DECwindows pour le System Manager

Overview

DECUS

7-SEP-1989

SOMMAIRE

INTRODUCTION

- Le but de cette presentation est de donner au "system manager" :

  -- Une meilleure vue que celle donnee par la documentation sur toutes les operations liees au "system management" specifiques a DECwindows

  -- Des idees plus claires sur l’ensemble des composantes de DECwindows sans rentrer pour autant dans les details de l’architecture

  -- Une approche permettant le "troubleshooting" des problemes que l’on peut rencontrer sous DECwindows

INTRODUCTION

- Cette presentation presuppose d’avoir une idee au moins sommaire de DECwindows et de son architecture
- Elle s’adresse au "system manager" ayant deja installe DECwindows et pret a le demarrer pour la premiere fois
- L’ensemble de cette presentation est basee sur VMS V5.1-1
- Quelques idees sur VMS V5.2 et VMS V5.3 seront donnees en complement

Le "startup" de DECwindows

Les fichiers de commandes invoques pendant le startup :

Obligatoires : (SYS$MANAGER)

- DECW$STARTUP.COM
- DECW$LOGICALS.COM
- DECW$DEVICE.COM
- DECW$CHECK_PARAMS.COM
- DECW$STARTSERVER.COM
- DECW$KILLSERVER#.COM
- DECW$RGB.COM
- DECW$STARTLIBS.COM
- DECW$STARTAPPS.COM

Le "startup" de DECwindows

Optionels : (SYS$MANAGER)

- DECW$SERVER_DEVICE_SETUP_xx.COM
- DECW$PRIVATE_SERVER_SETUP.COM

Le "Startup" de DECwindows

Les images DECwindows invoquees pendant le startup :

- Images du server :
  - **SYS$SYSTEM:DECW$SERVER_MAIN.EXE**
  - **SYS$SYSTEM:DECW$START_MAIN_xx.EXE**
- Images transport :
  - **SYS$SHARE:DECW$TRANSPORT_COMMON.EXE**
  - **SYS$SHARE:DECW$TRANSPORT_DECNET.EXE**
  - **SYS$SHARE:DECW$TRANSPORT_LOCAL.EXE**

Le "Startup" de DECwindows

Les images DECwindows invoquees pendant le startup :

- Images des input drivers :
  - -- **SYS$LOADABLE_IMAGES:INDRIVER.EXE (Input Driver)**
  - -- **SYS$LOADABLE_IMAGES:IMDRIVER.EXE (Mouse Driver)**
  - -- **SYS$LOADABLE_IMAGES:IKDRIVER.EXE (Keyboard Driver)**
- Images des drivers de peripheriques graphiques :
  - -- **SYS$LOADABLE_IMAGES:GAADRIVER.EXE (VSII/GPX)**
  - -- **SYS$LOADABLE_IMAGES:GABDRIVER.EXE (VS2000, VS3100 couleur)**
  - -- **SYS$LOADABLE_IMAGES:GBBDRIVER.EXE (VS3520, VS3540)**

- **SYS$LOADABLE_IMAGES:GCADRIVER.EXE (VS II)**
- **SYS$LOADABLE_IMAGES:GCBDRIVER.EXE (VS2000, VS3100 noir et blanc)**

- Images des pseudo-device drivers :
  - **SYS$LOADABLE_IMAGES:TWDRIVER.EXE**
  - **SYS$LOADABLE_IMAGES:PYDRIVER.EXE**
  - **SYS$LOADABLE_IMAGES:WSDRIVER.EXE (Display information device)**

- Images des run-time :
  - **SYS$SHARE:DECW$XLIBSHR.EXE**
  - **SYS$SHARE:DECW$DWTLIBSHR.EXE**
  - **SYS$SHARE:CDA$ACCESS.EXE**

Le "Startup" de DECwindows

- Autres images :
  - -- **SYS$SHARE:DECW$LOGINOUT.EXE**
  - -- **SYS$SYSTEM:LOGINOUT.EXE**
  - -- **SYS$SYSTEM:DECW$SESSION.EXE**
  - -- **SYS$SYSTEM:DECW$TERMINAL.EXE**
  - -- **SYS$SYSTEM:DECW$WINMGR.EXE**

Le "Startup" de DECwindows

Les noms logiques utilises pour customiser le startup :

- DECW$STARTUP.COM :
  - **DECW$IGNORE_VERSION : Pas de verification de version de VMS**
  - **DECW$IGNORE_SUBPROCESS : Pas de spawn de soi-meme**
  - **DECW$IGNORE_AUTOGEN : Pas de verification des para-metres SYSGEN**
  - **DECW$IGNORE_DECNET : Pas d'attente de DECnet pour demarrer**

Le "Startup" de DECwindows

DECW$DEVICE.COM :

- **--** **DECW$INSTALL_100DPI : Utilisation des caracteristiques de moniteur 100 DPI**
- **--** **DECW$PRIMARY DEVICE : Permet le support des stations "multi-tete" (non supporte en V1.0)**

DECW$STARTSERVER.COM :

- **--** **DECW$SERVER_DUMP : Permet de placer le qualifier /DUMP lors du demarrage du server**
- **--** **DECW$SERVER_WSDEF : Valeur de WSDEF pour le server**
- **--** **DECW$SERVER_WSQUOTA : Valeur de WSQUOTA pour le server**
- **--** **DECW$SERVER_WSEXTENT : Valeur de WSEXTENT pour le server**

Le "Startup" de DECwindows

- DECW$STARTAPPS.COM :
  - -- **DECW$DEFAULT_TRANSPORT : Permet de definir le tran- port par defaut pour le server**

- LOGINOUT.EXE :
  - -- **DECW$LOGIN_LOG : Nom du fichier log de l'on peut speci- fier pour y placer les messages de LOGINOUT.EXE**
  - -- **DECW$LOGIN_BACKGROUND : Permet de supprimer le logo Digital**

Le "startup" de DECwindows

Les parametres de DECW$STARTUP.COM :

- **P1** **RESTART**
SERVER
LIBS
APPS
NULL ("")
- **P2** **Numero de Server**
- **P3** **Peripherique ecran**
- **P4** **Peripherique clavier**
- **P5** **Peripherique souris**

P2-P5 ne sont pas utilises par DECW$STARTUP.COM mais passes aux autres procedures :

- **P2-P5** **DECW$STARTSERVER.COM**
- **P2** **DECW$STARTLIBS.COM**
- **P2** **DECW$STARTAPPS.COM**

Le "Startup" de DECwindows

L’utilisation de DECW$STARTUP.COM :

- **Appelee par SYSTARTUP_V5.COM pour demarrer DECwindows pendant le startup (sans parametre d’appel)**
- **$ @SYS$MANAGER:DECW$STARTUP.COM**
- **Il n’est pas necessaire de placer cette ligne apres le demarrage de DECnet (@STARTNET) car cette procedure attend le demarrage de DECnet**
- **Peut etre utilisee par un utilisateur privilegie pour effectuer un RESTART du display server (arrete le display server courant)**
- **$ @SYS$MANAGER:DECW$STARTUP.COM RESTART**
- **Peut etre utilise par un utilisateur privilegie pour "demarrer" server, libraries ou applications**
- **$ @SYS$MANAGER:DECW$STARTUP.COM**

SERVER
LIBS
APPS
"LIBS,APPS"

Le "Startup" de DECwindows

- **"Demarrer" les libraries consiste a definir les noms logiques et a installer les images partageables**

Le "Startup" de DECwindows

L’algorithmique de DECW$STARTUP.COM :

1. **Verification de la version de VMS. Doit etre V5.1 (A,B,V) ou V5.1-1**

2. **Verification de la liste des privileges requis**

3. **Spawn de soi-meme avec creation d’un sous-process de nom DECW$STARTUP afin de ne pas bloquer le startup en attendant le demarrage de DECnet**

4. **Definition des noms logiques DECwindows (Execution de DECW$LOGICALS.COM)**

5. **Definition du fichier de ressources X utilise par TPU**

6. **Verification du fait que l’on effectue un RESTART du display server; si oui :**

    a. **Envoi d’un "warning" et question pour savoir si l’on continue**

    b. **Execution de DECW$DEVICE.COM pour positionner les symboles specifiques au server et chargement des drivers appropries**

c. **Execution de DECW$STARTSERVER.COM pour arreter le server courant et demarrer un nouveau "display server"**

d. **Exit**

7. **Execution de DECW$DEVICE.COM pour positionner les symboles specifiques au server et chargement des drivers appropries**

8. **Verification des parametres SYSGEN et execution d'AUTOGEN si un mauvais parametrage est trouve**

9. **Attente du demarrage de DECnet. Polling toutes les 10 secondes pendant 10 minutes. Arret de la procedure si DECnet n'est pas demarre a la fin du polling**

10. **Verification du fait que la valeur du parametre PIPELINE QUOTA de l'EXECUTOR DECnet est au moins a 10000**

11. **Installation des images partageables de transport**

12. **Appel de la procedure de commande appropriee en fonction de la valeur de P1 :**

a. **P1 = "", execution de:**

(1) **DECW$STARTSERVER.COM**

(2) **DECW$STARTLIBS.COM**

(3) **DECW$STARTAPPS.COM**

b. **P1 = "SERVER", execution de:**

(1) DECW$STARTSERVER.COM

c. P1 = "LIBS", execution de :

(1) DECW$STARTLIBS.COM

d. P1 = "APPS", execution de :

(1) DECW$STARTAPPS.COM

Le "Startup" de DECwindows

Les parametres de DECW$LOGICALS.COM :

- **P1** **Nom de la table de noms logiques a creer**
- **P2** **Symbole utilise par les autres procedures de com - mande DECwindows pour definir les noms logiques dans la table de noms logiques specifiee en P1**
- **P3** **Determine si l'on place ou non la table de noms logiques dans la table par defaut LNM$FILE_DEV**

Les fonctions de DECW$LOGICALS.COM :

- **Creation d'une table de noms logiques specifique a DECwindows pour le display server**
- **Creation d'un symbole utilise par les autres fichiers de commande DECwindows pour definir les noms logiques dans la table de noms logiques specifiee**

Le "Startup" de DECwindows

Les tables de noms logiques :

- **DECW$LOGICAL_NAMES**

  Table de noms logiques ajoute a LNM$FILE_DEV puisque ces noms logiques sont normalement utilises par les applications DECwindows

- **DECW$SERVERn_TABLE**

  Table de noms logiques apropre au display server n. Cette table n’est donc pas ajoutee a LNM$FILE_DEV

Le "Startup"de DECwindows

Les noms logiques DECwindows :

- **DECW$LOGICAL_NAMES :**

CDA$LIBRARY	Guide de style DDIF
DECW$BOOK	Directory de documentation online
DECW$EXAMPLES	Directory d'examples DECwindows
DECW$INCLUDE	Include files pour la programmation C MIT
DECW$KEYMAP	Definition du mapping des claviers util ise par le Session Manager
DECW$SERVER_TRANSPORTS	Liste des differents types de transports
DECW$STARTVUECOM	Fichier de commande de startup pour FileView
DECW$SYSTEM_DEFAULTS	Directory de fichiers de ressources DECwindows systeme
DECW$USER_DEFAULTS	Directory de fichiers de ressources DECwindows prives
DECW$WINMGREXE	Image du Window Manager
DECW$XLIBERRDB	Database des ressources d'erreurs Xlib
VUE$LIBRARY	Directory contenant les fichiers de commmande du FileView et le profile systeme par defaut

- **DECW$SERVERn_TABLE :**

DECW$SERVER_SCREENS	Nom du peripherique ecran
DECW$SERVER_KEYBOARD	Nom du peripherique clavier
DECW$SERVER_POINTER	Nom du peripherique souris
DECW$SERVER_XSIZE_IN_PIXELS	Largeur du "display" en pixel
DECW$SERVER_YSIZE_IN_PIXELS	Hauteur du "display" en pixel

Le "Startup"de DECwindows

DECW$MONITOR_DENSITY	**Densite du moniteur (75 ou 100 dpi)**
DECW$FONT	**Directory des fonts**
DECW$KEYMAP	**Directory du "mapping" des claviers**
DECW$DEFAULT_KEYBOARD_MAP	**Mapping par defaut du clavier**
DECW$DEFAULT_FONT	**Font chargee par defaut dans un GC**
DECW$COLOR	**Specifie si le moniteur est couleur**
DECW$SERVER_DISAGLE_CH	**Met en fonction le condition handler du server**
DECW$RGB_*	**Noms des couleurs utilises par XStoreNamedColor et XAllocNamed Color**

Le "Startup" de DECwindows

L’algorithmique de DECW$LOGICAL.COM :

1. Assignation des valeurs par defaut pour le nom de la table de noms logiques, le nom du symbole et une variable booleenne pour determiner si l’on place ou non la table dans LNM$FILE_DEV. Les defauts sont :

    a. Nom de la table : DECW$LOGICAL_NAMES

    b. Nom du symbole: DECW$DEFINE

    c. Booleen: TRUE

2. Creation de la table de noms logiques specifiee. Si la table existe, elle n’est pas ecrasee

3. Ajout de la table a LNM$FILE_DEV si le booleen est vrai

4. Creation du symbole speficie pour definir les noms logiques dans la table specifiee ($ DEFINE/TABLE=nom_de_table)

5. Exit

Le "Startup" de DECwindows

L’utiliation de DECW$DEVICE.COM :

- **Cette procedure de commande est utilisee pour :**
  - **Definir des symboles utilises par d’autres procedures de commande DECwindows**
  - **Charger les drivers necesaires a DECwindows**
- **Cette procedure n’utilise pas de parametres**
- **Si les drivers VWS sont deja charges, DECW$DEVICE.COM genere une erreur**

Le "Startup" de DECwindows

L’algorithmique de DECW$DEVICE.COM :

1. **Initialisation des symboles DECW$DEVICE, DECW$DEVICE_TYPE et CPU**

2. **On s’assure que les input device drivers existent**

3. **Positionnement de valeurs locales pour fonts et keyboards. ATTENTION, cette procedure a un BUG !!**

4. **Parametrage des caracteristiques des peripheriques DECwindows qui existent sur ce systeme. Attention, cette procedure n’a pas connaissance de la caracteristique d’intensite (voir plus loin)**

Le "Startup" de DECwindows

- **Le nom logique DECW$PRIMARY_DEVICE n'est pas utilise avant VMS V5.3 (DECwindows V2) qui supportera les stations "multi-tetes"**

- **DECW$DEVICE.COM ne parametre pas correctement la search-list de keymaps; au lieu de :**

  SYS$COMMON:[SYS$KEYMAP.USER],
  SYS$COMMON:[SYS$KEYMAP.SYSTEM]
  il faudrait
  SYS$COMMON:[SYS$KEYMAP.DECW.USER],
  SYS$COMMON:{SYS$KEYMAP.DECW.SYSTEM]

- **Ceci est parametre correctement dans DECW$STARTAPPS.COM donc correctement pour le Session Manager, mais par pour le server a travers DECW$PRIVATE_SERVER_SETUP.COM**

- **Le system manager doit ajouter les lignes suivantes a DECW$PRIVATE_SERVER_SETUP.COM :**

  $decw$keymap =="sys$common:[sys$keymap.decw.user],"+-
  "sys$common:[sys$keymap.decw.system]"

Le "Startup" de DECwindows

Les symboles de DECW$DEVICE.COM :

- **Valeurs pour VAXstation II monochrome (GCADRIVER)**

DECW$DEVICE	GCA0:
DECW$DEVICE_TYPE	VSII
DECW$KEYBOARD	GCA1:
DECW$POINTER	GCA2:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	FALSE

- **Valeurs pour VAXstation 2000 monochrome (GCBDRIVER)**

DECW$DEVICE	GCA0:
DECW$DEVICE_TYPE	VS2000
DECW$KEYBOARD	TTA0:
DECW$POINTER	TTA1:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	FALSE

- **Valeurs pour VAXstation 3100 monochrome (GCBDRIVER)**

DECW$DEVICE	GCA0:
DECW$DEVICE_TYPE	VS3100
DECW$KEYBOARD	TTA0:

DECW$POINTER	TTA1:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	FALSE

- **Valeurs pour VAXstation II/GPX (GAADRIVER)**

DECW$DEVICE	GAA0:
DECW$DEVICE_TYPE	VSII_GPX
DECW$KEYBOARD	GAA1:
DECW$POINTER	GAA2:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	TRUE

- **Valeurs pour VAXstation 2000 Couleur (GABDRIVER)**

DECW$DEVICE	GAA0:
DECW$DEVICE_TYPE	VS2000_GPX
DECW$KEYBOARD	TTA0:
DECW$POINTER	TTA1:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	TRUE

- **Valeurs pour VAXstation 3200/3500 (GAADRIVER)**

DECW$DEVICE	GAA0:
DECW$DEVICE_TYPE	MAYFAIR_GPX
DECW$KEYBOARD	GAA1:
DECW$POINTER	GAA2:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	TRUE

- **Valeurs pour VAXstation 3100 couleur (GABDRIVER)**

DECW$DEVICE	GAA0:
DECW$DEVICE_TYPE	VS3100_GPX
DECW$KEYBOARD	TTA0:
DECW$POINTER	TTA1:
DECW$XSIZE_IN_PIXEL	1024
DECW$YSIZE_IN_PIXEL	864
DECW$COLOR	TRUE

- **Valeurs pour VAXstation 3520/3540 (GBBDRIVER)**

DECW$DEVICE	GBA0:
DECW$DEVICE_TYPE	VS35x0-Leggs
DECW$KEYBOARD	TTA0:
DECW$POINTER	TTA1:
DECW$XSIZE_IN_PIXEL	1280
DECW$YSIZE_IN_PIXEL	1024
DECW$COLOR	TRUE

Le "Startup" de DECwindows

Utilisation de DECW$CHECKPARAMS.COM :

- **Parametres : cette procedure n'utilise qu'un parametre P1**
- **Si P1 = "AUTOGEN" les warnings a propos des parametres SYSGEN incorrects ne sont pas indiques a l'utilisateur**
- **Fonction : Verifier si les parametres SYSGEN suivants sont positionnes aux valeurs correctes :**

GBLSECTIONS	>300
GBLPAGES	>15000
GBLPAGEFIL	>6024
SWPOUTPGCNT	=500
MAXBUF	>2048
VIRTUALPAGCNT	>20000
PQL_DPGFLQUOTA	>5000
PQL_MASTLM	>100
PQL_MBIOLM	>40
PQL_MDIOLM	>40
PQL_MPRCLM	>8
PQL_MFILLM	>8
PQL_MBYTLM	>32768
PQL_MWSQUOTA	>60
PQL_MWSDEFAULT	>60
PQL_MWSEXTENT	>500
WINDOW_SYSTEM	=1

Le "Startup" de DECwindows

Les parametres de DECW$STARTSERVER.COM :

- **Cette procedure recoit ses parametres depuis DECW$STARTUP.COM. Le parametre le plus important est le parametre P5 : RESTART**

- **Parametres :**

P1	**Numero de server**
P2	**Nom du peripherique ecran**
P3	**Nom du peripherique clavier**
P4	**Nom du peripherique souris**
P5	**Parametre RESTART**

- **Il n’est pas recommande de lancer cette procedure en utilisant les parametres de P1 a P4 (risques d’erreurs)**

Le "Startup" de DECwindows

L'algorithmique de DECW$STARTSERVER.COM :

La fonction primaire de cette procedure est de demarrer le "display server"

1. **Exit si l'on n'est pas sur une station**

2. **Verification du fait que l'on effectue un "RESTART" du server. Si oui :**

   a. **Creation d'une procedure de commande DECW$KILLSERVERn pour arreter le server courant s'il existe**

   b. **Appel de DECW$STARTUP.COM. Tourne comme un process detache de nom "Server n Restart"**

   c. **Exit**

3. **Verification du fait qu'un server existe deja et utilise deja le peripherique DECW$DEVICE et EXIT si le reference count est different de 0**

4. **Initialisation de la table de noms logiques propre au server en appelant DECW$LOGICALS.COM et DECW$RGB.COM**

5. Execution du fichier de commande specifique aux peripheriques et du fichier de startup specifie par l'utilisateur s'ils existent :

   DECW$SERVER_DEVICE_SETUP_xx.COM
   DECW$PRIVATE_SERVER_SETUP.COM

   ou xx designe les 2 premieres lettres de DECW$DEVICE

6. Positionnement des caracteristiques des moniteurs 100 dpi si necessaire (le defaut est 75 dpi)

7. Definition des noms logiques du server dans la table DECW$SERVERn_TABLE

8. "Ecrasement" des defauts avec les valeurs des parametres passes a la procedure

9. Suppression des messages OPCOM sur OPA0: s'il l'on est satellite d'un cluster (pas en V5.2)

10. Nettoyage des fichiers de donnees et des fichiers de log des precedents servers. On ne laisse qu'une seule copie. Les fichiers impliques sont les suivants :

    DECW$SERVER_n_INPUT.DAT (SYS$INPUT, non utilise)
    DECW$SERVER_n_OUTPUT.DAT (SYS$OUTPUT du server)
    DECW$SERVER_n_ERROR.DAT (SYS$ERROR du server)

11. Verification de la presence du nom logique DECW$SERVER_DUMP pour dire au server s'il faut sauvegarder un dump en cas de crash du server. Definition du nom logique DECW$SERVER_DISABLE_CH a "Y" si les dumps sont sauvegardes et a "N" si les dumps ne sont pas sauvegardes. Ce nom logique est utilise pour mettre en fonction ou supprimer les condition handlers du server

12. **Verification de l'existence de valeurs de working set server specifies par l'utilisateur grace aux noms logiques :**

    DECW$SERVER_WSDEF
    DECW$SERVER_WSQUOTA
    DECW$SERVER_WSEXTENT

    Attention, il n'y a pas de verification de validite (defaut : 500, 1300, 4000)

13. **Determination du nom de l'executable du server. Par defaut, SYS$SYSTEM:DECW$SERVER_MAIN.EXE. Pour permettre des fixations de bugs et pour le support des VS3520/3540, une verification est faite pour determiner si un fichier de nom SYS$SYSTEM:DECW$SERVER_MAIN_xx.EXE existe (xx designe les 2 premieres lettres du peripherique designe par le nom logique DECW$DEVICE)**

14. **Execution de l'image trouvee precedemment comme un process detache. C'est ici que les autres parties du server sont chargees et que l'objet X$X0 est cree**

15. **Exit**

Le "Startup" de DECwindows

DECW$RGB.COM :

- Definit toutes les couleurs "nommees" (NAMED) que le server connaitra lors des appels Xlib XtoreNamed-Color et XAllocNamedColor
- Ces noms logiques peuvent etres trouves dans la table DECW$SERVERn_TABLE. La syntaxe generique du fichier est :

  $decw$define "DECW$RGB_colorname"
  "red_intensity,green_intensity,blue_intensity"
- Les valeurs d’intensite utilisees pour rouge, vert, bleu sont des valeurs codees sur 16 bits (0-65535) que le server utilise. Attension aux "" pour conserver majuscules et minuscules
- Si le system manager desire ajouter ses propres couleurs, il est possible de creer un fichier DECW$PRIVATE_RGB.COM et de l’appeler depuis DECW$PRIVATE_SERVER_SETUP.COM

Le "Startup" de DECwindows

Utilisation et algorithmique de DECW$STARTLIBS.COM :

- Fonction : Definition de noms logiques DECwindows, installation de Xlib, du Toolkit DECwindows et des librairies partageables CDA. Connexion de WS-DRIVER
- Parametres : Aucun
- Algorithmique :

1. Definition des noms logiques DECwindows pour l'environnement de developpement
2. Creation des tables de noms logiques RMS$SEMANTIC_TAGS et RMS$SEMANTIC_OBJECTS et definition des noms logiques pour DDIF, DOTS, TDIF

3. Installation des images partageables :

    DECW$XLIBSHR.EXE (Xlib)
    DECW$DWTLIBSHR.EXE (DECwindows Tk)
    CDA$ACCESS.EXE (CDA Tk)

4. Positionnement de la semantique de style CDA Par defaut dans CDA$LIBRARY comme etant DDIF

5. Connexion du pseudo-device driver utilise par la commande DCL $SET DISPLAY. La routine Xlib XOpen-Display utilise aussi ce peripherique por obtenir des renseignements sur la WS sur laquelle on veut etablir un connexion

6. Exit

Le "Startup" de DECwindows

Utilisation de DECW$STARTAPPS.COM :

- Fonctions :
  - -- Definition des noms logiques pour les applications DECwindows
  - -- Connexion des pseudo-terminal drivers DECterm
  - -- Installation des images communement utilisees par DECwindows
  - -- Creation du premier "display" DECwindows
  - -- Demarrage de la premiere application DECwindows qui cree la boite de "login"
- Parametres :
  - -- P1 : Cette procedure se termine juste apres l’installation des images communement utilisees par DECwindows si P1 n’est pas NULL ("")

Le "Startup" de DECwindows

Algorithmique de DECW$STARTAPPS.COM :

1. Definition des noms logiques qui sont normalement utilises par les applications

2. Connexion des pseudo-terminal device drivers DEC-term, TWDRIVER et PYDRIVER

3. Installation des images :

    SYS$SYSTEM:LOGINOUT.EXE
    SYS$SHARE:DECW$LOGINOUT.EXE
    SYS$SYSTEM:DECW$SESSION.EXE
    SYS$SYSTEM:DECW$TERMINAL.EXE
    SYS$SYSTEM:DECW$WINMGR.EXE

4. Si l’on n’est pas sur une WS ou si P1 n’est pas NULL, Exit

5. Creation d’un pseudo-device et definition du nom logique DECW$DISPLAY en utilisant la commande DCL $SET DISPLAY/CREATE

6. Run DECW$STARTLOGIN.EXE pour creer la boite de login
7. Exit

Le "Startup" de DECwindows

Le fichier de "customization" DECW$PRIVATE_SERVER_SETUP.COM :

- Fichier reserve au system manager pour modifier les caracteristiques du server et/ou d’autres caracteristiques DECwindows
- C’est LE fichier a modifier
- Ce fichier est structure de telle facon qu’il soit possible de l’executer depuis n’importe quel noeud d’un cluster
- Noms logiques qui peuvent etre modifies :

  DECW$DEFAULT_TRANSPORT
  DECW$RGB_*
  DECW$SERVER_DUMP
  DECW$SERVER_WSDEF
  DECW$SERVER_WSQUOTA
  DECW$SERVER_WSEXTENT
  DECW$LOGIN_LOG
  DECW$LOGIN_BACKGROUND

- Symboles qui peuvent etre modifies :

  DECW$SERVER_SCREENS
  DECW$KEYBOARD
  DECW$POINTER
  DECW$XSIZE_IN_PIXELS
  DECW$YSIZE_IN_PIXELS
  DECW$FONT
  DECW$KEYMAP
  DECW$DEFAULT_KEYBOARD_MAP
  DECW$DEFAULT_FONT
  DECW$COLOR

Le "Startup" de DECwindows

Algorithmique de DECW$PRIVATE_SERVER_SETUP.COM

1. Definition de la liste de noeuds separes par "/"
2. "Parsing" de la liste des noms de noeuds et "jump" a la routine do_NODENAME
3. Definitions propres au noeud
4. Exit

Le "Startup" de DECwindows

Exemple de DECW$PRIVATE_SERVER_SETUP.COM

```
$ ! This is a sample server customization file.  You can specify a number of
$ ! things about how you want the server to configure itself by defining some
$ ! DCL symbols here. You can use this file to specify the following:
$ !
$ !        - The density of your screen.
$ !
$ !        - Whether the screen is monochrome or color.  You must specify
$ !          monochrome or color only for hardware where both screen types
$ !          are available, such as the VAXstation II/GPX or the VAXstation
$ !          2000/GPX)
$ !          For hardware like the VAXstation II or the VAXstation 2000 which
$ !          support monochrome only, the specification here will be ignored.
$ !          look for font files by default.
$ !
$ !       -  The default keyboard map.  This specifies what language your
$ !          keyboard is designed for.  You also specify the country in the
$ !          DECwindows session manager setup, but the specification here
$ !          will prevail before you log in.
$ !
$ !       You make these specifications by defining DCL symbols.  The symbols
$ !       are listed in the following table:
$ !
$ !       Characteristi  Symbol                     Values          Default*
$ !-----------------------------------------------------------------------------
$ !       Density        DECW$SERVER_DENSITY        75 or 100       75
$ !       Color          DECW$COLOR                 T OR F T**
$ !       Keyboard Map   DECW$DEFAULT_KEYBOARD_MAP NORTH_AMERICAN_LK201LA***
$ !-----------------------------------------------------------------------------
$ ! * Defaults may be changed in future software versions, or for future
$ ! hardware.
$ ! **On a monochrome-only hardware, the default is F, and a value of T will
$ !   have no effect.
$ !***Valid values are the file names (without extension) in the directories
```

```
$ !   SYS$COMMON:[SYS$KEYMAP.DECW.SYSTEM] and
$ !   SYS$COMMON:[SYS$KEYMAP.DECW.USER]
$ !
$ !---------------------------------------------------------------------------
$ !
$ !       This file is setup so that you can customize it to act differently
$ !       depending on the node that it is being executed on.  This allows a
$ !       single file to reside in the cluster common SYS$MANAGER directory
$ !       and yet start different workstations in the cluster with different
$ !       characteristics.  To use this, replace the definition of node_list
$ !       with your own list of node names, and add a label "do_nodename"
$ !       with the DCL code that you want to execute on that node.
$ !
$ !       Here is a sample list of valid node names:
$ !
$ node_list = "GPXBW/GPXCOL/DUTCH/UK100"
$ !
$ !       Find out if we have node that this file has been customized for.
$ !       If so, go there.  Otherwise, do the default.
$ !
$ node_number = 0
$ node_loop:
$   node = f$element(node_number,"/",node_list)
$   if node .eqs. "/" then goto do_default
$   if (node .eqs. f$getsyi("NODENAME"))then goto do_'node
$   node_number =  node_number + 1
$   goto node_loop
$ !
$ ! Here is a sample list of definitions for each node:
$ !
$do_default:
$ !Don't know anything about it.  Don't specify anything.
$exit
$ !
$do_gpxbw:
$ ! A gpx monochrome server.  We need to specify a monochrome monitor
$ decw$color == "F"
$exit
$ !
$do_gpxcol:
$ ! A gpx color server.  Color is the default, but we will specify it for
$ ! good measure.
$ decw$color == "T"
$exit
$ !
$do_dutch:
$ decw$default_keyboard_map == "DUTCH_LK201LH_DP"
$exit
$ !
$do_UK100:
```

```
$ !This user wants a UK keyboard and has a 100 dpi monitor
$ decw$default_keyboard_map == "DECW$UK_LK201RE"
$ decw$server_density == 100
$exit
```

Le "Startup" de DECwindows

Utilisation de
DECW$PRIVATE_SERVER_SETUP.COM

- Les procedures de startup de DECwindows determinent la majorite des variables de configuration du server. Mais 2 variables ne sont pas definies :
  - -- Si l’on a une VS2000/GPX, VS3xxx/GPX, VSII/GPX, les procedures de startup supposent que l’on a affaire a un moniteur couleur. Si ce n’est pas le cas, il faut parametrer DECW$COLOR
  - -- Par defaut, le clavier est un "North American Keyboard" (LK201-AA ou -LA)
- Si ce n’est pas le cas :
  - -- Faire une copie de SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE en .COM

-- Pour les GPXs monochromes :

$decw$color == "FALSE"

-- Pour les claviers non "North American" :

$decw$default_keymap == "xxxxx"

ou xxxxx peut etre trouve en table 2-4 de VMS V5.1 Installation Guide

-- Placer aussi la modification de DECW$KEYMAP pour fixer le bug cite plus haut

Le "Startup" de DECwindows

Algorithmique de DECW$STARTLOGIN.EXE

1. Translation du nom logique DECW$DISPLAY. Examen du nom logique dans les table JOB et PROCESS. DECW$DISPLAY doit etre defini dans les 2 tables depuis DECW$STARTAPPS.COM

2. Envoi d'un message dans la boite aux lettres du Job-Control

3. "Unsollicited Input" sur le peripherique WSA# defini par DECW$DISPLAY pour demander a Job Control de demarrer LOGINOUT

4. Job Control cree un process executant LOGINOUT.EXE et de nom "_WSA#" (ou blanc). SYS$INPUT, SYS$OUTPUT et SYS$ERROR pointent sur WSA#. On voit un nom de process blanc si l'utilisateur a tente 3 fois de se connecter sans succes. Lors d'un "reset" du server, le nom de process est blanc

Le "Startup" de DECwindows

Algorithmique de LOGINOUT.EXE sous DECwindows :

1. $GETDVI sur SYS$INPUT (WSA#). Obtention :
   a. Du type de station
   b. D’un mode de login interactif
   c. D’un flag indiquant que l’on est en "mode" DECwindows
2. LIB$FIND_IMAGE_SYMBOL pour SYS$SHARE:DECW$LOGINOUT.EXE. LOGINOUT et DECW$LOGINOUT.EXE communiquent pour effectuer le login
3. Assignation de SYS$OUTPUT sur OPA0: (ou la traduction du nom logique DECW$LOGIN_LOG). Ce nom logique doit etre defini en mode EXEC dans la table DECW$LOGICAL_NAMES. Ceci peut etre fait en utilisant DECW$DEFINE dans DECW$PRIVATE_SERVER_SETUP.COM :

decw$define DECW$LOGIN_LOG SYS$MANAGER:DECW$LOGIN.LOG

4. Appel depuis LOGINOUT de DECW$LOGINOUT :

   a. Ouverture du display et affichage du logo

   b. Utilisation du fichier-
   DECW$SYSTEM_DEFAULTS:
   DECW$LOGIN.DAT
   comme fichier de ressources

   c. Map de la boite de dialogue du login

Le "Startup" de DECwindows

Le "Login" de DECwindows

Les fichiers de commandes invoques pendant le login :

SYS$SYSTEM :

- LOGINOUT.EXE
- DECW$TERMINAL.EXE
- DECW$SESSION.EXE
- DECW$WINMGR.EXE
- VUE$MASTER.EXE

SYS$SHARE:

- DECW$LOGINOUT.EXE

SYS$MANAGER :

- DECW$STARTSM.COM
- DECW$STARTVUE.COM
- DECW$SYLOGIN.COM

SYS$LOGIN (ou DECW$SYSTEM_DEFAULTS) :

- DECW$SM_COLOR.DAT
- DECW$SM_BW.DAT
- DECW$SM_GRAY.DAT
- DECW$SM_GENERAL.DAT
- DECW$PRINTWGT.UID

SYS$LOGIN :

- DECW$LOGIN.COM
- DECW$XDEFAULTS.DAT

Le "Login" de DECwindows

Overview de l’ algorithmique du LOGIN de DECwindows :

- LOGINOUT et DECW$LOGINOUT travaillent ensemble pour obtenir un couple USERNAME/PASSWORD valide
- Demarrage du Window Manager DECwindows
- Chainage des fichiers de commande DECW$SYLOGIN.COM, DECW$LOGIN.COM et DECW$STARTSM.COM
- Demarrage du Session Manager qui demarre le DECterm controller et optionellement un FileView

Le "Login" de DECwindows

Algorithmique du LOGIN de DECwindows :

On commence avec la boite de dialogue de login dans laquelle l'utilisateur entre USER-NAME/PASSWORD.

1. DECW$LOGINOUT rappelle LOGINOUT pour valider le couple (Username/Password)
2. Si un probleme survient, LOGINOUT rapelle DECW$LOGINOUT pour reporter le probleme dans la boite de dialogue (si l'on n'est pas en mode EXEC ou KERNEL et si la variable display n'est pas vide). Si l'on est en mode EXEC ou KERNEL ou si la variable display est vide, le probleme est envoye en broadcast sur OPA0:. Si une erreur Toolkit survient, cette erreur est envoyee sur SYS$OUTPUT (OPA0: ou le fichier pointe par DECW$LOGIN_LOG).
   Si l'erreur n'est pas severe, on retourne a LOGI-NOUT. Si l'erreur est severe, on envoie un message a Job Control, on effectue une "unsollicited input" sur WSA# pour redemarrer la session de login. Cela a

pour effet de reinitialiser le server car l'erreur severe est probablement due a un probleme de communication

3. LOGINOUT initialise les informations utilisateur du process et appelle DECW$LOGINOUT pour initialiser le CLI. DECwindows execute 3 fichiers a la place de SYLOGIN et LOGIN :

   SYS$MANAGER:DECW$SYLOGIN.COM
   SYS$LOGIN:DECW$LOGIN.COM
   SYS$MANAGER:DECW$STARTSM.COM

   Retour a LOGINOUT

4. LOGINOUT valide les restrictions d'acces utilisateur et appelle DECW$LOGINOUT pour positionner SYS$INPUT et SYS$OUTPUT

5. LOGINOUT met a jour les enregistrements de l'UAF et appelle DECW$LOGINOUT pour positionner le password flag de l'UAF (mot de passe expire). DECW$LOGINOUT rappelle LOGINOUT pour :

   a. Generation de mots de passe

   b. Verification des erreurs de syntaxe

   c. Enregistrement du nouveau mot de passe dans l'UAF

DECW$LOGINOUT retourne ensuite a LOGINOUT

6. LOGINOUT appelle DECW$LOGINOUT pour positionner le parametre DECwindows "ressource property" sur la "root window". D'apres le type du display, determination du fichier a lire pour parametrer les ressources couleur du Session Manager :

   a. DECW$SM_COLOR.DAT (systeme multi-plan couleur)

   b. DECW$SM_BW.DAT (frame buffer monochome)

   c. DECW$SM_GRAY.DAT (systeme multi-plan monochrome)

   Puis, DECW$SM_GENERAL.DAT est lu pour definir les ressources du Session Manager. Ces fichiers sont cherches d'abord dans SYS$LOGIN puis dans DECW$SYSTEM_DEFAULTS. SYS$LOGIN:DECW$XDEFAULTS.DAT est lu en dernier pour le parametrage de ressources generales. Un certain nombre d'items comme couleur, clavier sont positionnes dans la "property" RESSOURCE_MANAGER de la "root window". Le toolkit propage ensuite ces "properties" aux autres applications au fur et a mesure de leur lancement

7. Apres avoir lu DECW$SM_GENERAL.DAT, DECW$LOGINOUT parametre la securite de la sta-

tion pour cette session. Dans le fichier DECW$SM_GENERAL.DAT, il y a 2 "ressources" qui sont le resultat du parametrage de la securite :

sm.host_list
sm.num_hosts

Pour ces "ressources" et le "username" de cette session, DECW$LOGINOUT passe outre a la securite du server. Ceci permet au Window Manager et au Session Manager de se connecter au server. Par defaut aucune permission n'est donnee au server quand le Session Manager ne tourne pas. Si la valeur num_hosts est plus grande que le nombre d'entrees dans la liste host_list, le server effectue un ACCVIO (BUG)

8. DECW$LOGINOUT demarre le Window Manager. C'est un process detache utilisant l'image referencee par le nom logique DECW$WINMGREXE trouve dans la table DECW$LOGICAL_NAMES. Le Window Manager a les caracteristiques suivantes :

   SYS$INPUT = NL:
   SYS$OUTPUT = WSA#
   SYS$ERROR = NL: (ou la traduction du nom logique DECW$WINMGR_OUTPUT qui doit etre defin comme un nom logique SYSTEM). Il ne faut pas definir ce nom logique dans DECW$SYLOGIN.COM ou DECW$LOGIN.COM car ces fichiers ne sont pas en-

core executes)
Le nom du process est DECW$WM_# ou # est le numero de peripherique WSA#

9. DECW$LOGINOUT retourne a LOGINOUT qui positionne les parametres suivants :

   SYS$INPUT =
   SYS$MANAGER:DECW$STARTSM.COM
   SYS$OUTPUT = SYS$ERROR =
   SYS$LOGIN:DECW$STARTSM.LOG

10. Passage au DCL avec execution des 3 fichiers suivants dans l'ordre :
   SYS$MANAGER:DECW$SYLOGIN.COM
   SYS$MANAGER:DECW$LOGIN.COM
   SYS$MANAGER:DECW$STARTSM.COM

Le "Login" de DECwindows

Algorithmique de DECW$STARTSM.COM :

1. Verification du fait qu’aucune erreur ne soit survenue pendant DECW$SYLOGIN.COM ou DECW$LOGIN.COM. En cas d’erreur, ces erreurs sont ignorees
2. Verification du fait que des erreurs sont survenues dans le Session Manager. En cas d’erreur, Logout
3. Run SYS$SYSTEM:DECW$SESSION.EXE
4. Exit

Le "Login" de DECwindows

```
$ !
$ ! DECW$STARTSM.COM - Initialize the DECwindows session manager
$ ! This command procedure sets up the DECwindows session manager
$ ! development environment.
$ ! It is not advisable to edit this procedure as it may be replaced in
$ ! future software updates.
$ !
$!
$! if no command files executed, then status will be blank
$!
$if $status .EQS. "" then goto do_session
$!
$! if status is ok, no errors in command procedures, so start session manager
$!
$if  $status then goto do_session
$!
$! if error on entry, then check if error came while doing session manager.
$! sys$input points to this file, so errors will cause process to re-enter
$! with an error status
$!
$if f$logical("decw$doing_session") .EQS. "T" then goto do_exit
$!
$! start the session manager - and set logical that remembers this
$!
$do_session:
$define decw$doing_session T
$set working_set/quota=200/nolog
$run sys$system:decw$session
$!
$! log out and delete the logical
$!
$do_exit:
$if f$logical("decw$doing_session") .NES. "" then deassign decw$doing_session
$stop/id=0
$logout
$exit 1
```

Le "Login" de DECwindows

Enchainement des fichiers de startup DECwindows :

- Le nom de process du Session Manager suit les conventions normales de LOGINOUT :

  > USERNAME de la session de login
  >Peripherique WSA# si le nom de process existe deja
  >Il est possible de changer le nom de process depuis DECW$SYLOGIN.COM ou DECW$LOGIN.COM par un $SET PROCESS/NAME=name.

- Dans DECW$SYLOGIN.COM, les working set quotas sont positionnes a 200. Ces quotas sont reserves a l'usage du Session Manager. Le Session Manager est un process actif quand il effectue les operations demandees par l'utilisateur (demarrage de DECterm(s) ou de FileView(s)). Apres ces operations, le Session Manager est "idle" tant que des operations specifiques au Session Manager ne sont pas effectuees. La raison pour laquelle cette reduction de working set est effectuee est que le memory management de

VMS n’est pas tres efficace pour effectuer un trimming de processes qui ne tournent que tres peu frequemment et pendant un temps tres court

- Ce positionnement de quotas affecte tout process cree avec un SPAWN/NOWAIT. Cela peut accroitre la pagination. Il est donc souhaitable de modifier le working set quota de chaque process avant d’effectuer un SPAWN/NOWAIT ou bien de creer des processes detaches
- Les processes DECterm et FileView ne sont pas affectes car ils sont crees DETACHED
- DECW$LOGIN.COM peut ne pas etre execute en placant le qualifier /NOCOMM apres le USERNAME dans la boite de login

Le "Login" de DECwindows

Demarrage de processes :

SPWAN/NOWAIT

- Cela permet de partager les ressources entre le Session Manager et les autres processes crees par SPWAN
- Peut avoir l’inconvenient de faciliter les problemes de ressources
- Peut donc necessiter l’augmentation des quotas dans l’UAF
- Dans DECW$SYLOGIN.COM, le working set quota est positionne a 200. Cette restriction sera imposee a tous les processes crees par SPAWN. Pour lever cette restriction :

-- S’assurer que l’on demarre tous les processes avant la commande SET WORKING_SET/QUOTA dans DECW$SYLOGIN.COM

-- Ajouter un SET WORKING_SET/QUOTA en debut de DECW$LOGIN.COM

-- SPAWN de tous les sous-processes

-- Ajouter la commande SET WORKING_SET/QUOTA=200 en fin de DECW$LOGIN.COM

Le "Login" de DECwindows

Exemple de DECW$SYLOGIN.COM :

```
$ !
$ ! DECW$SYLOGIN.COM - Initialize the DECwindows applications
$ ! This command procedure is executed when a user logs into the workstation
$ ! with DECwindows.  You may run clients from this command file which
$ ! you would like all users of your system to have running when they
$ ! log into their workstation.
$ !

$ SPAWN/NOWAIT/PROCESS="Clock"/OUTPUT=NL: MCR DECW$CLOCK
$ SET WORKING_SET/QUOTA=200
```

Le "Startup" de DECwindows

Exemple de DECW$LOGIN.COM :

```
$!
$! Set Working Set as in UAF
$!
$ SET WORKING_SET/QUOTA=1024
$!
$ Define logicals for FileView and DECterm
$!
$ define DECW$DECTERM_OUTPUT SYS$LOGIN:DECW$DECTERM_OUTPUT.LOG
$ define DECW$VUE_OUTPUT SYS$LOGIN:DECW$VUE_OUTPUT.LOG
$!
$! Start DECwindows Applications
$!
$ SPWAN/NOWAIT/PROCESS="BookReader"/OUTPUT=NL: MCR DECW$BOOKREADER
$ SPWAN/NOWAIT/PROCESS="Calendar"/OUTPUT=NL: MCR DECW$CALENDAR
$ SPWAN/NOWAIT/PROCESS="Cardfiler"/OUTPUT=NL: MCR DECW$CARDFILER
$ SPWAN/NOWAIT/PROCESS="Mail"/OUTPUT=NL: MCR DECW$MAIL
$!
$! Reset working set for Session Manager
$!
$ SET WORKING_SET/QUOTA=200
$ EXIT
```

Le "Login" de DECwindows

Demarrage de processes :

RUN/DETACHED

- Cela permet de resoudre le probleme de ressource precedent
- Le seul probleme est de faire en sorte que les sous-processes reconnaissent le bon "display". Cela necessite generalement l'utilisation d'une procedure de commande. VMS utilise SYS$OUTPUT si le nom logique DECW$DISPLAY n'est pas defini

Le "Login" de DECwindows

Overview de l’algorithmique de DECW$SESSION :

- Entree et Sortie :

  SYS$INPUT = DECW$STARTSM.COM
  SYS$OUTPUT = SYS$ERROR = DECW$SM.LOG

- Creation d’un pseudo-terminal pour recuperer les messages broadcasts
- Etablissement d’un condition-handler
- Demarrage du DECterm Controller
- Creation du Session Manager panel
- Demarrage de FileView (si demande)
- Demarrage de DECterms (si demande)

Le "Login" de DECwindows

Le terminal du Session Manager :

- Ce terminal est "output only"
- Obtention d’un pseudo-terminal TWA#
- Ce nom de terminal est place dans le PCB du process
- Definition de caracteristique de "boadcast" pour ce terminal
- Positionnement d’une AST a executer en cas de "broadcast"

Le "Startup" de DECwindows

L'activation du DECterm Controller depuis le Session Manager :

Caracteristiques :

- Image : SYS$SYSTEM:DECW$TERMINAL.EXE
- SYS$INPUT = NL:
- SYS$OUTPUT = WSA#
- SYS$ERROR = NL: ou traduction du nom logique DECW$DECTERM_OUTPUT. Ce nom logique doit etre defini dans la table process et donc dans DECW$LOGIN.COM
- Nom de process : DECW$TE_# ou # est le numero de peripherique WSA# pointe par SYS$OUTPUT

Le "Startup" de DECwindows

Overview du DECterm Controller :

- Execute comme un process detache depuis DECW$SESSION.EXE
- Definition du nom logique SYS$LOGIN
- Definition de DECW$DISPLAY comme etant WSA#
- Creation d’une boite aux lettres pour communiquer avec le Session Manager :
    - -- Utilise le nom de noeud, le numero de server et le numero d’ecran
    - -- Nom logique : DECW$DECTERM_MAILBOX_node::server.scre en
    - -- Defini dans la table de noms logiques SYSTEM
- Imprime les messages sur le control panel du Session Manager

- Le Session Manager tente un maximum de 10 fois de redemarrer le DECterm controller

Le "Startup" de DECwindows

Caracteristiques du FileView :

- Image : LOGINOUT.EXE
- SYS$INPUT: DECW$STARTVUECOM
  SYS$MANAGER:DECW$STARTVUE.COM
- SYS$OUTPUT: NL: ou traduction du nom logique DECW$VUE_OUTPUT. Ce nom logique doit etre trouve dans la table PROCESS et donc defini dans DECW$LOGIN.COM
- SYS$ERROR : WSA#
- Nom du process : <USERNAME>_VUE# ou # est un compteur pour eviter la duplication des noms de processes
- Flags du process :
  > DETACHED
  >INTERACTIVE
  >NOPASSWORD

Le "Startup" de DECwindows

Exemple de DECW$STARTVUE.COM :

```
$ ! This command procedure starts up a vue process.
$ ! It is not advisable to edit this procedure as it may be replaced in
$ ! future software updates.
$ set noverify
$ on error then goto done
$ run sys$system:vue$master.exe
$done:
$ logout
```

Le "Startup" de DECwindows

Demarrage du FileView depuis le Session Manager :

- Le nom logique DECW$STARTVUECOM se trouve dans la table DECW$LOGICAL_NAMES et pointe vers le fichier de startup du FilevVew : DECW$STARTVUE.COM
- DECW$TARTVUE.COM est execute si et seulemement si l'utilisateur le specifie dans le menu "Customize/Session"
- Le ressource correspondante au fait que l'on demarre le FIleView se trouve dans DECW$SM_GENERAL.DAT sous la forme :

  sm.create_vue: #

  ou # designe le nombre de FileView(s) a creer. Le menu de "customization" du Session Manager a seulement 2 fonctions: enable/disable. Il est possible de modifier le fichier DECW$SM_GENERAL.DAT pour augmenter le nombre de FileView(s) : c'est non supporte.

- LOGINOUT traduit le nom logique SYS$OUTPUT pour obtenir le numero du peripherique WSA# et definit DECW$DISPLAY dans le table JOB
- Puisque DECW$STARTVUE.COM est execute depuis LOGINOUT, SYLOGIN et LOGIN sont executes.
- Creation d'une boite aux lettres pour communiquer avec le Session Manager
- Impression d'un message sur le control panel du Session Manager

Le "Startup" de DECwindows

Creation de DECterm depuis le Session Manager :

- La resource "sm.create_terminal" trouvee dans DECW$SM_GENERAL.DAT est le nombre de DECterms crees par le Session Manager lors de son "startup". Le defaut est 0.

- Procedure de creation :

  -- Appel de DwtDECtermPort() qui retourne un numero de peripherique WSA

  -- Status retourne en cas d’erreur :

    > "No logical name match" - Boite aux lettres non definie

    > "Device Timeout" - Systeme trop charge, Pagefile trop petit ou trop fragmente

  -- En cas de succes, reset du nombre d’essais successifs du Controller (maximum 10)

-- Creation d’un process avec les caracteristiques suivantes :

- > Image : SYS$SYSTEM:LOGINOUT.EXE
- > SYS$INPUT=SYS$OUTPUT=SYS$ERROR= WSA#
- > Nom de process : <USERNAME>_TWA# ou # est le nom de peripherique WSA
- > Flags du process :
  >DETACHED
  >INTERACTIVE
  >NOPASSWORD

-- Puisque LOGINOUT s’execute, SYLOGIN.COM et LOGIN.COM sont executes

Le "Display Server" DECwindows

- C’est le process qui effectue l’interface entre l’application cliente et le hardware graphique
- Reside sur la station de travail ou se trouve le hardware de visualisation
- Gere l’acces reseau :
  - -- Local ou remote
  - -- Interprete les messages reseau
- Effectue les operations 2D :
  - -- Lignes
  - -- Rectangles
  - -- Polygones
  - -- Cercles
  - -- Texte

- Dirige les entrees utilisateur (clavier, souris) vers les applications clientes

Le "Display Server" DECwindows

- Le client et le server communiquent a l’aide d’un protocole : X11
- Construit au dessus de la couche transport
- Les transports sont :
  - -- Local (Memoire partagee)
  - -- DECnet
  - -- TCP/IP (Ultrix)
- Le X server est divise en 2 parties importantes :
  - -- DIX (Device Independant X) : Operations independantes du hardware graphique
  - -- DDX (Device Dependant X) : Operations dependantes du hardware graphique

Le "Display Server" DECwindows

Le "Display Server" DECwindows

Le "Startup" de DECwindows

Les fichiers du "display server"

SYS$SYSTEM :

- DECW$SERVER_MAIN.EXE
- DECW$SERVER_MAIN_xx.EXE
- DECW$SERVER_EXTENSIONS_*.EXE

SYS$LIBRARY :

- DECW$SERVER_DIX.EXE
- DECW$SERVER_DDX_GA.EXE
- DECW$SERVER_DDX_GB.EXE
- DECW$SERVER_DDX_GC.EXE
- DECW$TRANSPORT_COMMON.EXE

- DECW$TRANSPORT_LOCAL.EXE
- DECW$TRANSPORT_DECNET.EXE

SYS$LOADABLE_IMAGES :

- INDRIVER.EXE
- IMDRIVER.EXE
- IKDRIVER.EXE
- GAADRIVER.EXE
- GABDRIVER.EXE
- GBBDRIVER.EXE
- GCADRIVER.EXE
- GCBDRIVER.EXE
- YEDRIVER.EXE

Fichiers Keymap

- DECW$KEYAP:*.DECW$KEYMAP

Fichiers Font

- DECW$FONT:*.DECW$FONT

Fichiers Securite

- SYS$MANAGER:
  - -- DECW$SERVER_ACCESS_ALLOWED.DAT
  - -- DECW$SERVER_ACCESS_TRUSTED.EXE

- # Fichiers Log

- SYS$MANAGER:
  - -- DECW$SERVER_#_OUTPUT.DAT
  - -- DECW$SERVER_#_ERROR.DAT

Fichiers Dump

- SYS$MANAGER:
  - -- DECW$SERVER_MAIN.DMP

Le "Display Server" DECwindows

Noms logiques du server

- DECW$SERVER#_TABLE :

  DECW$COLOR
  DECW$DEFAULT_FONT
  DECW$DEFAULT_KEYBOARD_MAP
  DECW$FONT
  DECW$KEYBOARD
  DECW$KEYMAP
  DECW$MONITOR_DENSITY
  DECW$POINTER
  DECW$RGB_*
  DECW$SERVER_ACCESS_ALLOWED
  DECW$SERVER_ACCESS_TRUSTED
  DECW$SERVER_DISABLE_CH
  DECW$SERVER_SCREENS
  DECW$SERVER_RETRY_WRITE_MIN
  DECW$SERVER_RETRY_WRITE_MAX
  DECW$CLIENT_ERROR_THRESHOLD
  DECW$SERVER_ERROR_THRESHOLD
  DECW$SERVER_EXTENSIONS
  DECW$XSIZE_IN_PIXELS
  DECW$YSIZE_IN_PIXELS
  DECW$XPORT_SYNC_TIMEOUT

- DECW$LOGICAL_NAMES :

  DECW$SERVER_TRANSPORTS

- LNM$SYSTEM :

  DECW$SERVER_MBX_*_#

Le "Display Server" DECwindows

Le "Display Server" et le hardware :

- **Valeurs pour VAXstation II monochrome**

Driver d'entree	**GCADRIVER.EXE**
Driver de sortie	**GCADRIVER.EXE**
Peripherique de sortie	**GCA0:**
Peripherique d'entree	
Clavier	**GCA1:**
Souris	**GCA2:**
Image DDX	**DECW$SERVER_DDX_GC.EXE**

- **Valeurs pour VAXstation 2000 monochrome**

Driver d'entree	**YEDRIVER.EXE**
Driver de sortie	**GCADRIVER.EXE**
Peripherique de sortie	**GCA0:**
Peripherique d'entree	
Clavier	**TTA0:**
Souris	**TTA1:**
Image DDX	**DECW$SERVER_DDX_GC.EXE**

- **Valeurs pour VAXstation II/GPX et 3200/3500 couleur**

Driver d'entree	**GAADRIVER.EXE**
Driver de sortie	**GAADRIVER.EXE**
Peripherique de sortie	**GAA0:**
Peripherique d'entree	
Clavier	**GAA1:**
Souris	**GAA2:**
Image DDX	**DECW$SERVER_DDX_GA.EXE**

- **Valeurs pour VAXstation 2000 et 3100 couleur**

Driver d'entree	**YEDRIVER.EXE**
Driver de sortie	**GABDRIVER.EXE**
Peripherique de sortie	**GAA0:**
Peripherique d'entree	
Clavier	**TTA0:**
Souris	**TTA1:**
Image DDX	**DECW$SERVER_DDX_GA.EXE**

- **Valeurs pour VAXstation 3520/3540**

Driver d'entree	**YEDRIVER.EXE**
Driver de sortie	**GBBDRIVER.EXE**
Peripherique de sortie	**GBA0:**
Peripherique d'entree	
Clavier	**TTA0:**
Souris	**TTA1:**
Image DDX	**DECW$SERVER_DDX_GB.EXE**

Le "Display Server" DECwindows

Les mecanismes de transport DECwindows :

- Transport Local :
  - -- Utilise le memoire partagee
  - -- Le plus rapide
- Transport DECnet :
  - -- L'objet DECnet X$X0 doit exister dans la base volatile DECnet
  - -- Necessite un pipeline quota de 10000 bytes (taille de buffer en bytes/link pour les packets sans acquittement)
  - -- Si DECnet est arrete et relance, l'objet n'existera pas : il faut relancer DECwindows par la commande :

    @SYS$MANAGER:DECW$STARTUP RESTART

Le "Display Server" de DECwindows

La securite par defaut du server :

Ceci concerne la securite quand le Session Manager ne tourne PAS

- Fichiers :

  SYS$MANAGER:DECW$SERVER_ACCESS_ALLOWED.DAT
  SYS$MANAGER:DECW$SERVER_ACCESS_TRUSTED.DAT

- Noms logiques :

  DECW$SERVER_ACCESS_ALLOWED
  DECW$SERVER_ACCESS_TRUSTED

- **Ces noms logiques n’ont pas a etre utilises. Si ces noms logiques n’existent pas, on utilise les fichiers par defaut ci-dessus**

- **Le securite par defaut quand le Session Manager ne tourne pas et que ces fichiers n’existent pas permet l’acces du compte SYSTEM du noeud local (* 0 SYSTEM)**

Le "Display Server" DECwindows

Le format des fichiers securite :

- **Syntaxe :**

  TRANSPORT NOEUD USERNAME DROITS

- **Transport :**
  >Decrit le transport utilise pour les connections
  >Peut etre DECnet ou * (pour tout type de transport)

- **Noeud :**
  >Le nom de noeud du systeme autorise a se connecter
  >Nom de noeud ou * (pour tout nom de noeud)

- **Username :**
  >Nom de l’utilisateur associe a la demande de connection

- **Droits d’acces :**
  >Permet d’autoriser ou d’interdire une connection
  >Peut etre ALL, NONE ou * (ALL)
  >Le defaut est ALL

Le "Display Server" DECwindows

Securite :

Allowed :

> Les clients sont autorises a se con necter a ce server

Trusted :

> Les clients sont autorises a se con necter et a MODIFIER la liste des cli ents autorises

> La liste "Trusted" est copiee dans la liste "Allowed" quand il y a un "reset" du server et si une liste "Allowed" exi stait deja, elle est mise en fin de la meme liste

Le "Display Server" DECwindows

Les limites du server sous VMS :

- Nombre maximum de connections : 32
- Taille maximum de requete : 16384 bytes

Le parametrage par defaut du server sous VMS :

- DECW$SERVER_RETRY_WRITE_MIN
  - -- Represente le minimum de temps pendant lequel le server tente a nouveau une ecriture sans re- ponse
  - -- Valeur en 1/10 de millisecondes
  - -- Temps entre 2 "retries"
  - -- Defaut 15000 soit .15 secondes
- DECW$SERVER_RETRY_WRITE_MAX
  - -- Represente le maximum de temps pendant lequel tente a nouveau une ecriture sans reponse

-- Valeur en 1/10 de millisecondes

-- Valeur par defaut 300000 soit 3 secondes

- DECW$XPORT_SYNC_TIMEOUT

  -- Temps "elapsed" pendant lequel le server autorise une connection en "mauvais etat" a rester connectee avant de casser le lien logique reseau. Cela se passe en general quand le "network pipeline" est plein

Le "Display Server" DECwindows

Attention, les valeurs de :

DECW$SERVER_RETRY_WRITE_MIN
DECW$SERVER_RETRY_WRITE_MAX

auraient du etre parametrees en millisecondes et non en 1/10 de millisecondes. Il faut donc multiplier les valeurs par defaut par 10 :

DECW$SERVER_RETRY_WRITE_MIN = 1.5 secondes
DECW$SERVER_RETRY_WRITE_MAX = 30 secondes

Le "Display Server" DECwindows

La gestion d’erreur du server :

- Un condition handler permettant d’absorber les exceptions run-time et de les gerer peut etre declare au niveau du server
- Ce condition handler peut etre supprimer en definissant un nom logique :

  DECW$SERVER_DISABLE_CH

- Ce condition Handler est supprime si le nom logique DECW$SERVER_DUMP est defini
- Les valeurs de seuils d’erreur peuvent etre definis par les noms logiques :
  - -- DECW$CLIENT_ERROR_THRESHOLD (def. 1)
  - -- DECW$SERVER_ERROR_THRESHOLD (def. 10)

- La connection client est arretee quand la valeur de seuil est depassee
- Un message d’erreur est envoye en "broadcast" quand la valeur de seuil est depassee

Le "Display Server" DECwindows

Ou regarder quand un probleme survient avec le server :

Fichiers de sortie du server :

- SYS$MANAGER:DECW$SERVER_#_OUTPUT.LOG
- SYS$MANAGER:DECW$SERVER_#_ERROR.LOG
- Fichiers ASCII
- Dans le fichier d’erreur :
  >Information sur le "startup"
  >Information sur un "reset" eventuel
  >Adresses utiles
  >Status du server
  >Stack dump

Fichiers de dump du server :

- SYS$MANAGER:DECW$SERVER_MAIN.DMP
- Fichier binaire analysable par $ANALYZE/PROCESS
- Supprime par defaut
- Peut etre demarre par l’utilisation du nom logique DECW$SERVER_DUMP

Le "Display Server" de DECwindows

"Tuning" du server pour obtenir le maximum de connections :

- Le server est un process detache dont la majorite du code s’execute en mode user
- Surveiller les problemes de quotas sous SDA
- Une connection en mode LOCAL consomme 2 locks par connection
- ENQLM doit etre augmente (66 est une bonne valeur (2*32+2))
- GBLPAGFIL = 16384 est un bon point de depart

Idees de "Customization"

- La "customization" des fichiers de ressources X sous VMS n’est pas documentee ni supportee
- Raisons :
  - -- Le support deviendrait trop complexe en raison de la mediocre qualite des messages d’erreur
  - -- La gestion d’erreur dans le Ressource Manager X n’est pas robuste (erreur de syntaxe ne generant pas de message d’erreur)
  - -- La version 2 de DECwindows (V5.3) integrera la plupart des ressources dans des declarations UIL

Les applications sous DECwindows

Applications Locale ou Remote : Avantages Inconvenients

- Local (client et server sur le meme noeud) :

  -- Avantages :

  >I/Os plus rapides
  >Messages de type Round Trips (drags, queries) plus agrables
  >Plus facile a parametrer

  -- Inconvenients :

  >Toutes les ressources sont consommees sur la meme machine
  >Peut poser un probleme (Memoire, CPU)

- Remote (client et server sur 2 noeuds DECnet differ-ents) :

  -- Avantages :

  >Partage de ressources. Certaines applications qui sont tres consommatrices de CPU peuvent tourner plus rapidement remote
  >Permet de liberer un peu de resource sur le sys teme local

  -- Inconvenients :

  >Les performances sont dependantes de la charge reseau
  >Les applications qui communiquent frequem ment avec le server peuvent etre plus lentes
  >Environnement utilisateur plus complexe

Les applications sous DECwindows

Methodes pour faire tourner une application DECwindows :

- Utilisation de RUN ou MCR :

  $RUN SYS$SYSTEM:DECW$BOOKREADER
  $MCR DECW$BOOKREADER

- Execution a l'aide du FileView

- Utilisation d'un SPAWN/NOWAIT

- Soumission d'une procedure de commande pour executer l'application DECwindows :

  $SUBMIT SYS$LOGIN:APPLI.COM

  $SUBMIT/REMOTE NODE::APPLI.COM

- Demarrage d’un process detache :

  ```
  $Run/Detached/Authorize/Input=-
  Sys$Login:Appli.Com/Output=NL:/Error=NL:-
  /Proc="APPLI" Sys$System:LOGINOUT
  ```

Les applications sous DECwindows

Comment indiquer a l’application ou se connecter :

- DECwindows utilise un nom logique et un pseudo-device pour obtenir les informations necessaires a l’application pour se connecter sur le bon server :
  - -- Nom logique : DECW$DISPLAY
  - -- Pseudo-device : WSA#
- Le nom logique pointe sur le pseudo-device qui est du type :
  - -- Device Class : DC$_WORKSTATION
  - -- Device Type : DC$_DECW_PSEUDO
- WSA0: est le "template device" utilise pour effectuer des "clones"
- Fonctionne comme le "nul device" NLA0: en cas de lecture ou ecriture

- Permet en cas de $QIO Sense de recuperer les informations suivantes :

    >Noeud
    >Transport
    >Numero de Server
    >Numero d’ecran

- Utilise par XOpenDisplay pour obtenir les informations sur les connections

Les applications sous DECwindows

Ou est cree DECW$DISPLAY et WSA ? :

- Sur une station, quand l'utilisateur demarre sa session, chaque DECterm cree a son propre nom logique DECW$DISPLAY defini dans la table JOB. C'est le peripherique cree quand DECwindows est demarre dans la procedure SYS$MANAGER:DECW$STARTAPPS.COM.

- Il n'est donc pas necessaire de creer un peripherique WSA quand on fait tourner une application DECwindows depuis un DECterm ou un FileView

- Par contre, quand on fait un $SET HOTST (0 ou remote), DECW$DISPLAY n'est pas cree, il faut utiliser la commande $SET DISPLAY/CREATE qui place le nom logique DECW$DISPLAY dans la table PROCESS

- Il n’est pas recommande de definir le nom logique a la main :

  $DEFINE DECW$DISPLAY "NODE::0.0"

  car certaines applications necessitent un peripherique WSA#

Les applications sous DECwindows

Les problemes de peripheriques WSA :

- La documentation indique que lors d'un $SET DISPLAY/CREATE/NODE=, le defaut pour le /NODE est SYS$REM_NODE si defini et sinon SYS$NODE; ce n'est pas le cas, le defaut est 0
- Quand un peripherique WSA est cree par la commande $SET DISPLAY, le peripherique WSA# cree devrait etre supprime lors d'un logout mais ce n'est pas le cas. Il reste donc des peripheriques WSA# (avec un reference count egal a 0) sur le systeme. Probleme fixe avec VMS V5.2 ?
- Ne pas utiliser la commande $SET DISPLAY/NOPERMANENT car cette commande supprime la table de noms logiques process !!! Il n'est donc pas possible de supprimer les peripheriques WSA# qui restent sur le systeme. Probleme resolu avec DECwindows V2 (VMS V5.3)

- Il n'y a pas de protection de peripheriques WSA# par defaut : l'UCB contient l'UIC du process ayant cree ce peripherique mais les protections sont de type W:RWLP. Les problemes poses sont les suivants :

  >Un utilisateur peut oublier le /CREATE lors d'un SET DISPLAY et changer ainsi le peripherique WSA cree par DECW$STARTAPPS.COM. Cela se termine en general par un ACCVIO du server

  >Un utilisateur peut supprimer le peripherique WSA utilise par DECW$STARTAPPS.COM. Cela se termine aussi par un ACCVIO du server

  >Un utilisateur malicieux peut supprimer ou modi fier les caracteristiques d'un peripherique WSA# qui ne lui appartient pas

  > Non fixe en V5.3 (pour l'instant)

Les applications sous DECwindows

Une commande non documentee :

$SET DISPLAY/SYMBOL

permet de positionner les symboles globaux :

>DECW$DISPLAY_NODE
>DECW$DISPLAY_SCREEN
>DECW$DISPLAY_SERVER
>DECW$DISPLAY_TRANSPORT

Cela peut etre utile pour ecrire des procedures de commande

Ces symboles ne sont pas supprimes par $SET DISPLAY/NOPERM

Les applications sous DECwindows

Execution de clients DECwindows avec RUN ou MCR:

Avantages :

- Facile a "debugger" car SYS$OUTPUT et SYS$ERROR sont associes au terminal
- Tres simple a demarrer

Inconvenients :

- Utilise le terminal virtuel

Exemple :

$RUN SYS$SYSTEM:DECW$CALC

$MCR SYS$SYSTEM:DECW$CALC

Les applications sous DECwindows

Execution de clients DECwindows avec le FileView :

Avantages :

- Les clients heritent du display DECW$DISPLAY
- Appel a travers un menu

Inconvenients :

- Les processes sont lances en SPAWN
- Probleme eventuel de partage de quotas
- Quand on sort du Fileview, les processes sont supprimes

Les applications sous DECwindows

Execution de clients DECwindows avec SPAWN/NOWAIT :

Avantages :

- Les clients heritent du display DECW$DISPLAY
- Libere le terminal
- Facile a utiliser

Inconvenients :

- Probleme eventuel de partage de quotas
- Les sous-processes sont supprimes lors d’un LO-GOUT

Exemple :

$SPAWN/NOWAIT/INPUT=NL:/PROCESS="Calc"-
SYS$SYSTEM:DECW$CALC

Les applications sous DECwindows

Execution de clients DECwindows en batch :

Avantages :

- Libere le terminal
- Utile pour demarrer DECwindows remote sans se connecter remote ($SUBMIT/REMOTE)

Inconvenients :

- Necessite l’ecriture d’une procedure de commande
- Peut necessiter l’utilisation d’une queue batch speciale parametree par DECwindows
- En cas de probleme, il faut regarder le fichier log

Les applications sous DECwindows

Exemples:

Local :

$SUBMIT RUNCALC.COM

RUNCALC.COM

```
$SET DISPLAY/CREATE/NODE=0/TRANSPORT=LOCAL
$RUN SYS$SYSTEM:DECW$CALC
$SET DISPLAY/NOPERM
$EXIT
```

Remote :

$SUBMIT/REMOTE NODE::RUNCALC.COM

RUNCALC.COM

```
$SET DISPLAY/CREATE/NODE=MYNODE/TRANSPORT=DECNET
$RUN SYS$SYSTEM:DECW$CALC
$SET DISPLAY/NOPERM
$EXIT
```

Les applications sous DECwindows

Execution de clients DECwindows comme process detache :

Avantages :

- Libere le terminal
- Flexible

Inconvenients :

- Necessite l’ecriture d’une procedure de commande
- Complexe pour l’utilisateur
- Plus difficile a "debugger"
- Peut necessiter d’augmenter les parametres PQL_D*

Les appliations sous DECwindows

Exemples :

Pas de besoin de CLI :

**$RUN/DETACH/PROCES="Clock"/AUTHORIZE/INPUT=NL:/OUTPUT=WSA1:-
/ERROR=CLOCK.LOG SYS$SYSTEM:DECW$CLOCK**

Besoin de CLI :

**$RUN/DETACH/PROCES="Clock"/AUTHORIZE/INPUT=CLOCK.COM:/OUTPUT=C
LOCK.LOG/ERROR=CLOCK.LOG SYS$SYSTEM:LOGINOUT**

CLOCK.COM :

**$SET DISPLAY/CREATE/NODE=0/TRANSPORT=LOCAL
$RUN SYS$SYSTEM:VUE$MASTER
$SET DISPLAY/NOPERM
$EXIT**

Le "Session Manager" DECwindows

Fonction :

- Controle de la session de login de l'utilisateur
- Permet :
  >La creation de DECterm(s)
  >Le demarrage de FIleView(s)
  >L'impression d'ecran
  >La "customization"
- Noms logiques du "Session Manager" :

  > DECW$LOGICAL_MAMES :

  DECW$KEYMAP
  DECW$STARTVUECOM
  DECW$SYSTEM_DEFAULTS

  > SYSTEM (ou user pour le print widget)

  DECW$PRINTER_FORMAT_TEXT
  DECW$PRINTER_FORMAT_LINE
  DECW$PRINTER_FORMAT_TERM
  DECW$PRINTER_FORMAT_ANSI2
  DECW$PRINTER_FORMAT_ANSI
  DECW$PRINTER_FORMAT_REGIS
  DECW$PRINTER_FORMAT_TEK

DECW$PRINTER_FORMAT_DDFF
DECW$PRINTER_FORMAT_DDIF

- Quand on effectue une operation "customize", les fichiers suivants sont modifies dans SYS$LOGIN

  Attributs de couleurs :

  >DECW$SM_COLOR.DAT (Moniteur Couleur)
  >DECW$SM_GRAY.DAT (Moniteur multiplan B/W)
  >DECW$SM_BW.DAT (Moniteur B/W)

  Les autres attributs :

  >DECW$SM_GENERAL.DAT

- Il est possible de remplacer les definitions des imprimantes (noms logiques precedents) en definissant une search-list :

  $DEFINE DECW$PRINTER_FORMAT_PS "POST-SCRIPT,SCRIPT,PS"

- Il ne faut pas utiliser le print widget en V1.0 (probleme fixe en V5.2) car il peut y avoir des ACCVIOs pendant l'operation d'impression lors d'un cancel generant un ACCVIO du "Session Manager" laissant la session dans un etat bizarre. Il faut passer par la fonction de CAPTURE suivi d'un PRINT DCL

Le "Window Manager" DECwindows

Fonction :

- Sous DECwindows, c’est l’utilisateur qui "gere" l’ecran, pas l’application
- L’utilisateur decide de :
  >Changer la taille des fenetres
  >Choisir les fenetres visibles
  >Choisir la fenetre qui a l’"input focus"
- Les applications ne doivent pas se "battre" contre le Window Manager
- Le "Window Manager" permet de :
  >Changer la taille des fenetres
  >Bouger les fenetres
  >Changer l’ordre
  >Iconifier/Desiconifier
  >Changer d’"input focus"
  >S’assurer que c’est bien l’utilisateur qui "gere" l’ecran

- Le fichier de ressources est DECW$SYSTEM_DEFAULTS:DECW$WINMGR.DAT
- On ne peut pas modifier les quotas process du "Window Manager" :

> PGFLQUOTA	5000
>BYTLM	10000
>WSEXTENT	4000
>FILLM	92
>DIOLM	100
>BIOLM	60
>ASTLM	100
>WSDEFAULT	512
>WSQUOTA	650

- Pour les memes raisons que precedemment, on ne documentera pas les possibilites de "customization" a partir du fichier de ressource

Le "Terminal Emulator" DECwindows

Fonction :

- Permet d’emuler des terminaux de type VTxxx :

  > VT52
  >VT1xx
  >VT2xx
  >VT3xx

- Permet aussi une emulation Regis et Sixel

- Fichiers de ressource :
  >DECW$SYSTEM_DEFAULTS:DECW$TERMINAL.DAT
  >DECW$SYSTEM_DEFAULTS:DECW$TERMINAL_DEFAULT.DAT

- Noms logiques :

  -- DECW$DECTERM_OUTPUT : Permet de definir le fichier utilise pour SYS$ERROR quand le controller est demarre depuis le Session Manager. Doit etre dans la table PROCESS

-- DECW$DECTERM_CTRL_WSEXTENT : Valeur pour le WSEXTENT du DECterm controller

-- DECW$DECTERM_CTRL_WSQUOTA : Valeur pour le WSQUOTA du DECterm controller

-- DECW$DECTERM_CTRL_WSDEFAULT : Valeur pour le WSEFAULT du DECterm controller

-- DECW$DECTERM_CTRL_PSWAPM : Permet de positionner le process flag du process pour ne pas swapper

- Quotas par defaut :

-- WSQUOTA	650
-- WSEXTENT	4000

Le "VMS User Executive" DECwindows

Fonction :

- Permet une interface "easy to use" du DCL
- Fichier ressource :

  >DECW$SYSTEM_DEFAULTS:VUE$DEFAULTS.DAT

- Fichier profil :

  >DECW$LIBRARY:VUE$SYSTEM_PROFILE.VUE$DAT (system)
  >DECW$LIBRARY:*.VUE$DAT (public)
  >DECW$USER_DEFAULTS:VUE$PROFILE.VUE$DAT (prive)

- Noms logiques :

  >VUE$LIBRARY =
  SYS$COMMON:[VUE$LIBRARY.USER],[.SYSTEM]
  Definit la directory des profils et fichiers de commandes du FileView

  >VUE$PROFILE =
  DECW$USER_DEFAULTS:VUE$PROFILE.DAT
  Definit le profil specifique utilisateur

>VUE$PUBLIC_PROFILE = VUE$LIBRARY:*.VUE$DAT
Definit les profils partageables

>VUE$SYSTEM_PROFILE =
VUE$LIBRARY:VUE$SYSTEM_PROFILE.VUE$DAT
Definit le profil system-wide

Le "VMS User Executive" DECwindows

Que sont les fichiers profil :

- Fichiers sequentiels indexes
- Fichiers binaires
- Contient les informations suivantes :
  >Startup View
  >Saved Views
  >Verbs
  >File Type Default verbs
  >Menu Bar Menus
  >Popup Menus
  >Dialog Box settings

Comment sont charges les fichiers profil :

1. FileView charge le profil systeme. Contient Menu, Verb, Saved Menu livre avec DECwindows
2. FileView effectue un "merge" du profil Public

3. FileView effectue un "merge" du profil Prive. Les verbs, Menus contenus dans ce profil sont en "bold" quand est en mode "customization"

Comment modifier le profil public :

1. **DEFINE VUE$PROFILE VUE$LIBRARY:PUB_PROFILE.VUE$DAT**
2. **MCR VUE$MASTER**
3. **Modifier la partie "Verbs and Menus"**
4. **Exit**
5. **DEASSIGN VUE$PROFILE**

Comment modifier le profil system :

1. **DEFINE VUE$SYSTEM_PROFILE NL:A.DAT**
   DEFINE VUE$PUBLIC_PROFILE NL:B.DAT
2. **DEFINE VUE$PROFILE**
   VUE$LIBRARY:SYSTEM_PROFILE.VUE$DAT
3. **MCR VUE$MASTER**
4. **Modifier ce que l’on desire**
5. **Exit**

6. DEASSIGN VUE$SYSTEM_PROFILE, VUE$PUBLIC_PROFILE, VUE$PROFILE

Le "VMS User Executive" DECwindows

Tuning des quotas de l’UAF pour FileView :

- BYTLM : Minimum 10000, plus 5000 par process cree
- PRCLM : Nombre de sous-processes
- ASTLM : Augmenter de 3 par sous-process
- BIOLM : Augmenter de 3 par sous-process
- DIOLM : Augmenter de 3 par sous-processes
- PGFLQUOTA : Dependant des applications
- WSDEF : Au moins 512
- WSQUOTA : Au moins 1024
- WSEXTENT : 3 a 4 fois WSQUOTA

Le Futur

- **VMS V5.2 effectue le demarrage de DECwindows automatiquement (VMS$LPBEGIN-050_STARTUP.COM)**
- **VMS V5.3 (DECwindows V2) supportera :**
  - **-- Internationalisation ("Europeanisation")**
  - **-- Independance du moniteur graphique (PC...)**
  - **-- Support des ecrans "multi-tetes"**
  - **-- Support de TCP/IP comme transport (avec VMS/Ultrix Connection)**
  - **-- Support de LAT comme transport**
  - **-- Compatibilite avec X Version 11 Release 3**
  - **-- Possibilite de "customization" plus elaboree (Icon Box, File View,...)**
  - **-- $CREATE/TERMINAL (permet de creer un DECterm avec toutes les caracteristiques d'un terminal parametrables)**
  - **-- Bien d'autres choses encore**