 README 文件中附加的指示。

15) 加入设备驱动程序或重新配置内核。

16) 建立附加的虚拟终端机。查看操作系统说明 /dev/MAKEDEV 并询问系统管理员相关的细节部分。

17) 阅读有关新的显示管理器 xdm 的手册, 这个程序提供了一个自动执行 X 的方法, 同时为了初学者提供了很多非常好的界面符号。

18) 确定所有的 X11 的用户的搜索路径中均为 BINDIR 的目录 (通常为 /usr/bin/X11)。

## B.4 操作系统需求

X受欢迎的原因之一是与操作系统无关。虽然我们讨论的样例均为在 UNIX系统下，但有许多非UNIX操作系统的制造厂商也提供X。本次发行的服务程序已在下列系统建立：

```
4.3+tahoe
Ultrix 3.0 FT2 ( Ultrix 2.0 )
SunOS 3.4
HP-UX 6.01
Apollo Domain/IX 9.7
IBM AOS 4.3
A/UX 1.0
```

如果使用较早的版本，可能招致一些麻烦，特别是服务程序将无法在 IBM 4.2A第二次发行版上执行。在server/ddx/中不同的README文件描述了对编译器、函数库、预处理器等等的特定需求。例如前述的A/UX 1.0的用户将需要建立一个新版的C预处理器，而Apollo的用户将需要9.7.1版的C编译器。

你应在使用X前先确定你的网络系统和相互通信实用程序作用正常。如果像talk和rlogin这样的程序无法工作，X大概也不行。

## B.5 使用符号链接

本发行版使用符号链接以避免文件重复。如果你在未提供配置文件情况下建立本发行版，你必须在util/imake.includes/\*.macros文件中检查LN配置参数。如果操作系统不提供软件链接，LN需被设定为建立硬链接或拷贝原始文件。

如果需要在安装后将本发行版移动到其他的机器，可以用tar替代CP或rcp以保留日期和链接，通常可用下列的命令完成：

```
% (chdir/usr/local/src/X;tar cf - .) | \
rsh othermachine " (chdir/moredisk/X; tar xpBf -)"
```

可向系统管理员咨询以获得帮助。

## B.6 配置本发行版

这个发行版广泛使用 imake 实用程序，它可从特定机器的Imakefiles中产生机器特有的Makefiles。假如你不在乎可移植性，虽然我们建议你使用imake与makedepend，但是仍可以使用makefiles。

imake的配置文件在util/imake.includes目录中。makefiles文件是由一个称做imake.tmpl的模板文件、一个机器特有的macros文件与一个站点特有的site.def文件所产生的。除极少的例外情况，配置参数应该为cpp符号。它可以在每个服务器的基础上定义，或在给定站点里的所有服务器上定义。不应该修改此模板文件。

util/imake.includes/README文件指出了每一个配置参数以及它可以设定的值。缺省值的选择是考虑到了可适用于多种机器且易于维护。站点特有的配置应在site.def文件内使用下列语法描述：

```
#ifndef BuildParameter
#define BuildParameter site-specific-value
#endif
```

## B.7 编译本发行版

配置参数设定后,你应该可以在构造树的最上层用下列命令来编译 core 软件:

```
% make World > &make.world &
```

若在建立过程中,将 make.log 这个特殊文件删除,则请不要将输出重定向到 make.log 中。依照使用机器的不同,这个步骤将花费 2 ~ 12 小时,且在大多数的机器上皆能顺利完成。

在所有 Makefiles 与相关性都建好之后,你必须重新启动以便使编译发生作用。请在构造树的最上层输入下列命令:

```
%make -k > &make.out &
```

如果以后决定改变配置参数,你将需要作另一个完整的编译过程。

## B.8 安装发行版

假如每一个编译皆成功,可以输入下列命令,从建造树最上层安装软件:

```
# make install
```

假如你不想安装在根目录,你必须建一些可写入的目录并安装它们,然后检查在 BINDIR 目录(通常是 /usr/bin/x11) 内的 xterm 与 xload 的所有权与保护模式。xterm 必须把 setuid 安装至根目录以便它能设定虚拟终端机与更新 /etc/utmp。xload 需要把 setuid 安装至根目录,或把 setgid 安装至拥有文件 /dev/kmem 的目录以使它们可以取得系统的平均负载。

假如 /etc/termcap 与 /user/lib/terminfo 数据库没有 xterm 的入口,可参考在客户机/xterm 目录内的样本入口。System V 的用户将需要用 tic 实用程序编译 terminfo 入口。

假如你计划使用 xinit 程序去执行 X,你可能会想产生一个将指定的 X 指到适当服务程序的链接(通常的名称像 /usr/bin/x11/ 目录里的 Xmachine)。

假如你想安装使用手册,请检查在 util/imake.includes/ 里的 MamDirectoryRoot、ManDir 与 LibManDir 三个配置参数,并在构造树的最上层输入:

```
# make install.man
```

假如你喜欢建立与安装 lint 程序库,在构造树的最上层输入下列命令:

```
# make install.ln
```

最后,确定所有用户在它们的 PATH 环境变量都有 BINDIR (通常是 /usr/bin/x11/)。

## B.9 内核与特殊文件的注意事项

在某些机器上,如果出现了一个新的设备驱动程序,则必须重建内核或至少经过重新配置。假如你以前从未执行 X,你可能需要确定在你的内核配置文件中的 csr 位置能与你的硬件相匹配。另外,你应该确定在系统启动时内核会自动配置显示器。

你可能需要为显示器、鼠标或键盘建立特殊的设备。例如:

```
# /etc/mknod/dev/bell c 12 2 # for bell on Sun
```

```
# MAKEDEV displays # for displays on the RT/PC
```

应把显示器设备文件上的保护模式设定成只有服务程序可以打开它。假如服务程序是以 /etc/init 启动,这个保护可以被根用户读/写,任何其他的人不能存取。

更详细的资料请看服务程序 /ddx/ 目录里的 README 文件与手册。

## B.10 测试本发行版

即使你计划使用 xdm 执行 X,你也应该要从另一个终端机执行它,以便检查每个安装部分

是否已安装且正常工作。

来自服务程序的错误信息将会显示在终端机上，而不是写到 `xdm-errors` 或 `/usr/adm/X?msgs` 文件中(? 是显示器的号码)。

测试服务程序最简单的方法是先进入 `/usr/bin/x11` (或任何你已安装 X 程序的地方)然后执行 `xinit`, 如下:

```
% cd/usr/bin/X11
% xinit
```

你应该可以看到一个灰色的带状图案覆盖屏幕、一个形状像“X”的光标追踪着鼠标指针以及一个终端机模拟窗口。如果没有这样的显示，请按下列方式检查：

1) 假如没有显示出灰色屏幕背景，检查是否有在 `server/ddx/` 子目录内的 `README` 里描述的任何特殊设备文件的使用权 (通常是保存在 `/dev/` 中)。

2) 假如背景出现，但光标仍是白色方形。请确定字体是否已安装 (特别是 `usr/lib/ X11/fonts/misc/` 里的字体 `cursor.snf`。见配置参数 `DefaultFontPath`)。并确定在你的每一个字体目录都有一个文件叫做 `font.dir`。这个文件是由 `mkfontdir` 程序产生的。服务程序用它来找出一个目录里的字体。

3) 假如光标出现但不追踪鼠标指针，请确定相应特定设备文件 (像 `/dev/mouse`) 已安装 (见服务程序的 `README` 文件)。

4) 假如服务程序启动但不久之后变成黑色，表示启动的客户程序 (`xterm` 或 `xdm`) 停止了。请确定 `xterm` 安装在根目录并建立足够的虚拟终端机。假如你正执行 `xinit`，且在你的 `home` 目录一个有叫做 `.xinitrc` 的文件，确定它是可执行的而且上一个它所启动的程序在前台执行 (亦即命令行尾端没有一个 `&` 符号)。否则，`.xinitrc` 将立刻完成，这也是 `xinit` 所假设的以及你所想要的。只要有正确工作的初始窗口，试着从 `xterm` 执行其他的程序。若想用 `uwm` 窗口管理器定位一个新窗口，可在闪动的方框出现时按下按钮 1 (通常在鼠标的最左边按钮)：

```
% xlock -g 200x200-0+0 &
% uwm &
% xlogo &
% xeyes &
...
```

X Window 现在可以使用了，读一下手册，看看新字体，你会从中得到乐趣。

## B.11 建立额外的虚拟终端机

因为每个 `xterm` 需要一个不同的虚拟终端机，所以你应该给它们建立很多个虚拟终端机 (在一个小型多用户系统你可能至少要 32 个)。每个 `pty` 有两个设备，一个主控，一个从属，它们通常叫做 `/dev/tty[pqrstuv][0-f]` 与 `/dev/pty[pqrstuv][0-f]`。假如你没有最少的 `p` 与 `q` 行配置，你应该要求系统管理员加上它们。这通常是以 `/dev` 里执行的 `MAKEDEV` script 来完成：

```
# cd/dev
# ./MAKEDEV pty0
# ./MAKEDEV pty1
```

## B.12 从 `/etc/rc` 启动 X

本发行版提供一个可以从系统启动文件 `/etc/rc` 中 执行 X 服务器的实用程序，叫做 `xdm`。由于考虑到可以很容易地适应不同工作站的需求，`xdm` 可以周到地维护服务程序的执行，提示输入用户名与密码并管理用户的对话。当前样本配置使用外壳脚本，以提供相当简单的环境。

xdm 灵活性的关键是它广泛使用资源,允许站点管理员快速地测试不同的设置。当 xdm 启动时,它读入一个配置文件(缺省是/usr/lib/X11/xdm/xdm-config 但可以用命令行指定-config)列出不同的数据文件、缺省参数以及启动与关机时执行的程序。因为它使用标准的工具箱资源文件格式,所以任何可以在 xdm-config 文件设定的参数也可以在命令行使用 -xrm 选项指定。

缺省结构如下:

```
DisplayManager.servers: /usr/lib/X11/Xservers
DisplayManager.errorLogFile:/usr/lib/X11/Xdm-errors
DisplayManager*resource: /usr/lib/X11/Xresource
DisplayManager*startup: /usr/lib/X11/Xstartup
DisplayManager*session: /usr/lib/X11/Xsession
DisplayManager*reset: /usr/lib/X11/Xreset
```

服务程序文件包含要启动的服务器列表。 errorLogFile 是 xdm 重定向输出的文件。 resource文件包含 xdm Login 窗口缺省的资源。特别的地方是可以指定特殊键的顺序(在 xlogin\*login.translations resource), startup文件应该是一个程序或可执行的脚本,它可以在用户打入正确的密码之后执行。它对站点进行初始化、记录等的挂接。 session 入口是用于指定会话管理程序或可执行脚本的名称。假如用户在主目录内没有一个可执行的 .xsession 文件,有一个简单的版本可提供一个简单的 xterm 窗口与 uwm 窗口管理器。最后,在用户注销之后,将执行一个复位程序或可执行的脚本。如果希望使用缺省配置执行 xdm,在boot文件加入下面一行(通常叫做 /etc/rc 或 /etc/rc.local):

```
/usr/bin/X11/xdm &
```

毫无疑问,大多数站点想建立它们自己的配置。我们建议你们将站点特有的 xdm-config 文件与其他的 xdm 文件放在不同的目录。假如将文件保存在 /usr/local/lib/xdm中,可用下面的命令启动xdm:

```
/usr/bin/X11/xdm -config /usr/local/lib/xdm-config &
```

许多服务程序都假设它们是尝试读取键盘的唯一程序,这样键盘设定成非阻断式 I/O。不幸的是,某些 /etc/getty 的版本(特别是A/UX)将立刻显示一个零长度读取的连续流,因为它解释成 EOF 文件尾指示。最后 /etc/init 将取消登录,一直到某人打入下列命令:

```
# kill -HUP 1
```

在 A/UX 下,一个变通的方法是禁止从控制台登录,且从 /etc/inittab中执行 xdm。然而,请确定你存有一个旧的 /etc/inittab 拷贝,以备在出错状况以及必须从网络上或单用户模式恢复登录的时候使用。

另一个较少为人彻底研究的是如何设定一个帐号,它的外壳是在client/xdm/ 里的 xdmshell 程序。缺省状况下,这个程序不会安装,所以站点管理员需要特别查看它是否符合需求。xdmshell 实用程序确定它执行于适当的终端机型号、启动 xdm、等待它完成与重置控制台(如果需要的话)。假如 xdm 资源文件包含一个终止——显示器动作的对应,就像下列命令:

```
xlogin*login.translations: #override Ctrl<Key>R: abort-display()
```

然后在 xdm login 窗口按下指定键(上例是 <Ctrl-R>), 控制台即可以被恢复。

xdmshell 程序通常安装在根目录,但仅能被一些特殊的用户执行。这样以 xdmshell作为外壳的帐号的人是唯一的:

```
% grep xdm /etc/passwd
X:aB9i7vhDVa82z:101:51:Acount for starting up X: (contd.)
/tmp:/etc/xdmshell
% grep 51 /etc/group
xdmgrp:*:51:
```

```
% ls -g /etc/xdmshell
-rws--x--- 1 root xdmgrp 20338 Nov 1 01:32 /etc/xdmshell
```

假如 xdm 资源不曾有一个键对应至 abort-display() 动作, 一般用户将没有方法直接登录到控制台。

### 1. 支持旧系统——从 /etc/init 启动X

**警告** 下面所提供的适合较老的系统且在未来的发行版将不再提供。

Ultrix 与 4.3bsd 使用 /etc/ttyss 配置文件的一个新的扩充格式, 它允许你指定一个窗口系统与初始化时被执行的程序。虽然一般人比较喜欢使用 xdm, 但系统仍然提供可在 xterm 里启动 X 与一个从 /etc/ttyss 初始的 xterm 窗口。

### 2. 建立 ttyv 终端机

因为大多数的 /etc/init 版本需要给每个 /etc/ttyss 的入口一个真实的终端行, 所以你必须给每一个显示器指定一个虚拟终端机。虽然正常状态下 xterm 动态地配置一个 pty, 但是 -L 选项可以强迫它使用从 /etc/init 传过来的虚拟终端机。

依照协议, 最高的小设备号码改名为 [pt]tyv0, 次高的改为 [pt]tyv1, 依此类推。最高的被改为最低的, 可使他们不以正常地从低至高的搜索方法 (大多数程序配置一个 pty 的方法) 取得。在仅有 p 与 q 的小系统, 下面的命令可用以将两个显示器设定成 v 终端机。

```
# cd /dev
# mv ttyqf ttyv0; mv ptyqf ptyv0
# mv ttyqe ttyv1; mv ptyqe ptyv1
```

从系统管理员那里可以获得更多的帮助。

### 3. 在 /etc/ttyss 中增加更多的窗口入口

只要你已经改变了虚拟终端机的名称, 就能为它们在 /etc/ttyss 增加一个入口。再次提醒这个仅对具有以新的 4.3bsd 格式启动窗口系统的系统有效, 不能在使用 4.2bsd 小入口格式与 /etc/ttytype 的较老系统上使用。

x 服务程序手册里包含关于设定 /etc/ttyss 入口的更详细叙述。通常, ttyv 置于文件的底部且类似如下:

```
ttyv0"/usr/bin/X11/xterm -L -geometry 80x24+0+0 -display :0"(contd. )
xterm off window="/usr/bin/X11/X :0"
```

注意, 给 X 服务程序命令的服务程序号码的参数必须在前面置一个冒号。额外的命令行选项可以在 xterm 命令行或 X 命令行指定。然而, 许多 init 的版本具有相当小的程序名称缓冲区, 限制了入口的长度。也有某些版本不允许在入口里有 pound 记号, 这意谓着不能指定任意数目的颜色, 这也是为什么要使用 xdm 的原因。

只要你曾经增加或改变入口, 你必须通知 init, 告诉它重读 /etc/ttyss 与重新启动。在根目录下打入下列命令即可以达到这个目的:

```
# kill -HUP 1
```

这将会中止命令行上存在的进程, 所以应该只能由系统管理员去执行它。