

X-I

Utilisation du systeme X

- o **Le systeme X**
- o **Clients**
- o **Lancement de clients**
- o **Le gestionnaire de fenetres**

X-I

Le système X

- **Le système X est un système de création et de gestion d’interfaces graphiques.**
- **Le système X permet l’affichage, sur un ou plusieurs écrans,**
 - | **de plusieurs applications,**
 - | **tournant sur des machines différentes,**
 - | **dans des fenêtres séparées.**
- **Il coordonne les activités des différentes applications**
 - | **qui n’ont pas besoin de se connaître,**
 - | **et qui peuvent pourtant échanger des données.**
- **Les programmes construits sur les principes du système X sont appelés *clients X*.**
- **Le dispositif graphique de sortie (écran), et le dispositif de saisie (souris et clavier) constituent le “display”. Le processus chargé de gérer le display est appelé un *serveur*.**

X-I

Le système X

- **Chaque application X**
 - | se connecte elle-même à un serveur via un *réseau* ;
 - | et dispose sur son display d'une ou de plusieurs fenêtres.
- **L'usage d'un *gestionnaire de fenêtres* est recommandé, mais n'est pas indispensable.**
 - | il coordonne la position relative des fenêtres, leur taille et leur décoration,
 - | mais le contenu (l'intérieur) des fenêtres est de la responsabilité des applications.
- **Le système X constitue le fondement d'un système de fenêtrage sur lequel sont construit des interfaces graphiques utilisateurs. La plus répandue des interfaces graphiques est OSF/Motif, développée par Open Software Foundation.**
- **Motif fournit également un gestionnaire de fenêtres (*mwm*) qu'il convient de distinguer des outils d'interfaces ("widget sets").**

X-I

Clients X

- o **Le système X a été développé au MIT, dans le cadre du projet Athena. Il est actuellement maintenu et développé par le Consortium X qui regroupe les principaux constructeurs.**
- o **Le système est *fourni* avec un certain nombre de clients; de plus, de nombreuses *contributions* émanant d'autres origines circulent.**
- o **Quelques clients importants :**
 - | `xterm` **émulateur de terminal alphanumérique, avec un shell**
 - | `xedit` **un éditeur de texte simple**
 - | `bitmap` **un éditeur de bitmap**
 - | `xfig` **un logiciel de dessin**
- o **Quelques accessoires de bureau :**
 - | `xcalc` **une calculatrice**
 - | `xclock` **une horloge**
 - | `xmh` **un gestionnaire de courrier**
 - | `xload` **donne la charge du système**
- o **Configuration**
 - | `xset` **configure le display (pointeur, sonnerie)**
 - | `xsetroot` **configure le fond d'écran (dessin)**

**X-I**

Clients d'information

- o `xman` **pages de manuels**
- o **listes**
 - | `xlsclients` **liste les clients connectés**
 - | `xlsfonts` **liste les polices disponibles**
 - | `xlswins` **liste l'arbre des fenêtres (obsolète)**
 - | **listres** **liste les noms des ressources d'une widget**
- o **informations**
 - | `xdpyinfo` **liste les informations sur le display**
 - | `xwininfo` **informations sur les fenêtres**
 - | `xfd` **affiche les caractères d'une fonte**
- o **sélection**
 - | `editres` **arbre des widgets et édition des ressources**
 - | `xfontsel` **choix d'une fonte**
- o **configuration**
 - | `xrdb` **gestion de la base de ressources**
 - | `xdm` **le programme qui gère les terminaux X d'une machine.**

**X-I**

Lancement de clients X

- **X permet à l'utilisateur de spécifier des options sur la ligne de commande qui lance un client.**
 - | **certaines options sont communes à toutes les applications X.**
 - | **certaines applications admettent de plus des options spécifiques.**
- **Les options communes concernent**
 - | **le serveur utilisé;**
 - | **le choix de couleurs pour le contenu et la bordure de la fenêtre (principale);**
 - | **la fonte;**
 - | **la “géométrie”, c'est-à-dire la taille et le placement;**
 - | **le titre de la fenêtre;**
 - | **la spécification d'une ressource.**
- **Ces options spécifient (sauf en ce qui concerne le serveur) des valeurs de *ressources* :**
 - | **les ressources concernent des *widgets* (contraction de *window* et *gadget*, désigne une fenêtre avec un aspect et un comportement prédéfinis).**
 - | **une ressource est un *attribut* d'une widget qui peut recevoir une valeur.**
 - | **une ressource a une valeur par défaut; spécifier une valeur d'une ressource consiste donc à *personnaliser* cette ressource.**

X-I

Les options communes

- o Les options communes sont données en spécifiant
 - | leur *nom*, qui existe en version courte et version longue,
 - | suivi de la *valeur*.
- o Les principales options sont

<i>Option</i>	<i>Description</i>
-display, -d	numéro du serveur auquel se connecter
-background, -bg	couleur du fond de la fenêtre
-foreground, -fg	couleur des dessins et du texte
-borderwidth, -bw	largeur du bord (en pixels)
-bordercolor, -bd	couleur du bord
-geometry, -g	taille et position de la fenêtre
-font, -fn	police de caractères
-iconic	lancement sous forme icônifiée
-name	nom de cette instance de l'application
-title	titre de la fenêtre
-reverse, -rv	échange les couleurs foreground et background
+rv	refuse cette inversion
-xrm	permet de spécifier une ressource particulière

X-I

display

- o L'option `-display` permet de donner le nom de la machine, du serveur et de l'écran sur lequel s'afficheront les fenêtres du client.
- o Par défaut, cette valeur est celle contenue dans la variable d'environnement `DISPLAY`
- o L'option a la syntaxe
 - `-display [hôte]:serveur[.écran]`
 - | `hôte` est l'adresse Internet de la machine sur laquelle tourne le serveur. Ce peut être un nom, ou un numéro;
 - | `serveur` est le numéro du processus serveur sur la machine hôte; en effet, plusieurs serveurs peuvent tourner simultanément sur une même machine; en général, le numéro est 0;
 - | `écran` est le numéro de l'écran principal (par défaut); un serveur peut avoir plusieurs écrans, numérotés à partir de zéro; en général, le numéro est 0;
- o Si la connexion échoue, c'est peut-être parce que la machine du client n'est pas autorisée à se connecter sur la machine du serveur. On peut alors y remédier par la commande `xhost`.
- o Exemple : `rsh ibm5 xterm -display zeus:0.0 &`

X-I

couleurs

- o **Les options** `-foreground`, `-background`, `-bordercolor` **permettent de choisir d'autres couleurs.**
- o **Elles sont suivies d'une spécification de couleur.**
- o **Le moyen le plus simple est de fournir un nom pris dans la *base des noms de couleurs standard*. Cette base est située dans le fichier ASCII `rgb.txt`, normalement dans le répertoire `/usr/lib/X11`. Elle contient plusieurs centaines de noms. Un nom formé de plusieurs mots doit être spécifié entre une paire de guillemets.**
- o **Exemples :**

<code>ghost white</code>	<code>peach puff</code>	<code>lavender blush</code>
<code>slate gray</code>	<code>midnight blue</code>	<code>turquoise</code>
<code>khaki</code>	<code>chartreuse</code>	<code>olive drab</code>
<code>sienna</code>	<code>light yellow</code>	<code>violet red</code>
- o **Les couleurs standard** (`red`, `blue`, `green`, ...) **existent aussi, bien entendu.**

X-I

taille et position

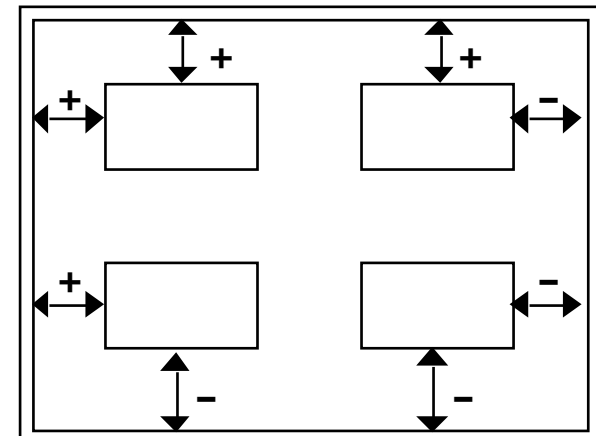
- L'option `-geometry` permet de donner l'emplacement, et la taille, de la fenêtre du client.
 - | *l'emplacement* se définit par rapport aux bords de l'écran,
 - | la taille est donnée par la *largeur* et la *hauteur* de la fenêtre.
- La valeur de cette option donnée par 4 paramètres, sous la forme
 - | *largeur* x *hauteur* ± *decalage_x* ± *decalage_y*
- Il n'y a *pas d'espace* entre les paramètres : ils ont été ajoutés ici pour plus de clarté.
- Exemples :

Coin supérieur gauche : `-geometry +0+0`

Coin supérieur droit : `-geometry +0-0`

Coin inférieur droit, avec une marge verticale de 10, et une marge horizontale de 50 :

`-geometry -50-10`



**X-I**

- o La largeur et la hauteur concernent *l'intérieur* de la fenêtre.
- o L'unité est en général le pixel, mais pour certaines applications, une unité plus significative :
 - l pour `xterm`, ce sont le nombre de caractères par ligne, et le nombre de lignes.
- o Chaque client a des valeurs par défaut, et il suffit de donner les paramètres qui changent :

```
xterm -geometry x40+-0
```

définit un `xterm` avec 40 lignes, et un nombre standard (80) de colonnes, touchant le bord droit de l'écran

X-I

nom, titre, et fonte

- o L'option `-name` permet de donner un nom particulier à l'instance du programme à lancer; elle sert à particulariser l'usage de ressources.
- o L'option `-title` permet de donner le titre qui s'affichera dans la barre de titre de la fenêtre; par défaut, c'est le nom de l'application. Ceci est utile pour distinguer plusieurs instances de la même application.
- o
- o L'option `-font` permet d'utiliser une police autre que la fonte par défaut. Le système X fournit une grande variété de fontes (plusieurs centaines, mais cela dépend de l'installation).
 - | le nommage des fontes est complexe, car chaque nom contient une information complète. Souvent, on choisira une fonte dont le "jeu de caractères" est conforme à `iso8859-1`, ce qui implique qu'elle contient les caractères accentués.
 - | pour choisir une fonte, utiliser `xfontsel`, et pour l'examiner de près, `xfd`. Ne pas oublier d'entourer le nom de la fonte choisi d'apostrophes, s'il contient des caractères spéciaux.

exemples :

```
-font '*courier-medium-r-*-120*-iso8859-1'  
-font 'b&h-lucida-*-r--18-*-iso8859-1'
```

X-I

Ressources

- La plupart des options correspondent à des attributs de l'application, appelés *ressources*, mais beaucoup d'applications X ont des ressources spécifiques supplémentaires.
- L'usage de ressources est un des outils majeurs de la programmation X.
- Si l'on veut changer de manière permanente la valeur de certaines ressources pour certaines applications, on peut utiliser un *fichier de ressources* .
- Ce fichier est à placer dans le répertoire de base de l'utilisateur, et a pour nom `.Xresources` ou `.Xdefaults`.
- Chaque client X (installé par l'administrateur système) dispose en plus d'un fichier par défaut, qui se trouve dans `/usr/lib/X11/app-defaults`.
- Au lancement, un client, (ou plus précisément le gestionnaire de ressources appelé automatiquement), consulte
 - | le fichier par défaut, puis
 - | le fichier `.Xresources` ou `.Xdefaults`, puis
 - | la ligne de commandepour déterminer la valeur des ressources. La dernière valeur trouvée est retenue.

X-I

Fichier de ressources

- Un fichier de ressources est un fichier ASCII, où chaque ligne spécifie une ressource et sa valeur. La forme la plus simple est

```
|          client*variable :  valeur
```

- mais en respectant la hiérarchie des widgets de l'application, des spécifications bien plus précises peuvent être données. Ici, *client* est

- | soit le *nom* du client (éventuellement modifié par `-name`)

- | soit le nom de la *classe* du client : en effet, chaque client est instance d'une classe.

- **Exemple** : le client `xclock` affiche une montre. En spécifiant certaines ressources, on peut personnaliser le comportement :

- `analog` est par défaut `true`, mais en demandant `false`, on obtient une montre digitale.

- `update` spécifie l'intervalle de mise-à-jour de l'heure (en secondes).

- `chime` spécifie si une sonnerie doit retentir toutes les dem-heures. Par défaut, cette ressource vaut `false`.

- On obtient *systématiquement* (sauf si spécifié autrement sur la ligne de commande) une montre digitale, à sonnerie, et à mise-à-jour toutes les 3 secondes, en mettant dans le fichier de ressources:

```
xclock*analog : false
```

```
xclock*update : 3
```

```
xclock*chime : true
```

X-I

Le gestionnaire de fenêtres

- Le gestionnaire de fenêtres (“window manager”) est un client X comme les autres, ou presque.
- On peut fort bien
 - | ne pas utiliser de gestionnaire de fenêtres, ou
 - | en changer en cours de session (si l’initialisation de la session est faite de manière appropriée),
 - | changer de machine exécutant le gestionnaire de fenêtres.
- Le rôle du gestionnaire est de faire vivre en commun les applications présentes, en permettant
 - | de gérer la *géométrie*, i.e. de déplacer, retailer, empiler différemment les fenêtres des applications;
 - | d’*icônifier* les applications;
 - | de contrôler le *foyer d’entrée* du clavier, c’est-à-dire de déterminer qu’elle est la fenêtre vers laquelle sont dirigées les frappes du clavier.
- Pour ce faire, la plupart des gestionnaires ajoutent, aux fenêtres (principales) des applications, une *décoration*, composée d’une bordure, d’une barre de titre avec divers boutons, voire d’un menu déroulant.

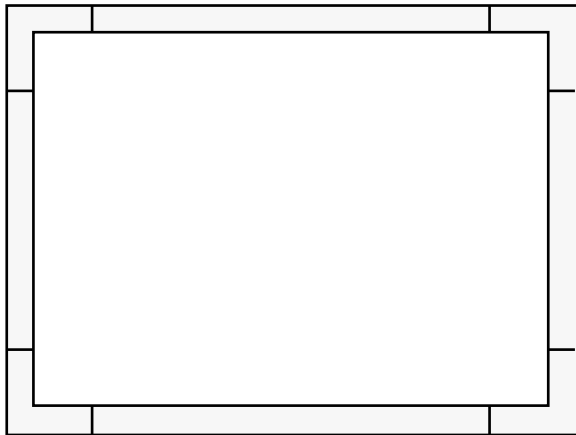
X-I

Décoration

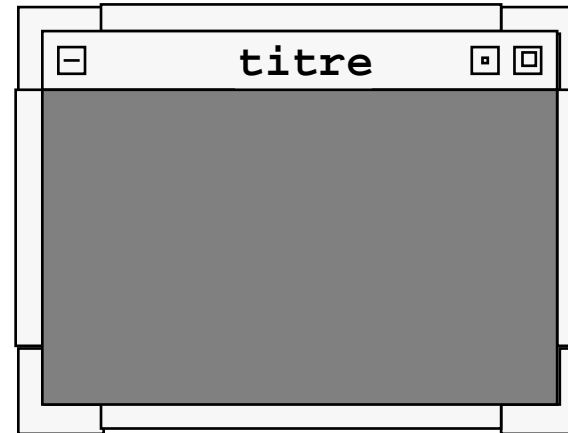
- Fenêtre du client, non décorée



- Bordure de décoration



- Fenêtre avec décoration



- Barre de titre



- Barre de titre, sans bouton



X-I

Les composants d'un gestionnaire de fenêtres

- Un gestionnaire de fenêtres fournit aux clients
 - | une certaine *apparence*, par la décoration qu'il leur donne,
 - | et des *possibilités* supplémentaires, par un éventuel *menu de fenêtre* et par l'*icônification*.
- De plus, un gestionnaire dispose en général d'un ou plusieurs *menu de fond* ("root menus") permettant de lancer de nouveaux clients ou d'en tuer, ou de relancer le gestionnaire.
- Enfin, le gestionnaire définit
 - | des *raccourcis clavier*,
 - | et un *comportement de transition*, qui décrit son comportement lorsque le curseur de la souris passe d'une fenêtre à une autre. Ceci concerne par exemple le comportement du foyer du clavier :
 - » "click-to-type"
 - » "follow-cursor".
- La plupart de ces comportements sont régis par des ressources, ou des fichiers particuliers.

X-I

Le gestionnaire *mwm* de Motif

- o **La personnalisation de *mwm* se fait au moyen de deux fichiers**
 - | **un fichier particulier, de nom `.mwmwrc`, dans le répertoire de base**
 - | **un fichier de ressources qui est soit un fichier de nom `Mwm`, soit inclus dans le fichier `.Xresources` ou `.Xdefaults`.**
- o **Il existe des fichiers par défaut, de nom `system.mwmwrc`, pour le premier fichier, et `Mwm` pour le deuxième.**
- o **Le fichier `system.mwmwrc` définit**
 - | **les *menus*, et les fonctions liées aux entrées des menus;**
 - | **les *raccourcis clavier* et leurs effets selon les contextes;**
 - | **les effets des *boutons* de la souris.**

Un fichier personnalisé sert donc à modifier ces aspects.

- o **Le fichier `Mwm` fixe les valeurs de certaines ressources, comme les couleurs, épaisseurs, dispositions de fenêtres, de titres, d'icônes. Il existe des douzaines d'options.**

X-I

Les menus

- o Deux menus sont toujours définis:
 - | le “Root Menu”, appelé lorsque l’on clique dans la fenêtre racine;
 - | le “Window Menu” attaché à la case de la barre de titre d’une fenêtre.

```

Menu DefaultRootMenu
{
    "Root Menu"          f.title
    "New Window"        f.exec "xterm &"
    "Shuffle Up"        f.circle_up
    "Shuffle Down"      f.circle_down
    "Refresh"           f.refresh
    "Pack Icons"        f.pack_icons
    "Utilities"         f.menu UtilitiesMenu
    no-label            f.separator
    "Restart..."      f.restart
}

```

- o Les *menus* sont spécifiés par:
 - | le mot-clé Menu, suivi d’un nom
 - | les entrées entre accolades. Chaque entrée a la syntaxe:
intitulé équivalent-clavier raccourci fonction
 - | l’équivalent-clavier et le raccourci sont facultatifs.

Des fonctions prédéfinies existent en grand nombre. Elles débutent par “f .”

X-I

Exemple de menu de fenêtre

```
Menu DefaultWindowMenu
```

```
{  
    "Restore"      _R      Alt<Key>F5      f.normalize  
    "Move"        _M      Alt<Key>F7      f.move  
    "Size"        _S      Alt<Key>F8      f.resize  
    "Minimize"    _n      Alt<Key>F9      f.minimize  
    "Maximize"    _x      Alt<Key>F10     f.maximize  
    "Lower"       _L      Alt<Key>F3      f.lower  
    no-label  
    "Close"       _C      Alt<Key>F4      f.kill  
}
```

- **La deuxième colonne donne l'équivalent clavier , c'est-à-dire le caractère qui permet d'activer l'entrée du menu, une fois le menu activé;**
- **La troisième colonne donne le raccourci clavier, c'est-à-dire la combinaison de touches qui permet d'activer l'entrée, même si le menu n'est pas activé.**

X-I

Les raccourcis clavier

```
Keys DefaultKeyBindings
```

```
{
    Shift<Key>Escape          window|icon          f.post_wmenu
    Alt<Key>Tab               root|icon|window    f.next_key
    Alt Shift<Key>Tab        root|icon|window    f.prev_key
    Alt<Key>Escape           root|icon|window    f.circle_down
    Alt Shift<Key>Escape     root|icon|window    f.circle_up
    Alt Shift Ctrl<Key>exclam root|icon|window    f.set_behavior
    Alt<Key>F6                window              f.next_key transient
    Alt Shift<Key>F6         window              f.prev_key transient
    Shift<Key>F10            icon                f.post_wmenu
}
```

- o **La syntaxe est semblable : mot-clé `Keys`, puis un nom.**
- o **La colonne centrale indique le contexte dans lequel cette combinaison s'applique.**
Les contextes sont :

```
root, window, icon, title, border, frame, app
```

```
| frame contient le bord et la barre de titre : toute la décoration
```

```
| app concerne uniquement la fenêtre propre au client (sans décoration)
```

```
| window est la réunion des deux précédents.
```

X-I

Liaisons des boutons

```
Buttons DefaultButtonBindings
{
    <Btn1Down>          icon|frame      f.raise
    <Btn3Down>          icon|frame      f.post_wmenu
    <Btn3Down>          root            f.menu  DefaultRootMenu
}
```

- o **Les liaisons des boutons effectuent certaines actions, selon le contexte. La syntaxe est similaire.**
- o **Comme on peut le voir, le *menu racine* (ici de nom `DefaultRootMenu`) est affiché si l'on clique avec le troisième bouton dans la fenêtre par défaut, alors que c'est le *menu de fenêtre* qui est affiché si c'est dans le cadre d'une fenêtre ou sur une icône.**
- o **Le fichier `system.mwmrc` contient trois ensembles de liaisons de boutons :**
 - | `DefaultButtonBindings`
 - | `ExplicitButtonBindings`
 - | `PointerButtonBindings`
- o **Le choix de l'ensemble retenu se fait dans le fichier de ressources pour `mwm`, en spécifiant la ressource `buttonBindings`:**

```
Mwm*buttonBindings : DefaultButtonBindings
```

X-I

Les fonctions

<code>f.beep</code>	
<code>f.circle_down, f.circle_up,</code>	circulation de fenêtres
<code>f.raise, f.lower,</code>	met en avant ou en arrière la fenêtre
<code>f.refresh,</code>	redessine toutes les fenêtres
<code>f.exec "Command"</code>	
<code>f.kill,</code>	termine un client X
<code>f.maximize, f.minimize, f.move, f.resize,</code>	
<code>f.normalize, f.restore,</code>	remet la fenêtre à la taille normale ou précédente
<code>f.menu MenuName</code>	
<code>f.next_key, f.prev_key,</code>	avance le foyer du clavier à la fenêtre ou à l'icône suivant (transient, même les fenêtres secondaires)
<code>f.nop, f.separator, f.title</code>	ne fait rien, crée une ligne
<code>f.post_wmenu,</code>	affiche le menu de fenêtre
<code>f.quit_mwm, f.restart,</code>	
<code>f.pack_icons,</code>	réarrange les icônes de façon optimale, selon la disposition choisie.

X-I

Le fichier de ressources Mwm

- Les très nombreuses ressources peuvent être groupées en 3 catégories
 - | les ressources qui déterminent *l'apparence* (le “look”), telles que les cadres des fenêtres, les couleurs des menus, la taille des icônes.
 - | le *comportement* (le “feel”), tels que la liaison des boutons, le foyer du clavier, etc.
 - | les ressources spécifiques à des clients, ou des clients particuliers, en ce qui concerne leur interaction avec le gestionnaire.
- L'apparence concerne quatre composants, chacun identifié par un nom:
 - | menu : pour les menus
 - | icon : pour les icônes
 - | client : pour la décoration des clients X
 - | feedback : pour les dialogues entre mwm et l'utilisateur.

X-I

Exemples

- o **Pour spécifier la couleur de fond**
- o `Mwm*client*background` : aquamarine
 - | **donne cette couleur à la totalité de la *décoration* des fenêtres des clients X**
- o `Mwm*client*title*background` : lightblue
 - | **donne cette couleur à la barre de titre.**
- o `Mwm*background` : lightgrey
 - | **donne cette couleur à tous les composants (décoration, menus, icônes, dialogues).**
- o `Mwm*menu*Utilities*background` : seagreen
 - | **donne cette couleur au menu de nom indiqué.**

Lorsqu'une fenêtre est active, c'est la couleur définie dans la ressource `activeBackground` qui est utilisée !

X-I

Quelques ressources utiles à connaître

- o `Mwm*keyboardFocusPolicy` : `explicit`
 - | **on doit cliquer dans une fenêtre pour que les caractères frappés soient dirigés vers elle (“click-to-type”).**
- o `Mwm*keyboardFocusPolicy` : `pointer`
 - | **les caractères frappés sont dirigés vers la fenêtre où se trouve le pointeur de la souris (“follow-pointer”).**

C’est fréquemment le premier qui est le mode par défaut.

- o `Mwm*focusAutoRaise` : `True` (ou `False`)
 - | **si vrai, la fenêtre où se trouve le pointeur de la souris est mise automatiquement en avant-plan.**
 - | **Vrai par défaut, si `keyboardFocusPolicy` est `explicit`.**
 - | **Faux par défaut dans le cas contraire.**

**X-I**

- o **clientDecoration** : spécifie la décoration de tous les clients; les options sont une combinaison de:
 - | maximize, minimize, menu, title (barre de titre),
 - | border, resizeh (poignées),
 - | all, none.
- o **Certaines options impliquent les autres :**
 - | on ne peut pas avoir de bouton ou de menu sans barre de titre
 - | on ne peut avoir de poignée ou de barre de titre sans bordure.
- o **Par défaut, l'option est all.**
 - | si la première option est précédée d'un signe - , les options qui suivent sont désactivées.
 - | si la première option est précédée d'un signe + , les options qui suivent sont seules activées.
- o **Exemple :**
 - | Mwm*clientDecoration : - minimize maximize menu
 - | Mwm*clientDecoration : title resize border
- o **transientDecoration** : spécifie la décoration des fenêtres éphémères. La même syntaxe s'applique. Par défaut, la ressource vaut : menu title

**X-I**

Le gestionnaire *fvwm*

Gestionnaire disponible notamment sous linux

- **dérivé de twm**
- **bureau virtuel, barre de taches.**
- **modules : programmes chargés sur demande**
- **en évolution**

X-I

Bureau virtuel

- L'écran est vu comme un cadre (viewport) posé sur un bureau (qui peut donc être plus grand que l'écran)
- Plusieurs bureaux peuvent exister ("un bureau par projet")
- Le placement d'un client est relatif au cadre (écran) courant:

```
xterm -geometry +0+0
```

place au coin supérieur gauche de la partie visible de l'écran.

- On peut spécifier le bureau par type d'application:

```
Style "*etscape" StartsOnDesk 2
```

Préprocesseur

Le fichier de configuration est en général passé par le préprocesseur m4.

- | soit à chaque lancement
- | soit après modification de la configuration.

Certains symboles sont prédéfinis, comme

HEIGHT	hauteur de l'écran
WIDTH	largeur de l'écran (en pixels)
COLOR	"Yes" si écran couleur, "No" si en niveau de gris

Exemples de macro:

```
define(geom,$1x$2)
define(MY_HEIGHT,eval(HEIGHT - 125))
define(MY_WIDTH,eval(WIDTH-10))
define(fullsize,geom(MY_WIDTH,MY_HEIGHT+0+0))
define(X_Sel,ifelse(WIDTH,1280,$1,ifelse(WIDTH,1152,$2,$3)))
define(Y_Sel,ifelse(HEIGHT,1024,$1,ifelse(HEIGHT,900,$2,$3)))
netscape.geometry: geom(X_Sel(910,910,790),Y_Sel(870,780,650))+0+0
```

X-I

Modules

Un module

- | est un programme séparé
- | s'exécute comme processus Unix
- | transmet des commandes à `f_vwm`
- | est lancé par `fvwm` qui établit les pipes.

Un module peut être lancé à l'initialisation ou sur demande; sa durée peut être temporaire ou pour la session.

Exemples de modules:

<code>FvwmAudio</code>	audio
<code>FvwmPager</code>	montre les bureaux
<code>FvwmButtons</code>	liaison entre boutons et fonctions
<code>FvwmIconBox</code>	boîte à icônes
<code>FvwmTaskBar</code>	barre des clients en cours

X-I

Fichier de configuration

L'exécution de `fvwm` s'accomplit avec un fichier de configuration, de nom `fvwmrc`, contenant les spécifications des ressources pour `fvwm` et les modules

La syntaxe est sous la forme *nom paramètre(s)*

Exemple:

```
DeskTopSize 1x1
ClickTime 750

# if the window occupies less than n% of the screen, draw it opaque
OpaqueMoveSize 5
EdgeScroll 100 100
EdgeResistance 10000 10000
```

X-I

Style

- Une façon souple de spécifier des paramètres
- Syntaxe

Style recepateur options

recepateur
options

**fenêtre, groupe, ressource, avec * et ?
liste de valeurs séparées par des virgules, parfois arguments
supplémentaires**

- Les spécifications sont additionnées, en cas de conflit, la dernière a raison.

X-I

Exemples

```
Style "*"          BorderWidth 6, HandleWidth 6
Style "*"          Icon u/unknown1.xpm, IconTitle
Style "*"          RandomPlacement, SmartPlacement
Style "*"          NakedTransient
Style "Fvwm*"      NoTitle, NoHandles, Sticky, WindowListSkip,

Style "*lock"     NoTitle, NoHandles, Sticky, WindowListSkip, CirculateSkip
Style "*lock"     BorderWidth 0
Style "xman"      Icon x/xman.xpm
Style "*osaic"    StartsOnDesk 3, Icon m/Mosaic.xbm
Style "*etscape"  StartsOnDesk 3
Style "*archie"   StartsOnDesk 3, NoPPosition,
Style "*fig"     StartsOnDesk 2
```

X-I

Fonctions

- Pour les définir et compléter:

```
# Definition de fonctions
# "I" immediate, "M" start when moving the mouse,
# "C" start after press and release, "D" double click

#####Initialization Functions #####
AddToFunc InitFunction      "I" Module FvwmButtons
#+                          "I" Module FvwmPager 0 3
#+                          "I" Module FvwmBacker

AddToFunc RestartFunction  "I" Module FvwmButtons
#+                          "I" Module FvwmPager 0 3
#+                          "I" Module FvwmBacker

#####Functions #####

AddToFunc Exec-or-Focus    "I" Next [iconic $0] Iconify
+                          "I" Next [$0] focus
+                          "I" None [$0] Exec $0 $1 $2 $3&

AddToFunc Move-or-Raise   "I" Raise
+                          "M" Move
+                          "D" Lower
```

X-I

Souris

```

# Definition des boutons de la souris
# Syntaxe:
# Mouse Numero Context Modifiers Function
#   Numero      = 0(tous),1,2,3
#   Context      = R(oot),W(indow),T(itle),S(ide),
#                 F(rame corner),I(con), 0-9(title_buttons), A(ll except T)
#   Modifiers    = N(o),C(ontrol),S(hift),M(eta),A(ll)

# Dans les boutons de la barre de titre
#   Button      Context Modif  Function
Mouse 01        A              Menu Window-Ops2 Close
Mouse 02        A              Maximize 100 100

# Dans la fenetre root
#   Button      Context Modif  Function
Mouse 1R        A              Menu Utilities Nop
Mouse 2R        A              Menu Window-Ops Nop
Mouse 3R        A              WindowList

# Misc
#   Button      Context Modif  Function
# Mouse 1       F           A    Resize-or-Raise
# Mouse 1       TS          A    Move-or-Raise
# Mouse 3       TSIF        A    RaiseLower
# Mouse 3       W           M    Mouse-Down-Action

```

X-I

Menus

```
AddToMenu Utilities
```

```
+      "Menu"                Title
+      " "                    Nop
+      "Shell"                Popup ShellMenu
+      "Programmes"           Popup AppMenu
+      "Gestionnaire de fentres"  Popup FvMenu
+      " "                    Nop
+      "Quitter la session"     Menu Quit
```

```
AddToMenu ShellMenu
```

```
+      "Shell"                Title
+      " "                    Nop
+      "&Shell Local"           Exec  xterm -sb -v -n $HOST -title $HOST -ls &
+      "Shells &Distants"     Popup Remote-Logins
```

```
AddToMenu AppMenu
```

```
+      "Programmes"           Title
+      ...
+      "Afficheur postscript"  Exec-or-Focus ghostview
```

```
AddToMenu Quit          "Etes-vous sur?" title
```

```
+      " "                    Nop
+      "&Non"                  Nop
+      "&Oui"                   Exec-or-Focus /usr/environnement/bin/tue.pl
```

X-I

Touches

```
# Syntaxe:
# Key Nom Context Modifiers Function
# Context = R(oot),W(indow),T(itle),S(ide),
#          F(rame corner),I(con), 0-9(title_buttons), A(ll execept T)
# Modifiers = N(o),C(ontrol),S(hift),M(eta),A(ll)
# Key Nom Context Modifiers Function
Key F1      A      S      Desk 0 0
Key F2      A      S      Desk 0 1
Key F3      A      S      Desk 0 2
Key F4      A      S      Desk 0 3
Key F1      A      M      Popup "Utilities"
Key F2      A      M      Popup "Window-Ops"
Key F3      A      M      Module FvwmWinList
Key Left    A      C      CursorMove -1 0
...
Key Left    A      M      CursorMove -10 +0
...
```

X-I

Fontes

```
# Definitions des fonts
```

```
WindowFont      7x14bold
```

```
IconFont        7x14bold
```

```
# Fontes du Pager
```

```
*FvwmPagerFont  -adobe-helvetica-bold-r-*-10-*-*-*-*-*iso8859-1
```

```
*FvwmPagerSmallFont 5x8
```

```
# Fonte pour l'identification
```

```
*FvwmIdentFont 7x14
```

```
#Fontes pour l'IconBox
```

```
*FvwmIconBoxFont      7x14
```

X-I

Barre de boutons, Pager

- La `ButtonBar` contient des boutons dédiés à des applications spécifiques.
- L'un des "boutons" est souvent un `Pager`, contenant une représentation des divers bureaux.

```
*FvwmButtonsFont -adobe-helvetica-bold-r-*-*10-*-*-*-*-*
*FvwmButtonsGeometry +0-0
*FvwmButtonsRows 1
*FvwmButtons(3x1,Swallow "FvwmPager" `FvwmPager 0 4`)
*FvwmButtons(Swallow "xclock" `Exec xclock -fg Black -bg AntiqueWhite -hd Red -hl Red
  -padding 0 -update 60 &`)
*FvwmButtons(Swallow "hbiff" `Exec hbiff -fg Black -bg AntiqueWhite -tally -name "e-mail"
  -H &`)
*FvwmButtons(Title Terminal,Icon xterm.xpm,Action `Exec xterm -sb -sl 4096 -v -n $HOST
  -title $HOST -ls &`)
*FvwmButtons(Title Editeur,Icon xedit.xpm, Action `Function Exec-or-Focus $EDITOR`)
*FvwmButtons(Title Aide,Icon xman.xpm,Action `Exec /usr/environnement/bin/mlv_help`)
*FvwmButtons(Title Netscape,Icon nscape.xpm,Action `Function Exec-or-Focus-Netscape`)
*FvwmButtons(Title Quitter,Icon e/exit5.xpm,Action `Menu Quit`)
```

X-I

Exemple

```
# Def pour Pager

*PagerGeometry 200x100-1-1
*FvwmPagerRows 1
*FvwmPagerColumns 4
*FvwmPagerLabel 0 Principal
*FvwmPagerLabel 1 Editeur
*FvwmPagerLabel 2 Gfx
*FvwmPagerLabel 3 Divers
```