

---

# X

X Window System est un système de fenêtrage qui tourne sur des ordinateurs avec écran. Le serveur distribut l'entrée utilisateur et accepte les demandes de sortie pour divers programmes clients au travers de différents canaux de communication inter-process. Les clients peuvent être lancés de manière transparente depuis d'autres machines. X supporte le sous-fenêtrage hiérarchique et les opérations graphique et texte, sur des afficheurs couleur et monochrome.

## Démarrage

Il y'a 2 principaux moyen de démarrer un serveur X et un jeu d'applications clients initial :

**Display Manager** : xdm, gdm ou kdm. Ce programme est typiquement lancé par le système au boot et gère le serveur et les connexions utilisateur.

**xinit** : Utilisé pour démarrer le serveur et un premier client X sur des systèmes qui ne peuvent démarrer directement X depuis /etc/init ou dans des environnements qui utilisent plusieurs système de fenêtrage.

## Noms d'affichage

Du point de vue d'un utilisateur, tout serveur X a un nom d'affichage sous la forme **hostname :displaynumber.screennumber**

Cette information est utilisée par l'application pour déterminer comment il devrait se connecter au serveur et quel écran il devrait utiliser.

**hostname** Spécifie le nom de la machine où l'affichage est physiquement connecté

**displaynumber** réfère à une collection de moniteur qui partagent un jeu commun de périphériques d'entrée. chaque affichage est numéroté en partant de 0

**screennumber** Spécifie le moniteur. Chaque moniteur est numéroté en partant de 0.

Sur les systèmes POSIX, le nom d'affichage est stocké dans la variable **DISPLAY**. Pour se logger sur une autre machine sur le réseau, il faut définir manuellement cette variable pour pointer sur votre afficheur. ssh peut être utilisé pour lancer des programmes X à distance, il définit correctement cette variable.

X écoute les connexions sur différents canaux de communication. pour déterminer le type de canal, il utilise le hostname. X supporte généralement les types de connexion suivants :

**local** La partie hostname devrait être vide. (exemple :0, :0.1)

**tcpip** Le hostname devrait être un nom d'hôte ou une IP. ex : x.org :0, expo :0, [ : :1 ] :0, 198.112.45.11 :0, bigmachine :1, et hydra :0.1)

## Contrôle d'accès

---

Un serveur X peut utiliser plusieurs types de contrôle d'accès :

MIT-MAGIC-COOKIE-1 Cookie en plain/text

XDM-AUTHORIZATION-1 Clé privé, DES.

xdm initialise le contrôle d'accès pour le serveur et place les informations dans un fichier accessible à l'utilisateur. Normalement la liste des hôtes depuis lesquels les connexions sont toujours acceptés devrait être vide, donc seul les clients explicitement autorisés peuvent se connecter. En ajoutant une liste d'hôte avec xhost, le serveur n'effectue plus aucune autorisation depuis ces machines.

Le fichier depuis lequel Xlib extrait les autorisations peut être spécifié avec la variable **XAUTHORITY**, défaut : **.Xauthority** dans le home de l'utilisateur. Pour gérer les fichiers d'autorisation, utiliser xauth. Les fichiers sont indépendants des machines, ce qui simplifie l'utilisation de nombreuses machines avec des homes partagés.

## Spécification de géométrie

De nombreux programmes X ont l'option **-geometry WIDTHxHEIGHT+XOFF+YOFF** pour spécifier une taille préférée et un emplacement dans la fenêtre principale. La syntaxe est la suivante :

**+XOFF** Le côté gauche de la fenêtre est placé à XOFF pixel du bord gauche de l'écran

**-XOFF** Le côté droit de la fenêtre est placé à XOFF pixel du bord droit de l'écran

**+YOFF** Le côté haut de la fenêtre est placé à YOFF pixel du bord haut de l'écran

**-YOFF** Le côté bas de la fenêtre est placé à YOFF pixel du bord bas de l'écran

Les offsets doivent être donnés en paire :

**+0+0** Coin supérieur gauche

**-0+0** Coin supérieur droit

**-0-0** Coin inférieur droit

**+0-0** Coin inférieur gauche

Exemple de placement de fenêtres au milieu de l'écran :

**xterm -fn**

**6x10 -geometry 80x24+30+200 &**

**xclock -geometry 48x48-0+0 &**

**xload -geometry 48x48-96+0 &**

**xbiff -geometry 48x48-48+0 &**

## Gestionnaire de fenêtre

La couche de fenêtres à l'écran est contrôlée par un programme spécial appelé le window manager. Xorg fournit twm.

## Noms de fonts

Les collections de caractères pour afficher du texte et des symboles dans X sont des fonts. Un font contient généralement des images qui partagent une apparence commune. Similairement, une collection de fonts sont une base de type commun ( les variations sont généralement appelées roman, bold, italic, bold italic, etc.)

---

le serveur X peut obtenir les fonts depuis des fichiers individuels dans un répertoire du système, ou depuis un ou plusieurs serveurs de font, ou un mixe des 2 (peut se changer avec xset).

Les fichiers de font bitmap sont créés en compilant une description de font textuelle en forme binaire en utilisant **bdftopcf**. Les bases de font sont créé en lançant **mkfontdir** dans le répertoire contenant la source ou les version compilé des fonts. Par exemple pour ajouter un font dans un répertoire privé, les commandes suivantes peuvent être utilisées :

```
cp newfont.pcf ~/myfonts
```

```
mkfontdir ~/myfonts
```

```
xset fp rehash
```

**xfontsel** et **xlsfonts** peuvent être utilisés pour naviguer dans les fonts disponibles sur un serveur. le serveur supporte les wildcard des noms de font, donc :

```
-adobe-courier-medium-r-normal-10-100-75-75-m-60-iso8859-1
```

peut être abrégé en :

```
-*-courier-medium-r-normal-*-*-*-*-*iso8859-1
```

## Serveur de fonts

Une des forme suivantes peuvent être utilisé pour nommer un serveur de font qui accepte les connexions TCP :

**tcp/hostname :port**

**tcp/hostname :port/cataloguelist**

exemple : **tcp/x.org :7100**, **tcp/198.112.45.11 :7100/all**

## colorName

Beaucoup d'applications fournissent une manière d'adapter les couleurs de divers éléments dans le texte et graphiques qu'ils affichent. Une couleur peut être spécifiée par un nom de couleur abstraite (red, blue), ou par une représentation numérique (RGB). Les noms abstraits de couleur peuvent être recherchés dans une base de donnée. Xlib recherche des bases côté client, ou dans la base de données du serveur (/usr/share/X11/rgb.txt).

une spécification de couleur numérique consiste d'un espace de nom de couleur et un jeu de valeur :

```
<color_space_name> :<value>/.../<value>
```

Une spécification de Périphérique RGB est identifié par le préfixe "rgb :" et a la syntaxe suivante :

```
rgb :<red>/<green>/<blue>
```

```
<red>, <green>, <blue> := h | hh | hhh | hhhh
```

où **h** est une valeur hexadécimale

---

Les 8 couleurs primaires peuvent être représentés :

**black** rgb :0/0/0  
**red** rgb :fff/0/0  
**green** rgb :0/fff/0  
**blue** rgb :0/0/fff  
**yellow** rgb :fff/fff/0  
**magenta** rgb :fff/0/fff  
**cyan** rgb :0/fff/fff  
**white** rgb :fff/fff/fff

une intensité RGB est représentée par le préfix "rgbi" : **rgbi** :<red>/<green>/<blue>

Les valeurs des couleur sont des virgules flottantes entre 0.0 et 1.0.

## Claviers

Le modèle de clavier X est dispatché en 2 couches : les codes spécifiques au serveur (keycodes) qui représentent les touches physiques, et les symboles indépendant du serveur (keysyms) qui représentent les lettres qui apparaissent sur les touches. 2 tables sont conservés sur le serveur pour convertir les keycodes en keysyms :

**modifier list** : certaines clés ( Shift, Control, Caps lock) sont des modifieur et sont utilisés pour sélectionner différents symboles qui sont attachés à une touche. Quand une touche est enfoncée ou relâchée, le serveur génère un event qui contient le keycode de la clé indiquée et un masque qui spécifie le modifier key.

**keymap table** : les applications traduisent les events keycode et modifier masks en keysyms en utilisant une table de keysyms qui contient une colonne pour chaque keycode est une colonne pour divers modifieurs. la sémantique exacte de l'interprétation de la table pour produire les keysyms dépends du programme, librairie et langage utilisé.

## OPTIONS

Les programmes X acceptent automatiquement les options suivantes :

- display display** Nom du serveur X à utiliser
- geometry geometry** Taille et emplacement initiale de la fenêtre
- bg color, -background color** Spécifie la couleur à utiliser pour la fond de la fenêtre
- bd color, -bordercolor color** Spécifie la couleur à utiliser pour la bordure de la fenêtre
- bw number, -borderwidth number** Largeur en pixel de la bordure de la fenêtre
- fg color, -foreground color** Couleur à utiliser pour les textes et graphiques
- fn font, -font font** Font à utiliser pour les textes à afficher
- iconic** Lance la fenêtre réduite
- name** Spécifie le nom sous lequel les ressources de l'application devraient être trouvés.
- rv, -reverse** Simule le mode vidéo reverse si possible
- +rv** Indique que le programme ne devrait pas simuler le mode vidéo reverse.
- selectionTimeout** Timeout en milliseconde pour 2 applications communicante doivent se répondre pour une requête de sélection
- synchronous** Indique que les requêtes au server X devraient être envoyé synchrone. Utile pour le débogage
- title string** Titre utilisé pour la fenêtre

- 
- xnllanguage language** [**\_territory**] [**.codeset**] Spécifie la langue, territoire et codeset pour aider à résoudre les ressources et autres noms de fichier
  - xrm resourcestring** Nom et valeur de ressource pour remplacer les valeurs par défaut. Utile pour les paramètres qui n'ont pas d'options de ligne de commande.

## Ressources

Pour simplifier l'adaptation des applications aux préférences utilisateurs, X fournit un mécanisme pour stocker les valeurs par défaut pour les programmes, qui sont utilisés par les programmes qui utilisent X toolkit (les programmes utilisant GTK et QT toolkit utilisent des mécanismes de configuration autre). Les ressources sont spécifiés en tant que chaîne. Les composants de programme sont nommés de manière hiérarchique où chaque noeud est identifié par une classe et un nom d'instance. En haut se trouve la classe et le nom de l'application elle-même. Par convention, la nom nom de la classe et le nom de l'application, mais avec la première lettre en majuscule. La syntaxe pour les ressources est :

**ResourceLine = Comment | IncludeFile | ResourceSpec | <empty line>**

Comment = " !" {<any character except null or newline>}

IncludeFile = "#" WhiteSpace "include" WhiteSpace FileName WhiteSpace

FileName = <valid filename for operating system>

ResourceSpec = WhiteSpace ResourceName WhiteSpace ":" WhiteSpace Value

ResourceName = [Binding] {Component Binding} ComponentName

Binding = "." | "\*"

WhiteSpace = {<space> | <horizontal tab>}

Component = "?" | ComponentName

ComponentName = NameChar {NameChar}

NameChar = "a"-"z" | "A"-"Z" | "0"-"9" | "\_" | "-"

Value = {<any character except null or unescaped newline>}

Les programmes basés sur X toolkit obtiennent les ressources depuis les sources suivantes :

**RESOURCE\_MANAGER root window property** Toutes les ressources globales disponibles pour les clients sur toutes les machines sont stockées dans la propriété RESOURCE\_MANAGER de la fenêtre root du premier écran en utilisant xrdp.

**SCREEN\_RESOURCES root window property** Toutes les ressources spécifiques à un écran donné disponible au client sur toutes les machines sont stockées dans la propriété SCREEN\_RESOURCES de la fenêtre root de cet écran

**application-specific files** Les répertoires nommés par la variable d'environnement XUSERFILESEARCHPATH ou XAPPLRESDIR (qui nomment un seul répertoire et doit se terminer par /), et les répertoires standard (généralement dans /usr/share/X11, mais peut être changé avec XFILESEARCHPATH) sont parsés pour des ressources spécifique à une application

**XENVIRONMENT** Toutes ressources spécifique à une application ou un utilisateur peuvent être spécifiés dans XENVIRONMENT à un nom de fichier de ressource à charger par toutes les applications. Si elle n'est pas définie, un fichier nommé \$HOME/.Xdefaults-hostname est recherché.

-**xrm resourcestring** Les ressources peuvent être spécifiées sur la ligne de commande. peut être spécifié plusieurs fois.

---

Les ressources des programmes sont organisés en groupes appelés **classes**, les collections de ressources individuels (chacune appelée **instance**) peuvent être spécifiés en une fois. Par convention, le nom de l'instance commence avec une lettre minuscule et le nom de la classe avec une lettre majuscule. Les applications auront au moins les ressources suivantes :

**background** (class **Background**) Cette ressource spécifie la couleur à utiliser pour le fond de la fenêtre.

**borderWidth** (class **BorderWidth**) Cette ressource spécifie la largeur en pixel de la bordure de la fenêtre.

**borderColor** (class **BorderColor**) Cette ressource spécifie la couleur à utiliser pour la bordure de la fenêtre.

En combinant les classes et instances, les préférences peuvent être définies rapidement et facilement. Par exemple :

```
bitmap*Dashed : off
XTerm*cursorColor : gold
XTerm*multiScroll : on
XTerm*jumpScroll : on
XTerm*reverseWrap : on
XTerm*curses : on
XTerm*Font : 6x10
XTerm*scrollBar : on
XTerm*scrollbar*thickness : 5
XTerm*multiClickTime : 500
XTerm*charClass : 33 :48,37 :48,45-47 :48,64 :48
XTerm*cutNewline : off
XTerm*cutToBeginningOfLine : off
XTerm*titeInhibit : on
XTerm*ttyModes : intr ^c erase ^? kill ^u
XLoad*Background : gold
XLoad*Foreground : red
XLoad*highlight : black
XLoad*borderWidth : 0
emacs*Geometry : 80x65-0-0
emacs*Background : rgb :5b/76/86
emacs*Foreground : white
emacs*Cursor : white
emacs*BorderColor : white
emacs*Font : 6x10
xmag*geometry : -0-0
xmag*borderColor : white
```

Si ces ressources étaient stockées dans un fichier nommé `.Xresources` dans le `home`, elles peuvent être ajoutées à des ressources existantes dans le serveur avec la ligne de commande :

```
% xrdp -merge $HOME/.Xresources
```

## Variables d'environnement

**DISPLAY** Pointe vers un serveur X.

**XAUTHORITY** Pointe vers un fichier qui contient des données d'autorisation (défaut : `$HOME/.Xauthority`)

**ICEAUTHORITY** Pointe vers un fichier qui contient des données d'autorisation (défaut : `$HOME/.ICEauthority`)

**LC\_ALL, LC\_CTYPE, LANG** La première valeur non vide détermine les informations de localité actuelles.

**XMODIFIERS** Contient des informations importantes pour la locale courante. (généralement : `@im=<input-method>`)

**XLOCALEDIR** Pointe vers un répertoire contenant le fichier `locale.alias` et les hiérarchies de fichiers `Compose` et `XLC_LOCALE` pour toutes les locales. (défaut : `/usr/share/X11/locale`)

**XENVIRONMENT** Pointe vers un fichier contenant les ressources X locales (défaut : `$HOME/.Xdefaults-<hostname>`)

**XFILESEARCHPATH** doit contenir une liste de templates de paths, séparés par une `,` ; défaut :

```
/usr/etc/X11/%L/%T/%N% C%S :\
```

```
/usr/etc/X11/%L/%T/%N% C%S :\
```

/usr/etc/X11/%T/%N%C%S :\n
 /usr/etc/X11/%L/%T/%N%S :\n
 /usr/etc/X11/%L/%T/%N%S :\n
 /usr/etc/X11/%T/%N%S :\n
 /usr/share/X11/%L/%T/%N%C%S :\n
 /usr/share/X11/%L/%T/%N%C%S :\n
 /usr/share/X11/%T/%N%C%S :\n
 /usr/share/X11/%L/%T/%N%S :\n
 /usr/share/X11/%L/%T/%N%S :\n
 /usr/share/X11/%L/%T/%N%S :\n
 /usr/share/X11/%T/%N%S :\n
 /usr/lib/X11/%L/%T/%N%C%S :\n
 /usr/lib/X11/%L/%T/%N%C%S :\n
 /usr/lib/X11/%T/%N%C%S :\n
 /usr/lib/X11/%L/%T/%N%S :\n
 /usr/lib/X11/%L/%T/%N%S :\n
 /usr/lib/X11/%T/%N%S

**un template est transformé en substituant :**

**%D** Chemin par défaut spécifique à l'implémentation  
**%N** nom (basename) de recherche  
**%T** type (dirname) de recherche  
**%S** suffix de recherche  
**%C** Valeur de la ressource "customization"  
**%L** Nom de locale  
**%I** Langue de la locale  
**%t** Territoire de la locales  
**%c** Encodage de la locale

**XUSERFILESEARCHPATH** Liste de templates de path, où libXt recherche les fichiers de ressources dépendant de l'utilisateur. défaut :

\$XAPPLRESDIR/%L/%N%C :\n
 \$XAPPLRESDIR/%L/%N%C :\n
 \$XAPPLRESDIR/%N%C :\n
 \$HOME/%N%C :\n
 \$XAPPLRESDIR/%L/%N :\n
 \$XAPPLRESDIR/%L/%N :\n
 \$XAPPLRESDIR/%N :\n
 \$HOME/%N

**XAPPLRESDIR** Pointe vers un répertoire où l'utilisateur stocke ses fichiers de ressource dépendant des applications (défaut : \$HOME") uniquement si XUSERFILESEARCHPATH n'est pas définis

**XKEYSYMDB** Pointe vers un fichier contenant des définitions de keysym non-standard (défaut : /usr/share/X11/XKeysymDB)

**XCMSDB** Pointe vers fichier de base de données de noms de couleur. (défaut : /usr/lib/X11/Xcms.txt)

**RESOURCE\_NAME** Sert d'identifiant principal pour les ressources appartenant à exécuter. (défaut : le basename du pathname du programme)

**SESSION\_MANAGER** Spécifie le gestionnaire de session

**XF86BIGFONT\_DISABLE** si non vide, désactive l'extension XFree86-Bigfont.

**XKB\_FORCE** Influence l'extension X Keyboard

**XKB\_DISABLE** Influence l'extension X Keyboard

---

**XKB\_DEBUG** Influence l'extension X Keyboard  
**\_XKB\_CHARSET** Influence l'extension X Keyboard  
**\_XKB\_LOCALE\_CHARSETS** Influence l'extension X Keyboard  
**\_XKB\_OPTIONS\_ENABLE** Influence l'extension X Keyboard  
**\_XKB\_LATIN1\_LOOKUP** Influence l'extension X Keyboard  
**\_XKB\_CONSUME\_LOOKUP\_MODS** Influence l'extension X Keyboard  
**\_XKB\_CONSUME\_SHIFT\_AND\_LOCK** Influence l'extension X Keyboard  
**\_XKB\_IGNORE\_NEW\_KEYBOARDS** Influence l'extension X Keyboard  
**\_XKB\_CONTROL\_FALLBACK** Influence l'extension X Keyboard  
**\_XKB\_COMP\_LED\_XKB\_COMP\_FAIL\_BEEP** Influence l'extension X Keyboard

## Exemples

Lignes de commande fréquemment utilisés :

```
% xrdp $HOME/.Xresources
% xmodmap -e "keysym BackSpace = Delete"
% mkfontdir /usr/local/lib/X11/otherfonts
% xset fp+ /usr/local/lib/X11/otherfonts
% xmodmap $HOME/.keymap.km
% xsetroot -solid 'rgb :.8/.8/.8'
% xset b 100 400 c 50 s 1800 r on
% xset q
% twm
% xmag
% xclock -geometry 48x48-0+0 -bg blue -fg white
% xeyes -geometry 48x48-48+0
% xbiff -update 20
% xlsfonts '*helvetica*'
% xwininfo -root
% xdpyinfo -display joesworkstation :0
% xhost -joesworkstation
% xrefresh
% xwd | xwud
% bitmap companylogo.bm 32x32
% xcalc -bg blue -fg magenta
% xterm -geometry 80x66-0-0 -name myxterm $*
```